Hewlett Packard Enterprise

HP-UX vPars and Integrity VM v6.4 Administrator Guide

Abstract

This document is intended for system and network administrators responsible for installing, configuring, and managing vPars and Integrity Virtual Machines. Administrators are expected to have an in-depth knowledge of HP-UX operating system concepts, commands, and configuration. In addition, administrators must be familiar with the Integrity machine console and how to install the operating systems running in the virtual environments (vPars and virtual machines).

Part Number: 762789-004
Published: August 2016
Edition: 2.1

Revision history

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Document Edition Number	Publication Date
762789-004	HP-UX	11i v3	2.1	August 2016
762789-003a	HP-UX	11i v3	2.0	March 2016
762789-003	HP-UX	11i v3	1.9	March 2016
762789-002a	HP-UX	11i v3	1.8	October 2015
762789-002	HP-UX	11i v3	1.7	March 2015

Contents

HPE secure development lifecycle

Starting with HP-UX 11i v3 March 2013 update release, HPE secure development lifecycle provides the ability to authenticate HP-UX software. Software delivered through this release has been digitally signed using HPE's private key. You can now verify the authenticity of the software before installing the products, delivered through this release.

To verify the software signatures in signed depot, the following products must be installed on your system:

- `B.11.31.1303` or later version of SD (Software Distributor)
- `A.01.02.00` or later version of HP-UX Whitelisting (`WhiteListInf`)

To verify the signatures, run: `/usr/sbin/swsign -v -s <depot_path>`

For more information, see software distributor documentation at: **http://www.hpe.com/info/sd-docs**.

NOTE: Ignite-UX software delivered with HP-UX 11i v3 March 2014 release or later supports verification of the software signatures in signed depot or media, during cold installation. For more information, see Ignite-UX documentation at: **http://www.hpe.com/info/ignite-ux-docs**.

---

1 Introduction

With the increased demand for Information Technology in recent years, data centers have seen a rapid growth in the IT infrastructure (servers, storage, networking) deployment. However, this sprawl has resulted in data centers having server hardware that is being underutilized. The same data centers are facing increasing demand for new applications that results in an increased demand for servers to satisfy their customers. These seemingly contradictory situations have led solution architects to conclude that they must be able to make better use of the resources they have already deployed. The HP-UX virtualization continuum offers several virtualization and partitioning technologies to help HP-UX customers deploy mission-critical applications in a manner that best aligns to their business goals.

HP-UX Virtualization Continuum

HP-UX has traditionally catered to differing workload or applications need by offering products based on partitioning and virtualization technologies. The partitioning solutions such as nPartition or Virtual Partition (vPar) have higher degrees of isolation and lesser resource sharing. At the other end of the spectrum, there are products based on Virtualization technology such as Integrity Virtual Machines, which have a higher degree of sharing of resources at the cost of lesser isolation.

HP-UX Virtual Partitions

The HP-UX Virtual Partitions (vPars) product runs multiple instances of HP-UX simultaneously on one server, or nPartition, by dividing it into vPars. Each vPar is assigned its own subset of hardware, runs a separate instance of HP-UX, and hosts its own set of applications. vPars provide application and operating system fault isolation.

The earlier version of the vPars product is Version A.05.10.

HP Integrity Virtual Machine

The HP Integrity Virtual Machine (Integrity VM) is a soft partitioning and virtualization technology that provides operation system isolation, with sub-CPU allocation granularity and shared I/O. The Integrity VM environment consists of two types of components:

- VM Host
- Virtual Machines (also called guests or VMs)

The VM host virtualizes physical processors, memory, and I/O devices, allowing you to allocate them as virtual resources to each VM.

The earlier version of the Integrity Virtual Machines product is Version 4.3.

Technology Convergence – vPars and Integrity VM v6

HP-UX vPars and Integrity VM Version 6 is a product that brings together vPars and Integrity Virtual Machines technology into a single, common, and easy-to-use management environment. Converging a soft partitioning technology and a virtualization technology into a single product, provides customer with a range of options. To improve system utilization, vPars can be preferred for mission-critical workloads that are CPU and IO intensive, whereas Integrity VMs can be chosen for consolidating physical systems into a virtualized environment. vPars and Integrity VM Version 6 solves the problem of lower server utilization and the simultaneous demand for greater server capacity to run mission-critical applications.

vPars and Integrity VM Version 6 provides the following unique features:

- Increased utilization and scalability.
- More flexibility and capacity.
- Improved performance and productivity.

- Better manageability (rich CLI and GUI).
- High Availability through Serviceguard Integrity Virtual Server Tool Kit.
- Virtual iLO remote console for each vPar and Integrity VM instance.

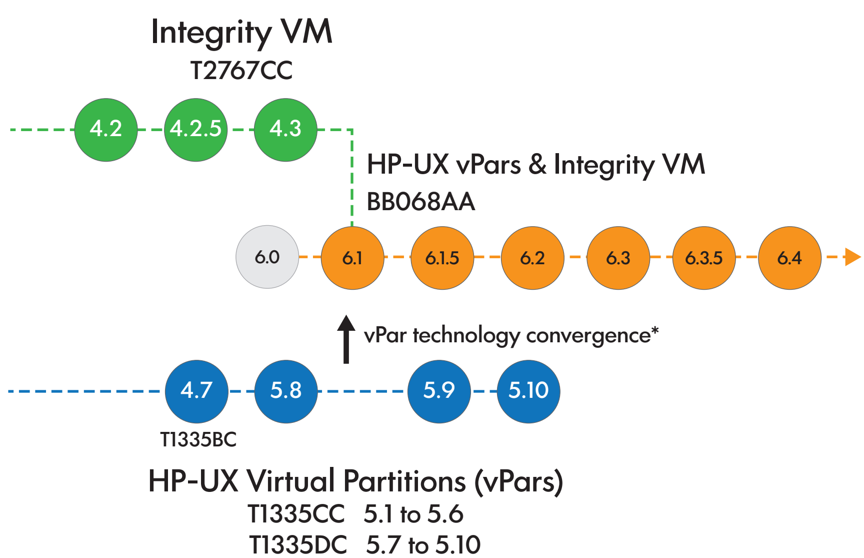
With vPars and Integrity VM Version 6 the vPar solution is purely software based, unlike the earlier vPar technologies which were vPar-monitor based or firmware based. Because the vPar technology is integrated into the Integrity Virtual Machine architecture framework:

- There is no direct upgrade path available from earlier versions of vPars to Version 6.
- Integrity VM guests from earlier versions can be easily upgraded to Version 6.

The Version 6.4 of the product released as part of 11i v3 March 2016 succeeds to Version 6.3.5.

Figure 1 (page 12) shows the details of the product evolution.

Figure 1 Product evolution



NOTE: vPar technology convergence means that similar functionality that was offered with Classic vPar or firmware based vPar product will be available in the vPars and Integrity VM v6.

What is new?

The v6.4 release adds support for:

- Migration across disjoint fabric
- Direct I/O with memory configuration changes
- UDP traffic over Multi-Queue Infrastructure for vPar Guests

The v6.4 with PK2 or superseding patches supports:

- Enablement for HP-UX vPars online guest migration
- Enablement of bandwidth management for NPIV HBAs

NOTE:

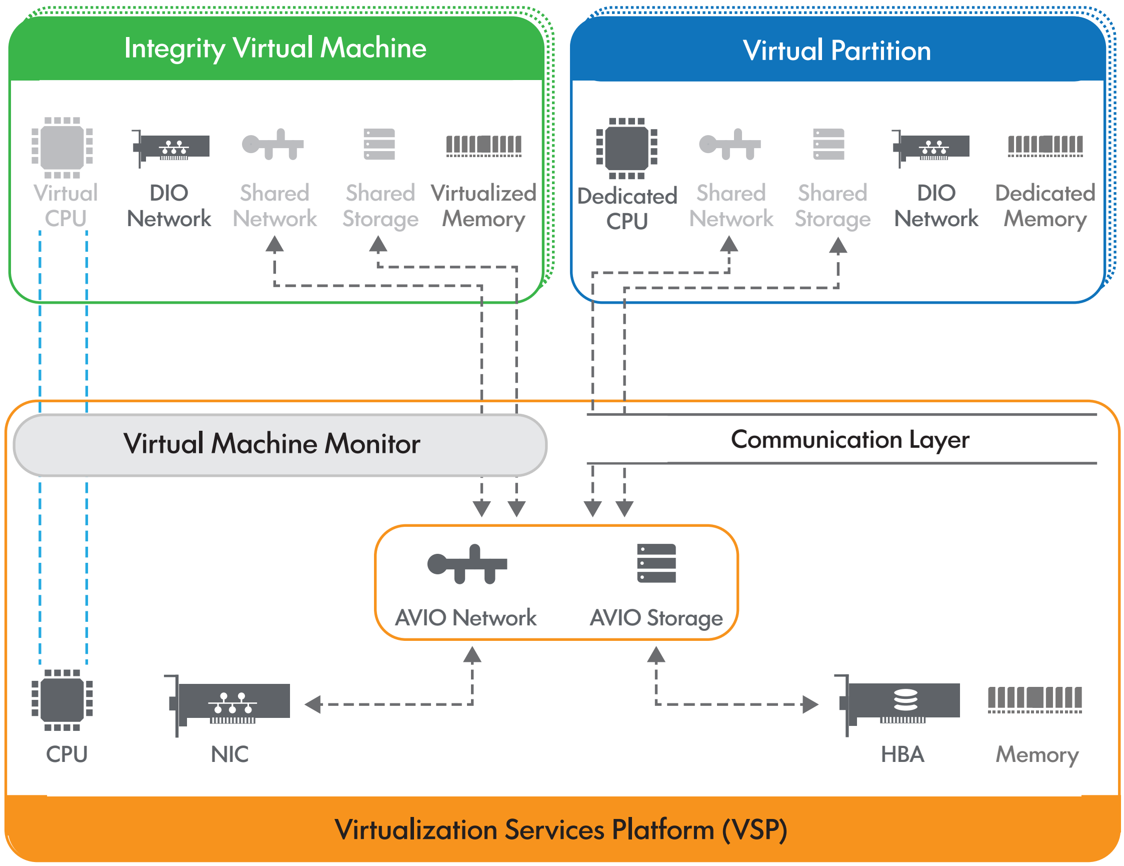
- For more information on software dependencies, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.
- For more information, see “Migrating vPars” (page 231) and “Bandwidth management for NPIV HBAs” (page 108) respectively.

---

vPars and Integrity VM v6 architecture

Figure 2 (page 13) shows the vPars and Integrity VM v6 architecture. The sub-systems are explained in the following sections.

Figure 2 HP-UX vPars and Integrity VM v6 framework



Overview of VSP

The HP-UX host on which the vPars and Integrity VM v6 product is installed is called VSP. The VSP manages the physical resources such as processor cores, memory, and IO devices on the system. The VSP has AVIO sub-systems running for Storage and Network IO. The AVIO sub-systems run on top of physical NIC and HBA instances.

The VSP is the manageability platform from where the vPars and VM guests are created, modified, booted, shutdown, or removed. The VSP provides a set of CLI options and GUI Management tools for administering and monitoring the vPars and Integrity VM instances.

The VSP is a specialized HP-UX host which is optimized to provide maximum system performance for vPars and Integrity VM guest instances, hence DO NOT run any type of resource intensive applications on the VSP. For more details about applications that can and must not be run on VSP, see “Running applications on VSP” (page 35).

For more information about VSP configuration, see “Configuring VSP” (page 30).

Overview of Integrity VM

Integrity VM instances are abstractions of real physical machines. The guest operating system runs on the VM as it would run on a physical Integrity server, with minimal modifications. The environment of the VM is virtualized and managed by the Virtual Machine Monitor (VMM) sub-system that resides on the VSP. Each VM runs an instance of HP-UX (OpenVMS operating system is not supported). Applications running within a VM guest run the same as when run on HP-UX natively. The VM is allocated Virtual CPUs and virtualized memory. The virtual CPUs run a fraction of time on the physical CPU, depending on the percentage of entitlement that is configured for the virtual CPUs and on number of virtual CPUs from other guests sharing the physical CPU and also on the current CPU usage on all the virtual CPUs. Figure 2 (page 13) shows an Integrity VM instance on the left side. The virtual CPU is shown mapped to a physical CPU on the VSP. For more information about virtual CPU entitlements, see “CPU entitlement” (page 148).

Each VM guest requires a minimum of one virtual CPU, one network port, one root disk, memory sufficient for HP-UX, and the hosted applications. The network and storage I/O is through AVIO. Direct IO is also supported on Integrity VMs.

HP-UX 11i v2 and HP-UX 11i v3 are supported as guest operating systems on Integrity VM. There are no set limit to the number of VMs that can be configured, but not more than 254 VMs can be booted simultaneously on a single VSP. For more information about VM attributes, see “Specifying VM attributes” (page 145).

Overview of vPars

Virtual Partition is an instance of an HP-UX 11i v3 operating system having its own dedicated physical cores and dedicated memory. Each instance of HP-UX running in a partition is isolated from all other instances providing application and operating system fault isolation. Applications running on top of HP-UX using vPars run the same as when run on HP-UX native-mode (standalone). Each vPar requires a minimum of one dedicated processor core (CPU), one network port, one root disk, and memory sufficient for HP-UX and the hosted applications. The storage and network I/O is in shared-mode inside the vPar. There are virtual NIC and virtual HBA that are configured and mapped to the AVIO sub-systems on the VSP. The VSP and vPars have a thin communication layer to exchange control information and messages between them. This ensures that vPars can provide near-native performance and provide minimum virtualization overhead to the hosted applications. Direct IO is also supported on vPars.

For more information about configuration limits, see “Configuration limits” (page 149).

Types of I/O

The vPars and Integrity VM supports two types of I/O device – AVIO and DIO. AVIO was introduced with Integrity VM Version 3.5 in December 2007 and since then it has been supported with all further releases of HPVM and vPars and Integrity VM product. AVIO is supported for both storage and networking devices and is available for both HP-UX 11i v2 and HP-UX 11i v3 guests. With HPVM Version 4.2.5, it was also introduced for OpenVMS guests, but, vPars and Integrity VM v6.x does not support OpenVMS guests.

The AVIO feature uses a new storage and networking AVIO guest driver for use within the guests and corresponding AVIO host drivers for use on the HPUX host. The guest drivers are para-virtualized there by eliminating some of the virtualization overhead, and together with the host driver they deliver a streamlined and re-architected I/O path for both storage and networking in improved performance for I/O intensive workloads. For a technical overview on the AVIO feature, see *Integrity VM Accelerated Virtual I/O Overview* at **http://www.hpe.com/info/hpux-hpvm-docs**.

With DIO, vPars and Integrity VM guests can have direct control of an I/O device and is supported only with networking devices. The DIO networking feature minimizes the device emulation overhead and also allows guest operating system to control devices for which emulation does not exist, thus enabling access to I/O hardware technology without requiring the support of either vPars or Integrity VM.

Overview of AVIO storage

To provide the flexibility required to meet a variety of data center needs, the vPar or VM storage subsystem consists of three storage architectures - shared I/O, attached I/O, and NPIV For more information about shared I/O, attached I/O and NPIV, see “Storage devices” (page 60) and “NPIV with vPars and Integrity VM” (page 99).

Shared I/O

The shared I/O architecture is a means by which a vPar and VM guest accesses an entirely virtualized storage subsystem provided by vPars and Integrity VM. The vPar and VM guest storage subsystem emulates real hardware to the vPar and VM guest while interacting with the VSP to complete the vPar or VM I/O operation to the VSP storage entity. This abstraction provides the ability of a VSP administrator to share physical VSP storage hardware across multiple VMs and to allocate that storage at sub-LUN levels.

The individual storage LUNs are shared by dividing a VSP LUN into smaller parts, such as logical volumes, or files. Each of these sub-LUN VSP entities can then be used as a media for separate virtual storage devices. The vPars and VM guests access the virtual storage devices as real storage devices, with no knowledge that the virtual storage media is actually a sub-LUN VSP entity.

The way the virtual storage media is accessed by the vPar or VM guest storage subsystem allows vPars or VM guests to share physical VSP storage adapters. All vPar and VM guest I/O requests in shared I/O are processed by virtual adapters. A virtual adapter is an emulation of a proprietary adapter that a special driver loaded into the guest OS accesses as a real device.

Attached I/O

Attached I/O allows a vPar or VM guest visibility to the real device and its properties. In this architecture, the vPar or VM guest storage subsystem attaches a path to a LUN on the VSP to a virtualized storage adapter. The LUN can be a tape, media changer, or burner.

The main difference between shared I/O and attached I/O is the degree to which a physical storage subsystem is virtualized. In attached I/O, only the storage adapter is virtualized. Therefore, only the VSP physical storage adapters might be shared.

NPIV devices

NPIV is a fibre channel technology that allows you to create multiple virtual Fibre Channel ports over a single physical port on the VSP. These are then assigned to vPars or VM guests on the VSP. With NPIV, a vPar or VM guest discovers SAN devices on its own, just the way it is done on a physical server. The vPar or VM guest views the real targets and devices to which the VSP does not have any visibility. For more information about NPIV and the steps to configure NPIV, see “NPIV with vPars and Integrity VM” (page 99).

AVIO-Networking

AVIO networking feature provides the facility for a vPar or VM guest to communicate with other guests, VSP, and with the outside world through a shared NIC. The shared NIC could be a physical NIC or an APA interface.

Before configuring the guest or virtual LAN interface, a vswitch must be created over a physical NIC. Guests can then be associated with one or more vswitches. The guests are assigned vNICs on these vswitches and these vNICs can be configured like a physical NIC on the host. The vswitch allows configuring VLANs on individual ports thus creating multiple subnets on the same vswitch. A vswitch can also be created without any physical NIC associated with it. Guests associated with such a switch can communicate with each other but not with the VSP or outside the VSP.

For more information about using AVIO networking, see “Creating virtual and direct I/O networks” (page 122).

Direct I/O-Networking

The direct I/O networking feature supported in vPars and Integrity VM Version 6 allows administrators to assign ports (or functions) of a NIC directly to a vPar or VM, giving the vPar or VM direct and exclusive access to the port on the NIC. NIC ports that are configured to be used for DIO cannot be shared and cannot be used to back a vswitch. Before an NIC port or card can be assigned to a vPar or VM, you must first add it to the DIO pool.

For more information about using direct I/O networking, see "Creating virtual and direct I/O networks" (page 122).

Manageability for vPars and Integrity VM v6

Manageability for vPars and Integrity VM v6 is provided with the dedicated GUI tools such as Virtual Server manager (VSMgr), formerly called Virtual Machine Manager (vmmgr). The HP VSMgr is launched through the System Management Homepage (SMH) for a single VSP, or through the dedicated icon in the HP Systems Insight Manager CMS interface. Integrity vPars and VMs can also be managed from the HPE Matrix Operating Environment (HPE MOE) suite of products, which includes HPE Logical Server Management and HPE Infrastructure Orchestrator. For more information about the tools, see "Managing vPars and VMs using GUI" (page 277).

Comparison between Integrity VM v6.4 and vPars v6.4

Table 1 (page 16) provides the feature comparison between Integrity VM v6.4 and vPars v6.4.

Table 1 Comparison between Integrity VM v6.4 and vPars v6.4

Feature	Integrity VM v6.4	vPars v6.4
CPU: Granularity	Sub-core (as little as 5%)	Core
CPU: Dynamic	Enable or Disable, Entitlements (see "vCPU entitlements" (page 49))	Online CPU Add or Delete
CPU: Scalability	32 cores	Server cores minus VSP resources
RAM: Scalability	256 GB	Server RAM minus VSP resources
RAM: Dynamic	Yes (Dynamic and Automatic)	Online Memory Add or Delete
STORAGE: AVIO	Yes	Yes
STORAGE: NPIV	Yes	Yes
NETWORK: AVIO	Yes	Yes
NETWORK: DIRECT	Yes	Yes
Migration Support	Online and Offline	Online and Offline
Oracle RAC Certified	No	Yes (ASM and CFS supported)
Supported VSP Server	All Integrity Servers	i2 blades, SD2, rx2800 i2, i4 blades, SD2 i4, rx2800 i4
Dynamic IO	Yes	Yes
PCI Online Replacement[1]	Yes	Yes

[1] Only on SD2 i2 and i4 VSPs

vPars and Integrity VM media

With the March 2008 release, Hewlett Packard Enterprise presents a set of new operating environments for Version 3 of HP-UX 11i. These new operating environments (OEs) provide a richer set of products and improved choices over the original set of HP-UX 11i OEs. Customers can obtain the OE's integration, testing, and ease of deployment, covering a powerful set of software designed to provide business-critical virtualization.

The following are the HP-UX OEs:

- HP-UX 11i v3 Base OE (BOE)

  The BOE provides an integrated HP-UX operating environment for customers who require less complex installation. The Base OE includes the entire original Foundation Operating Environment (FOE), offering complete HP-UX functionality including security, networking, web functionality, and software management applications.

- HP-UX 11i v3 Virtual Server OE (VSE-OE)

  The VSE-OE provides an integrated HP-UX operating environment for customers seeking higher resource utilization or embarking on consolidation projects and need virtualization for a flexibile UNIX environment. The VSE-OE contains all the products included in the BOE (and the original EOE) and a host of other products including the entire VSE suite. The VSE-OE includes HP-UX vPars and Integrity VM (BB068AA) and the VirtualBase bundle.

- HP-UX 11i v3 Data Center OE (DC-OE)

  Business-critical virtualization built-in—The Data Center OE is for customers who are consolidating, or building an infrastructure for the future. Because the powerful software within the DC-OE is integrated and tested with the operating system, it is an effective choice for a highly available virtualized environment. DC-OE is a complete, fully tested, and integrated UNIX offering. The DC-OE includes HP-UX vPars and Integrity VM (BB068AA), and the VirtualBase bundle.

- HP-UX 11i v3 High Availability OE (HA-OE)

  For customers requiring high availability for large mission critical applications, this OE contains all the products included in the BOE (and the original Enterprise OE), plus applications such as HP Serviceguard and HA toolkits required to enable a mission-critical server.

The HP-UX vPars and Integrity VM v6.4 software is distributed on the HP-UX 11i v3 Operating Environment media with the VSE-OE and the DC-OE. To install vPars and Integrity VM, select the optional software bundles for HP-UX vPars and Integrity VM (BB068AA), and Virtualization Base bundle (VirtualBase), before installing or updating HP-UX.

The HP-UX vPars and Integrity VM software for HP-UX 11i v3 is delivered in the following ways:

- As a stand-alone product on the HP-UX 11i v3 Application Software (AR) DVD
- As a product included in the HP-UX 11i v3 VSE-OE
- As a product included in the HP-UX 11i v3 DC-OE

Related products

Some of the HPE products that you can use with vPars and Integrity VM include:

- HP-UX operating system—HP-UX vPars and Integrity VM runs on HP-UX 11i v3 Integrity systems on the VSP. For all Integrity processors, v6.4 requires that you install either the HP-UX 11i v3 March 2016 (AR1603) release or the HP-UX 11i v3 March 2015 (AR1503) release plus AR1603 Feature11i patches. For more information, see *HP-UX 11i v3 Installation and Update Guide*.
- HP WBEM Services for HP-UX—Many related products, such as Virtual Server Manager, require the VSP system to run the WBEM Services.
- HPE Matrix Operating Environment—A graphical user interface for managing HPE Integrity Central Managed Systems (CMS). Runs on HP Systems Insight Manager. For more information, see the *Matrix Operating Environment 7.4 Getting Started Guide*.

- HP Integrity Virtual Server Manager—A graphical user interface for creating and managing vPars and VMs. Runs under either HP System Management Homepage (HP SMH) or HP Systems Insight Manager (HP SIM) as part of the HPE Matrix OE. For more information, see *Integrity Virtual Server Manager 6.4 User Guide*.
- HP Integrity VM Providers—To manage virtual environments with Virtual Server Manager or any Matrix OE components, install the appropriate provider software from the operating system media or the VirtualBase bundle.
- HP-UX GUID Manager (GUIDMgr)—A client-server based product that allocates and manages unique World Wide Names for NPIV Host Bus Adapters.
- VERITAS Volume Manager—A data storage solution product that can be used to manage the physical disks on the VSP. For more information, see *VERITAS Volume Manager Administrator's Guide*.
- HP Serviceguard—A software product that allows you to create clusters of HP-UX systems for high availability. For more information, see the managing serviceguard manual.

Using the vPars and Integrity VM documentation

The vPars and Integrity VM product bundle includes several useful sources of information, whether you are considering how to set up your vPar or VM, or determining how to upgrade the installation.

Integrity VM commands

Integrity VM commands provide a convergence point for vPars and Integrity VM. You can use Integrity VM commands to create, clone, start, and manage not only VMs, but also vPars. You can use vPars commands (whose manpages are listed in Table 4 (page 19)) to manage only vPars. Integrity VM commands provide a superset of features to accommodate both VMs and vPars.

For online information about using Integrity VM commands, see the following manpages on the VSP system.

Table 2 Integrity VM commands

Command	Description
hpvm(5)	Describes the Integrity VM environment.
hpvmclone(1M)	Describes how to create VMs based on existing VMs.
hpvmcollect(1M)	Describes how to collect VM support information.
hpvmconsole(1M)	Describes how to use the VM console.
hpvmcreate(1M)	Describes how to create VMs.
hpvmdevinfo(1M)	Reports about storage for a VM.
hpvmdevmgmt(1M)	Describes how to modify the way virtual devices are handled.
hpvmdevtranslate(1M)	Translates Integrity VM guest devices to agile devices.
hpvmdiorecover(1M)	Attempts to recover DIO-related inconsistencies between the Integrity VM device database, the krs(5), and ioconfig(4) databases.
hpvmhostgdev(1M)	Manages Integrity VSP devices available for VM access.
hpvmhostrdev(1M)	Manages VM access to devices used by the Integrity VSP system.
hpvmhwmgmt(1M)	Allocates resources to the specified resource pool for exclusive use by VMs.
hpvminfo(1M)	Describes how to get information about the VSP.
hpvmmigrate(1M)	Describes how to migrate active guests and offline VMs from one VSP to another.

Table 2 Integrity VM commands *(continued)*

Command	Description
hpvmmodify(1M)	Describes how to modify VMs.
hpvmmove_suspend(1M)	Moves suspend files to a different directory.
hpvmnet(1M)	Describes how to create and modify virtual networks.
hpvmnvram(1M)	Displays, creates, edits, and removes vPar or VM EFI variables in NVRAM files from a VSP.
hpvmpubapi(3)	Describes several new public APIs.
hpvmremove(1M)	Describes how to remove a VM.
hpvmresources(5)	Describes how to specify the storage and network devices used by VMs.
hpvmresume(1M)	Describes how to resume a VM.
hpvmsar(1M)	Displays performance information about one or several guests on the same host.
hpvmstart(1M)	Describes how to start VMs.
hpvmstatus(1M)	Describes how to get statistics about the guests.
hpvmstop(1M)	Describes how to stop a VM.
hpvmsuspend(1M)	Suspends a VM.

The following manpages are also provided in the HP-UX virtual environment:

Table 3 Integrity VM commands in the HP-UX virtual environment

Command	Description
hpvmcollect(1M)	Describes how to collect virtual environment support information.
hpvmdevinfo(1M)	Reports about storage for a virtual environment.
hpvminfo(1M)	Describes how to get information about the VSP.
hpvmmgmt(1M)	Describes how to manage dynamic memory from the vPar or VM.
hpvmpubapi(3)	Describes public APIs.

NOTE: VirtualBase provides the `gvsdmgr` utility, which manages AVIO HBAs. For information about the `gvsdmgr` utility, see HP-UX *gvsdmgr*(1M).

vPars commands

From the VSP you can run vPars commands to create, modify, and remove vPars and virtual switches. To run the commands from the VSP, you need superuser privilege. These commands cannot be run from the OA or from inside a vPar.

Table 4 (page 19) lists a summary of the VSP commands with descriptions of their use. The following section provides brief information about each command. For more information about the commands, see the respective manpages.

Table 4 VSP commands in vPars

Command	Description
vparboot(1M)	Boots a vPar.
vparcreate(1M)	Creates a new vPar.

Table 4 VSP commands in vPars *(continued)*

Command	Description
vparmodify(1M)	Renames or modifies the resources of a vPar. It can also suspend the configuration of the vPar.
vparremove(1M)	Removes an existing vPar.
vparreset(1M)	Resets a vPar. Simulates, at the vPar level, the hard reset, soft reset (Transfer Of Control, TOC), power off, or graceful shutdown operations. When compared with the earlier versions of vPars, the `vparreset` operation closely matches with the operation of physical hardware.
vparstatus(1M)	Displays information about one or more vPars. The `vparstatus` can also display details about the available resources that can be added to a vPar.
vparhwmgmt(1M)	Manages the pool of CPU resources dedicated for use by the vPars on the VSP.
vparnet(1M)	Creates and controls a vswitch.
vparconsole(1M)	Connects to the console of a vPar.

When you use the `vparcreate` command to create a vPar, resources are reserved even while the vPar is off. The vPar is set to automatically boot whenever the VSP reboots. However, if you use the `hpvmcreate` command to create a vPar, the resource reservations are not configured, and the vPar is not set to reboot automatically. For more information about resource reservations, see “Reserved resources and resource over-commitment” (page 54).

Virtual environment console

The virtual environment console is a special interface for managing vPar or VMs. To start the virtual console after you create a vPar or VM, enter either the `vparconsole` command or the `hpvmconsole` command and specify the vPar or VM name. To get help on how to use the virtual console, enter the `HE` command. For more information about the virtual console, see “Using the virtual console” (page 249).

Using this manual

This manual provides all the information you must know to install Integrity VM, create VMs, install, and manage guests, and use all the features of Integrity VM. Table 5 (page 20) describes each chapter in this manual.

Table 5 Chapters in this manual

Chapter	Read if...
Chapter 1 (page 11)	You are new to HP Integrity VMs.
Chapter 2 (page 22)	You are installing either HP-UX vPars and Integrity VM product or guest operating system or both.
Chapter 3 (page 30)	You are configuring the VSP.
Chapter 4 (page 38)	You are upgrading the VSP from earlier versions of Integrity VM.
Chapter 5 (page 49)	You need to understand more about CPU and Memory resource for vPar and VM.
Chapter 6 (page 60)	You are configuring storage to be used by the VSP or virtual environments.
Chapter 7 (page 99)	You are configuring a vPar or VM guest with an NPIV based virtual HBA.
Chapter 8 (page 122)	You need to make changes to the network devices on the VSP system or to the virtual network devices used by the VMs.
Chapter 9 (page 145)	You are setting up a new VM on your VSP system.

Table 5 Chapters in this manual *(continued)*

Chapter	Read if...
Chapter 10 (page 164)	You are setting up a new vPar on a VSP system.
Chapter 11 (page 173)	You need information about PCI OLR support on VSPs.
Chapter 12 (page 203)	You need to move vPars or VMs from one system to another.
Chapter 15 (page 239)	You need to manage an existing vPars, VMs, and resources using CLI.
Chapter 16 (page 277)	You need to manage an existing vPars, VMs, and resources using GUI.
Chapter 17 (page 282)	You need information about HPE support.
Appendix A (page 286)	You encounter problems related to creating VM, storage and NPIV.
Appendix B (page 301)	You encounter problems while creating or using virtual environments.
Appendix C (page 307)	You want to specify multiple storage devices at one time for a guest.
Glossary (page 315)	You do not understand the definition of a term used in the vPars and Integrity VM product documentation.

This manual and the HP-UX vPars and Integrity VM v6.4 release notes are available on the Instant Information DVD or may be viewed, downloaded, and printed from the web at **http://www.hpe.com/info/hpux-hpvm-docs**.

2 Installing HP-UX vPars and Integrity VM

This chapter describes the requirements and procedure for installing vPars and Integrity VM product and guest operating system.

Installation requirements for VSP

Before installing the vPars and Integrity VM product on the VSP, ensure that the following software bundles are installed on the VSP:

- HP-UX 11i v3 March 2016 OE.

  OR

  HP-UX 11i v3 March 2015 (AR1503) plus AR1603 Feature11i patches.
- If using HP Serviceguard, supersede by PHSS_43698 and PHSS_43620 patches.
- If using VxVM 5.0.1, PHKL_43186, PHCO_42677, and PHCO_43185 patches.

  If using VxVM 5.1 SP1, PHKL_43527 and PHCO_43526 patches.

Bundle names

The HP-UX vPars and Integrity VM release contains the following software:

- BB068AA – vPars and Integrity VM.
- VirtualBase – Base virtualization software for vPar/VM and VSP.
- GUIDMGR – GUID Manager software.
- PRMKernelSW – HP PRM Kernel software.
- T8718AC – Intergrity VM Online Migration software.

NOTE:

- GUIDMGR and PRMKernelSW are installed as dependent software bundles of BB068AA and VirtualBase.
- T8718AC is provided as a separate licensed product on the HP-UX 11i v3 Application Release (AR) DVD, VSE-OE, and DC-OE. It must be purchased and installed separately.

Installing vPars and Integrity VM

NOTE: Before installing the product, ensure VSP is installed with the required OS version and patches mentioned in “Installation requirements for VSP” (page 22). You can install this product on a physical Integrity server or an nPar running HP-UX 11i v3. Do not attempt to install it on a vPar.

Some files or software if present on the VSP, might interrupt the installation. Check for the following before beginning the installation:

- Hierarchical Files System (HFS) mount points in the `/etc/fstab` file:

  ```
  # grep -i hfs /etc/fstab
  ```

  If present, change to using VERITAS File System (VxFS).
- HP SIM Server bundle:

  ```
  # swlist | grep HPSIM-HP-UX
  ```

  If present, uninstall it:

```
# swremove HPSIM-HP-UX
```

- HP-UX Virtual Partitions bundle v5.x or earlier:

  ```
  # swlist -l bundle | grep VirtualPartition
  ```

  If present, uninstall it:

  ```
  # swremove VirtualPartition
  ```

To install the HP-UX vPars and Integrity VM software:

Mount the installation media, if you have it (for example, `/depot/path`). If you are installing from the network, identify the VSP and path name that correspond to the software distribution depot that contains the BB068AA and VirtualBase bundles (for example, `my.server.example.com:/depot/path`).

- If you are using the CLI:

  Enter the following `swinstall` command including the path to the depot:

  ```
  # swinstall -x autoreboot=true -s my.server.example.com:/depot/path
  BB068AA VirtualBase
  ```

- If you are using the GUI:

  Set the shell variable, DISPLAY appropriately and invoke the `swinstall` command.

  For example,

  ```
  # export DISPLAY=my.client.example.com:0.0
  # swinstall
  ```

  Select the BB068AA bundle and the VirtualBase bundle from the list presented by the GUI.

NOTE: If you have purchased the Integrity VM Online Migration software, you can install it by selecting the bundle T8718AC.

The VSP system reboots automatically after the installation is completed.

After installation, you will find the components in various locations, as listed in Table 6 (page 23).

Table 6 Components of VSP and their location

Component	Location
Software and man pages	`/opt/hpvm`
VirtualBase software	`/opt/hpvm/guest-images` directory
Commands	`/opt/hpvm/bin` directory
Configuration and data files	`/var/opt/hpvm` directory

You can now create vPar and VM guests using the `hpvmcreate` command.

NOTE: The guest configuration files are stored in the `/var/opt/hpvm/` directory. The new configuration files are not compatible with those of earlier versions of the product. So, to upgrade the current version, the guest configuration files (except the `/ISO-Images/` directory) are saved in the `/var/opt/hpvm/backups/` directory. If you fallback to previous version of the product, use the backup configuration files to restore the VSP and guest configurations.

Verifying the installation of vPars and Integrity VM v6 product

To verify that the installation was successful:

- Enter the `hpvminfo` command:

  ```
  # hpvminfo
  ```

  The following output must be displayed:

  ```
  hpvminfo: Running on an HPVM host.
  ```

- Enter the `swlist` command:

  ```
  # swlist |grep -e "BB068AA" -e "VirtualBase"
  ```

  Check the version numbers.

  ```
  BB068AA       B.06.40 HP-UX vPars & Integrity VM v6
  VirtualBase   B.06.40 Base Virtualization Software
  ```

  NOTE: The `what` string output of HPVM product have the version as `HPVM B.06.40 PATCH_02`.

- Check whether the configuration file `/etc/rc.config.d/hpvmconf` was created.

If you face any issues during the verification, it indicates the installation was not successful. In such cases, contact HPE Support for help.

Uninstalling vPars and Integrity VM

To uninstall the vPars and Integrity VM product on VSP, remove the BB068AA and VirtualBase bundles:

```
# swremove –x autoreboot=true BB068AA VirtualBase
```

NOTE: If you have purchased Integrity VM Online Migration Software bundle T8718AC, you must also uninstall it.

Installing or Reinstalling the HP-UX guest operating system

After a vPar or VM guest is created, you can proceed with the installation of HP-UX guest operating system. For a list of supported versions of the HP-UX operating system, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.

Start up information (boot order and boot path) for guests are stored in a VSP file used to emulate the virtual NVRAM for the guest. This information may be modified as part of installation. For this reason, it is advisable to take a backup of `/var/opt/hpvm/guests/<Guest-Name>/` on the VSP, (for the guest being installed or reinstalled or upgraded) immediately after the installation and stored along with the most recent working copy of the VSP full-system backup.

There are multiple ways to install HP-UX 11i on a vPar or VM guest. The following approach describes the use of the network to directly install HP-UX 11i from an Ignite-UX server. For more information about Ignite-UX based installation, see *Ignite-UX Administration Guide for HP-UX 11i* available on the website at **http://www.hpe.com/info/ignite-ux-docs**.

NOTE: Before using the Ignite-UX server approach, ensure the following:

- The vPar or VM guest is created and assigned a network interface.
- At least one disk has been added to the vPar or VM guest with sufficient space to install HP-UX 11i on it.
- The Ignite-UX server is set up and accessible from the LAN interface assigned to the vPar or VM guest.

Configuring guest lanboot from the VSP

You can use the `hpvmnvram` command from VSP to add a lanboot entry by creating a database profile for the corresponding guest. For more information about the `hpvmnvram` command, see *hpvmnvram*(1M) manpage.

Enabling guest lanboot from the VSP

1. Create database profile named `master` for the VM named `guest1`

```
# hpvmnvram -P guest1 -dn master -cip 15.213.225.26 -sip 15.146.225.227
   -gip 15.213.152.1 -m 255.255.248.0 -b "/opt/ignite/boot/nbp.efi"
You should make a backup copy of this nvram file before proceeding with changes.
  Continue? Enter Y or N:Y
```

2. Add directed lanboot as first boot option for the VM named `guest1` with database profile `master`

```
# hpvmnvram -P guest1 -a 0xB27A4F72629B::master
You should make a backup copy of this nvram file before proceeding with changes.
Continue? Enter Y or N:Y
 hpvmnvram: Adding boot option 'LanBoot:0xB27A4F72629B:master' (0xB27A4F72629B) ..
```

NOTE: To get the MAC address, you can use the `hpvmstatus -P guest1 -d`.

3. List all the boot options in the VM named `guest1`

```
# hpvmnvram -P guest1 -l
Boot Order  EFI Boot Variable  Description
==========  =================  ============
1           Boot0001           LanBoot:0xB27A4F72629B:master
2           Boot0000           EFI Shell [Built-in]
```

4. After adding the lanboot entry successfully, start the VM from the VSP administrator account using the `hpvmstart` command.

```
# hpvmstart -P guest1
(C) Copyright 2000 - 2013 Hewlett-Packard Development Company, L.P.
......
hpvmstart: Successful start initiation of guest 'guest1'
```

5. Connect to the guest console

```
# hpvmconsole -P guest1
vMP MAIN MENU
CO: Console
CM: Command Menu
CL: Console Log
SL: Show Event Logs
VM: Virtual Machine Menu
HE: Main Help Menu
X: Exit Connection
[guest1] vMP>
```

The `hpvmconsole` command opens the VM console. The VM prompt is displayed. From the VM console, you can control the VM as if it is a physical Integrity Server.

6. Enter the `co` command at the VM prompt:

```
[guest1] vMP> co
EFI Boot Manager ver 1.10 [14.62] [Build: Tue Oct 2 03:33:06 2012]
Please select a boot option
   LanBoot:0xB27A4F72629B:master

 EFI Shell [Built-in]
  Boot option maintenance menu
Use ^ and v to change options. Use Enter to select
an option. Default boot selection will be booted in 9 seconds.
```

7. Boot from the newly added lanboot entry and follow the steps as prompted by the install kernel to install HP-UX.

Configuring guest lanboot from EFI

To install the HP-UX operating system on the VM guest named `guest1`, start it from the VSP using the `hpvmstart` command.

1. Start the VM from the VSP administrator account using the `hpvmstart` command.

   ```
   # hpvmstart -P guest1
   (C) Copyright 2000 - 2015 Hewlett-Packard Development Company, L.P.
   ....
   hpvmstart: Successful start initiation of guest 'guest1'
   ```

   After the command is executed, check the status of the VM:

   ```
   # hpvmstatus
   [Virtual Machines]
   Virtual Machine Name VM # OS Type  State   #VCPUs #Devs  #Nets  Memory
   ==================== ==== =======  ======= ====== =====  =====  ======
   config1                1    HPUX     Off     1      5      1     512 MB
   config2                2    HPUX     Off     1      7      1       1 GB
   guest2                 5    HPUX     On(OS)  1      5      1       1 GB
   guest1                12    UNKNOWN  On(EFI) 1      0      0       2 GB
   ```

2. Connect to the guest console:

   ```
   # hpvmconsole -P guest1
    vMP MAIN MENU

            CO: Console
            CM: Command Menu
            CL: Console Log
            SL: Show Event Logs
            VM: Virtual Machine Menu
            HE: Main Help Menu
             X: Exit Connection

   [guest1] vMP>
   ```

   The `hpvmconsole` command opens the VM console. The VM prompt is displayed.

   From the VM console, you can control the VM just as if it were a physical Integrity server.

3. Enter the `co` command at the VM prompt:

   ```
   [guest1] vMP> co

   EFI Boot Manager ver 1.10 [14.62] [Build: Wed Jun  4 11:37:36 2008]

   Please select a boot option

       EFI Shell [Built-in]
       Boot option maintenance menu


       Use ^ and v to change option(s). Use Enter to select an option
   ```

   The `EFI Boot Manager` is displayed.

4. Select the `EFI Shell` and create a data base profile by running the following command:`<Shell> dbprofile -dn newdbprof -sip <IP_address_of_ignite-server> -cip <IP_address_of_vPar> -gip <IP_address_of_gateway> -m <network_mask> -b "/opt/ignite/boot/nbp.efi"`.

   NOTE: IP_address_of_gateway is the IP address of gateway from the LAN domain of the vPar to the LAN domain of the Ignite server. network_mask is the netmask (in dotted notation) of the LAN to which vPar is connected.

5. Now, exit back to the main screen and select `Boot option maintenance menu`:

```
EFI Boot Maintenance Manager ver 1.10 [14.62]
Main Menu. Select an Operation

        Boot from a File
        Add a Boot Option
        Delete Boot Option(s)
        Change Boot Order

        Manage BootNext setting
        Set Auto Boot TimeOut

        Select Active Console Output Devices
        Select Active Console Input Devices
        Select Active Standard Error Devices

        Cold Reset
        Exit
```

   The `EFI Boot Maintenance Manager` is displayed.
6. Select `Add a Boot Option`.

```
EFI Boot Maintenance Manager ver 1.10 [14.62]

Add a Boot Option.  Select a Volume

    Removable Media Boot [Acpi(PNP0604,0)]
    Load File [Acpi(PNP0A03,0)/Pci(1|0)/Mac(763AE48F393F)]
    Load File [EFI Shell [Built-in]]
    Legacy Boot
    Exit
```

   From the displayed options, do one of the following:

   - Select `Removable Media Boot` to install from virtual DVD.
   - Select the entry with your MAC address to install from the Ignite-UX server. For example:

```
Device Path Acpi(PNP0A03,0)/Pci(1|0)/Mac(763AE48F393F)

Enter New Description:  lan0boot
Is This A Directed LAN Boot Option [Y-Yes N-No]: No
Enter db-profile name [max 12 characters] :  newdbprof

Save changes to NVRAM [Y-Yes N-No]:  Y
```

7. Exit the EFI Boot Maintenance Management screen and return to the EFI Boot Manager screen.

   NOTE: For more information about NPIV boot option, see “Installing the guest image on NPIV disks” (page 105).

8. Boot from the appropriate boot entry and follow the steps as prompted by the install kernel to install HP-UX.

NOTE:

- If you are installing from Ignite-UX server, the installation process continues just as if the VM was an Ignite-UX client.
- Installing guest from the co-located Ignite-UX server is not supported configuration. For more information, see "Miscellaneous AVIO Networking problems" (page 297).

Using golden images for guest installation

VSP must not be used to create golden images that will be used for guest OS installations using Ignite-UX. Instead, an Integrity system can be used to create a golden image suitable for OS installation on a VM or vPar, provided it has all of the VSP software, except the VirtualBase bundle removed. To do so, remove these bundles BB068AA, T8718AC, and GUIDMGR:

1. Enter the `swremove` command:

   ```
   # swremove -x autoreboot=true BB068AA T8718AC GUIDMGR
   ```

2. Verify that neither of these bundles are installed:

   ```
   # swlist BB068AA T8718AC GUIDMGR
   # Initializing...
   # Contacting target "foo"...
   ERROR:   Software "BB068AA" was not found on host "foo:/".
   ERROR:   Software "T8718AC" was not found on host "foo:/".
   ERROR:   Software "GUIDMGR" was not found on host "foo:/".
   ```

   These errors must be displayed.

For more information about using Ignite-UX golden images, see *Ignite-UX Administration Guide*.

Installing VirtualBase on a vPar or VM Guest

The guest OS must have the VirtualBase bundle installed to work in a VSP environment. If a new guest OS is installed or an existing VSP is upgraded to v6.4, the corresponding VirtualBase product must be installed in the guest operating system.

NOTE: Required vPar patches cannot be installed on an older guest OS, where the VirtualBase bundle is upgraded to v6.4. For more information on installing the online vPar migration feature, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.

A copy of VirtualBase depot is installed onto the VSP system when vPars and Integrity VM is installed or upgraded. It is stored on the VSP system in the `/opt/hpvm/guest-images` directory. A subdirectory contains an SD tape depot with VirtualBase for the HP-UX operating system, as shown in the following example:

```
# cd /opt/hpvm/guest-images/hpux/11iv3
# ls
hpvm_guest_depot.11iv3.sd
```

Copy the SD tape depot file to a directory in the vPar or VM guest. Before installing the VirtualBase bundle, preview the install task for the installation analysis. This provides the opportunity to identify and address any warnings before the actual installation. For example, the analysis phase includes checks for installation of the appropriate AVIO drivers on the guest. To preview the installation, use the `-p` option of `swinstall` as shown in the following example:

```
# swinstall -p -x autoreboot=true -s path to hpvm_guest_depot.11iv#.sd VirtualBase
```

Installing the vPars or VM VirtualBase software kit causes the vPar and VM guest to reboot.

Each subdirectory in `/opt/hpvm/guest-images` contains a `README.txt` file that describes how to install the software for that type of vPar or VM. For information about any additional software updates, see *HP-UX vPars and Integrity VM v6.4 Release Notes* available at **http://www.hpe.com/info/hpux-hpvm-docs**.

Other patches required

Apart from the VirtualBase bundle, the following software patches are also required for the vPars and VM guests:

- For versions earlier than HP-UX 11i v3 March 2013, the PHKL_43308 patch.
- If using HP Serviceguard, PHSS_42136 and PHSS_42137 patches.

Applications that can be run on a vPar or Integrity VM

You can run the following software in a vPar or VM environment:

- HP-UX 11i v3 Virtual Server Operating Environment (VSE-OE).
- Software installation tools (Ignite-UX and Software Distributor-UX).
- System performance monitoring tools (GlancePlus, Measureware, OpenView Operations Agent).
- Applications such as databases and so on.

Applications to be avoided on a vPar or Integrity VM

Hewlett Packard Enterprise strongly recommends that you do not run the following types of applications on a vPar or VM guest:

- Virtualization platform (HP-UX vPars and/or Integrity VM software).
- Utility pricing tools (run on the VSP).
- Capacity planning tools (run on the VSP).
- Applications that require direct access to physical hardware (for example, disaster-tolerant solutions).

You must purchase licenses for any software you run on a VM or a vPar, including the HP-UX operating system and any Hewlett Packard Enterprise or third-party layered software. You can purchase the licenses for HPE software under the HPE Virtualization Licensing program. For more information, contact your HPE support representative.

Before installing any software product, Hewlett Packard Enterprise recommends to read the product release notes to get the latest information.

3 Configuring VSP

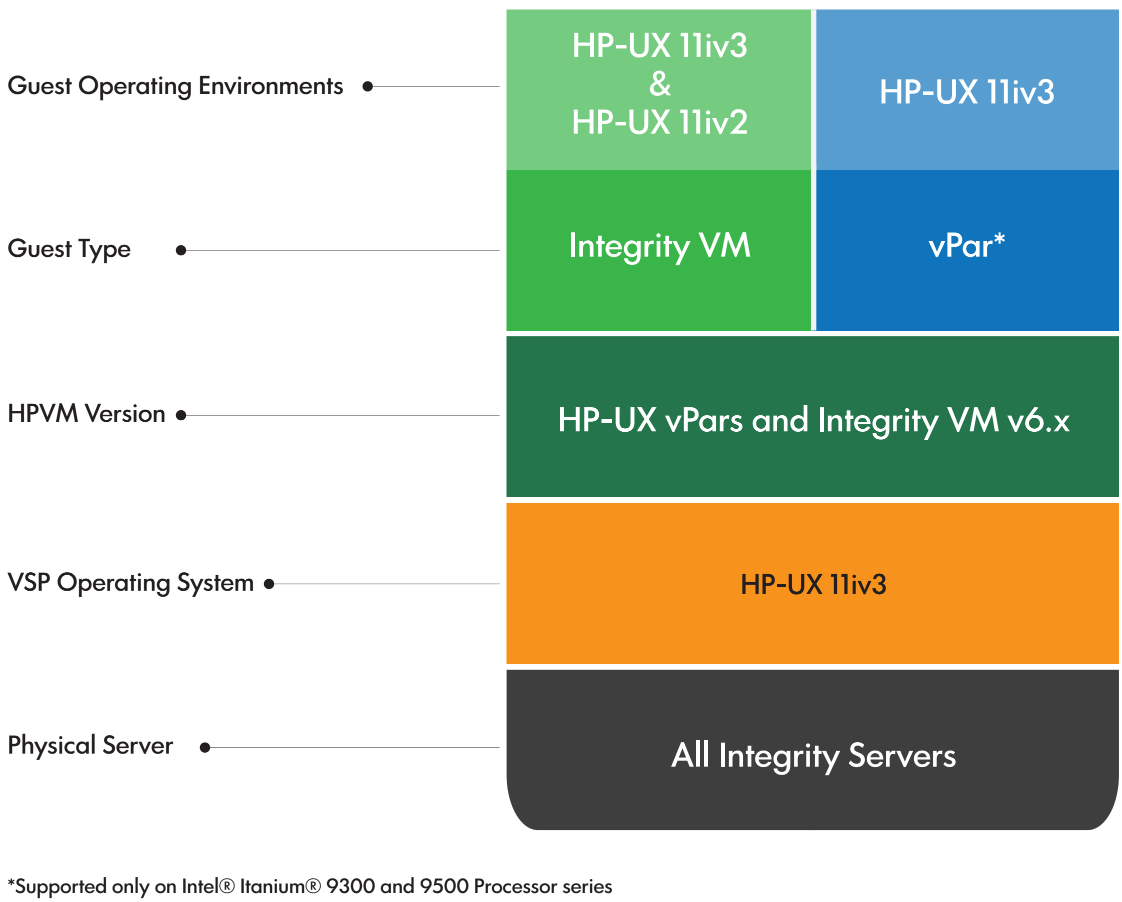
VSP is the manageability platform for vPars and VMs, running the standard HP-UX 11i v3 OE. VSP has a controlled environment tuned for supporting the vPars and Integrity VM v6 product functionality. DO NOT install any application on the VSP, that is CPU or memory or IO intensive in nature. Running such applications on the VSP can cause unpredictable behaviour.

The product startup scripts located at `/sbin/init.d/` configure the VSP resources (cores and memory) during every reboot of VSP. There is no explicit configuration change needed on the VSP, unless otherwise documented.

Starting from vPars and Integrity VM v6.2, you can run both vPars and VMs concurrently on the VSP (mixed mode environment).

Figure 3 (page 30) shows the different layers in a VSP.

Figure 3 Layers in a VSP



VSP cores

The CPU cores are configured into two pools:

- VSP pool
- vPars and Integrity VM pool

To view the allocation of CPU cores, enter the following command:

```
# hpvmhwmgmt -p cpu -l
```

VSP pool

The CPU cores in the VSP pool run normal VSP processes. In addition, the cores run special threads to service I/O requests for vPars and Integrity VM. These cores cannot be used for vPars and VM guest configurations.

By default, there are no cores in the VSP pool. When you configure the first reserved vPar or when you start the first non-reserved vPar, a single core is added to this pool. If vPar configurations exist (for example, after upgrading an existing vPars v6.0 system), the cores reserved for VSP is non-zero. If the configuration does not contain any vPars (for example, after upgrading an existing Integrity VM v4.3 system), the cores reserved for the VSP is zero, matching the Integrity VM Host in a Integrity VM v4.3 environment.

Consider the function of the VSP CPU pool. The CPUs in the pool provide I/O services to vPar guests. It handles incoming and outgoing I/O to all physical IO cards present on all running vPars system (plus do the work for the HP-UX instance running on the VSP itself, though this should be negligible). Except when cards are configured to use Direct I/O, the I/Os are handled directly inside vPars and DIO can be used to unload or transfer some tasks from the VSP CPU pool to vPars.

The required number of CPUs in the VSP pool varies based on the factors such as system size, number and type of I/O cards, number of vPars, vPars' load and so on. This prevents giving exact guidance that would fit everyone's need.

While there may be a perception that VSP CPUs represent an overhead which reduces the number of CPUs available to service the customer’s load inside vPars, this is not the case if the VSP and vPars are sized correctly. The CPUs in the VSP pool handle interrupts that the vPar itself would have to handle. It is therefore possible for the vPars to be sized smaller than would be required for the equivalent standalone systems running the same load, since some tasks are performed on their behalf by the VSP.

When a specific recommendation is requested, values may vary widely depending on the details of the environment. Typical ratios (of CPUs in the VSP pool vs CPUs assigned to vPars) would be 1:16 (rarely), 1:32 or 1:64, based on system load observed and HW configuration. When there are many 10Gbit LAN cards and heavy I/O load, the smaller ratio (that is, 1:16) is appropriate. While for computational type of load without heavy I/O the higher ratios (1:32 or even 1:64), or the default of one CPU, should be sufficient. In all cases, the recommendation still requires verification on customer's system under load.

Generally we recommend adding CPUs to the VSP pool, if CPU load on VSP pool CPUs stays above 50% for extended periods of time or when it frequently peaks above 80%. As suggested below, standard performance tools like Glance or top can be used to detect this. The load is likely to show high interrupt load (this could be coming from I/O cards as well as from interrupts that signal traffic from vPars) and also threads named "gParAvioIoThread" that HPVM driver uses to offload asynchronous tasks when they cannot be served directly from interrupt handler.

NOTE: VSP is really an appliance which happens to be HP-UX based and not a general-purpose HP-UX system. As such it should not be used for anything else, only for serving guests.

VSP CPUs can be found in either the VSP pool, or the Guest pool. When the first instance of a vPar is launched, one core is allocated to the VSP pool. By default, if a core is not allocated to either the VSP pool or a vPar, it will be found in the Guest pool.

As mentioned, the VSP pool CPUs handle interrupts from vPars (each vPar interrupts one of these CPUs and these interrupt assignments are rebalanced automatically when the number of CPUs in VSP CPU pool changes). They also handle I/O cards interrupts. The CPUs in the Guest Pool (either serving VM guests or idle) also handle I/O cards interrupts.

NOTE: However, for a system with many configuration changes, the I/O interrupts are not necessarily well balanced (that is, across all the CPUs visible by VSP), because by default HP-UX does not automatically rebalance I/O interrupts when the number of CPUs change (for example by stopping a vPar or adding a TiCAP CPU). For VSP with such dynamic usage or after a significant configuration change (like switching guest type from vPar to a VM, which makes more CPUs visible to VSP) it may be advisable to rebalance I/O interrupts by `intctl -b` command. Refer to its man page for details.

A single VSP core can service moderate to heavy I/O loads and vPars management requests. You can use performance tools such as `glance` and `top` to determine the CPU utilization of the VSP. When the VSP core becomes saturated, the response time of vPars commands and other applications on the VSP might increase. In such a situation, use the `hpvmhwmgmt` or `vparhwmgmt` command to add more cores to the VSP pool.

The sum of vPar cores, VM cores (when you consider the VCPU entitlements) and VSP cores cannot exceed the total number of cores on the system. While adjusting the VSP core count, if you exceed the system core count, and if the vPars and Integrity VMs are already configured, an error occurs. In such a situation, to meet the required core count for the VSP, first adjust the core count of one or more vPars and VM guests using the `hpvmmodify` or `vparmodify` command. Then, adjust the VSP CPU core count using the `hpvmhwmgmt` or `vparhwmgmt` command.

IMPORTANT: If the system is brought down due to a faulty CPU core and the cores are deconfigured, then the vPars and Integrity VMs might not boot during the subsequent boot of the VSP. This is possible if the sum of the remaining cores is less than the sum of the cores allocated to the VSP, vPars and Integrity VMs as displayed by the `hpvmhwmgmt` or `vparhwmgmt` command. Fix this by removing the cores from the vPars and Integrity VM or VSP, or even by reducing the overall CPU entitlement for one or more VMs to meet the configuration requirements.

Increased resources for Integrity VM guests

You can create 11iv3 Integrity VM guests with as many as 32 virtual processors (vCPUs) and up to 256 GB RAM. HP-UX 11iv2 VM guests, however, still have the old limit of 16 vCPUs and up to 128GB RAM.

vPars and Integrity VM pool

These cores are available for vPars and Integrity VMs. By default, all the cores on the VSP will be in this pool. As mentioned, when the first reserved vPars is configured or when the first non-reserved vPars is started, a single core is moved from this pool to the VSP pool. When reserved vPars are configured on the VSP, cores are reserved from this pool. When the vPars instances are initiated, the reserved cores are removed from this pool and assigned to the particular vPars.

Hyperthreading on the VSP

By default, VSP has the hyperthreading (firmware setting) set to ON in the npartition or server; and HP-UX kernel tunable `lcpu_attr` set to OFF in the VSP. This setting enables optimal performance and responsiveness of the VSP. DO NOT change the default hyperthreading settings in the VSP, unless it is recommended in the documentation.

NOTE: Hyperthreading is not supported for Integrity VM.

Hyperthreading is supported in individual vPars. To verify whether hyperthreading is enabled in an individual vPars, use the `setboot` command.

If hyperthreading is enabled, it shows that HT is ON.

To turn on `lcpu_attr` in an individual vPars, use the `kctune` command.

By default, `lcpu_attr` is OFF in the vPars.

NOTE: Even when `lcpu_attr` is OFF in the VSP, each vPars can have its individual `lcpu_attr` enabled to get hyperthreading functionality in the vPars.

VSP memory

On startup, the HP-UX vPars and Integrity VM product reserves a significant portion of the free system memory available on the VSP for the vPars and Integrity VM memory pool. This memory will be used for supporting the memory requirements of various vPars and VM guests on the VSP. The remaining available memory in the VSP is sufficient for the optimal functioning of the vPars and Integrity VM guests product on the VSP.

About 92% of free memory available at the vPars and Integrity VM product start time (after HP-UX has booted up on the VSP) is reserved for the vPars and Integrity VM memory pool. The amount of memory reserved also depends on the total system memory and the total number of system cores.

To view the allocation of memory, enter the following command:

```
# hpvmhwmgmt -p memory -l
```

The `/var/opt/hpvm/common/command.log` file also has information about the free memory available when the vPars and Integrity VM memory pool was allocated.

Only 64 MB or larger contiguous chunks of memory are reserved for the vPars and Integrity VM memory pool. Therefore, the system memory fragmentation at start time affects the amount of memory that can be reserved for the vPars and Integrity VM memory pool. If vPars and Integrity VM products are stopped and restarted, it is possible that there are not enough contiguous memory ranges in the VSP to match the memory, that was reserved for vPars and Integrity VM pool previously. This can lead to an over-commitment of the memory assigned to the Integrity VM or vPars.

NOTE: Hewlett Packard Enterprise strongly recommends that you restart the VSP when restarting the vPars and Integrity VM product, so that system memory fragmentation impact on vPars and Integrity VM memory pool size can be minimized.

Memory availability for VSP use can be controlled using the `HPVM_MEMORY_OVERHEAD_PERCENT` configuration variable. If this variable is set to an appropriate value in the `/etc/rc.config.d/hpvmconf` file, that value is used to determine the amount of memory reserved for vPars and Integrity VM in the memory pool. For example,

```
# ch_rc -a -p HPVM_MEMORY_OVERHEAD_PERCENT='N' /etc/rc.config.d/hpvmconf
```

If `HPVM_MEMORY_OVERHEAD_PERCENT` is set to 'N', then (100-N)% of free system memory available at vPars and Integrity VM product start time (after HP-UX has booted up on the VSP) is reserved for vPars and Integrity VM memory pool. The default setting is 8.

When determining the percentage, consider the following:

- Amount of memory in the system
- Number of guests
- Memory size of the guests you want to run

The higher the percentage, the less memory is available for guest usage.

NOTE: A VSP restart (or vPars and Integrity VM product restart) is required for this change to take effect. Hewlett Packard Enterprise strongly recommends that you do not use this configuration variable to change the memory available for VSP unless otherwise documented or recommended by Hewlett Packard Enterprise field personnel.

Memory overhead estimation

VSP requires certain amount of memory for the optimal functioning of the product. Given below is a rough estimate of the memory overhead required for the VSP.

The vPars and Integrity VM memory pool reserved is roughly about 92% of the system free memory available at the time of vPars and Integrity VM v6 product startup. The remaining memory is left out as free memory available for VSP use. This is in addition to the memory taken up by HP-UX to boot on the VSP. The memory used by HP-UX to boot depends on the size of the system, including total memory, number of cores, and the I/O devices on the system.

This equation indicates the following:

The overall VSP memory overhead = Amount of memory HP-UX requires to boot up + Free memory remaining in the VSP for optimal functioning of VSP

VSP memory overhead = ~1500 MB + 8.5% of total physical memory

To see how much memory is available for vPars and integrity VM memory pool size, enter the following command:

```
# hpvmhwmgmt -p memory -l
```

NOTE:

- The calculation for how much memory is in VSP versus what is available for vPars and VM guests is done at product start time.
- In addition to the VSP memory overhead, individual vPars and VM have a memory overhead depending on their size. For more information about memory, see “CPU and Memory” (page 49).

Reserving VSP devices

HPVM protects all the VSP system resources during the product start automatically, by marking them as restricted devices. This helps to protect storage and networking resources used by the VSP against unintended usage and corruption by vPars or VM guests. Any additional resources added to the VSP can be similarly protected against vPar or VM guest access.

The `hpvmdevmgmt` command allows you to mark the restricted devices.

Example 1 Example of restricting a device

You can reserve the disk storage on which the VSP operating system and swap space reside. This prevents guests from accessing the same disk storage devices.

For a sample device `/dev/rdisk/disk1`, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk1
```

To complete the restriction of volumes, each device included in the volume must also be restricted.

Configuring storage space for diagnostic data

It is necessary to provide sufficient storage space on the VSP to gather crucial diagnostic data, if problems are encountered. Table 7 (page 34) lists the major types of diagnostic data for which sufficient storage space must be allocated.

Table 7 Types of diagnostic data

Diagnostic data type	Storage location
Firmware diagnostic data for the VSP	`/var/tombstones/`
HP-UX system diagnostic data, which consists of several log files and crash-dumps	`/var/adm/crash/`

Table 7 Types of diagnostic data *(continued)*

Diagnostic data type	Storage location
HPVM Monitor Log file records diagnostic information from the Virtual Machine Monitor	`/var/opt/hpvm/common/hpvm_mon_log`
HPVM Monitor Dump files created when a guest encounters a fatal situation	`/var/opt/hpvm/guests/<Guest-Name>/vm.core`

The size of monitor dump file for any specific guest is roughly twice the value obtained from running the following command:

```
# hpvmstatus -P <GuestName> -V | egrep -i "Overhead memory"
```

VSP kernel tunables

Upon installation of vPars and Integrity VM product, tunables are modified to the values listed in Table 8 (page 35).

NOTE: The tunable values are set to enable optimal functioning of the product. Hence, DO NOT change any of these tunables unless otherwise specified by Hewlett Packard Enterprise.

Table 8 VSP kernel tunables

Tunable	Modified value
maxdsiz_64bit	34359738368
filecache_min	134217728
filecache_max	134217728
lockable_mem_pct	99%
base_pagesize	64
vx_ninode	131072
vxfs_ifree_timelag	-1
vxfs_bc_bufhwm	64000

Optionally, you can use expanded_node_host_names(5) tunable to activate the capability to set longer node and host names on the VSP. For more information about the instructions, see ***Node and Host Name sizes on HP-UX: Using the Expanded Capabilities***.

Running applications on VSP

Recommended applications

VSP is the manageability platform for vPars and Integrity VMs. Though VSP runs the standard HP-UX OE, it is a controlled environment, and customer applications must not be installed or run on the VSP. You can run applications on individual vPars and Integrity VM.

The VSP runs vPars and Integrity VM software, which is responsible for allocating processor and memory resources to the running guests. The VSP can run physical resource, performance, and software management and monitoring tools.

On the VSP, you can install and run the following software:

- Software installation tool Software Distributor-UX
- Hardware diagnostic and support tools to monitor guests (WBEM, online diagnostics, IRS (Insight Remote Support)
- HP Integrity Virtual Server Manager a GUI tool to manage VSP and guests

- System performance monitoring tools (GlancePlus, Measureware, OpenView Operations Agent)
- Utility pricing tools (Instant Capacity, Pay per use)
- Backup software like HPE Data Protector (client only)
- Hardware management tools (nPartition Manager, storage and network management tools)
- Multipath storage solutions
- HP Serviceguard (which can be run on HP-UX guests as well)

Applications not recommended

DO NOT run other applications on the VSP regardless of whether Integrity VM guests or vPars are running. Examples of applications that should not be run on the VSP are: Oracle, Workload Manager (WLM), HP SIM, and so forth. HP-UX vPars and Integrity VM v6 installation modifies kernel parameters, making the system unsuitable for running applications.

Hewlett Packard Enterprise also does not recommend configuring VSP as an Ignite UX server.

Applications specific recommendations

The following are the recommendations on running certain applications:

Backup solutions for VSP and virtual environment backups

Backup solutions such as HPE Data Protector or Veritas NetBackup can be used on both the VSP system and the vPars and Integrity VM systems. Consult the support matrix of such products for supported versions. Install the backup (client) agents on the VSP and the vPars and Integrity VMs. Hewlett Packard Enterprise highly recommends that the `/var` and `/opt` directories, in addition to the standard locations, be backed up regularly on the VSP system. Do not use the VSP system as a backup server. For more information, see *HP-UX 11i v3 Installation and Update Guide*.

HPE GlancePlus to monitor virtual environments

You can use Glance on the VSP to monitor vPars or VM data, but recorded measurements can be misleading. Glance receives the CPU accounting information from the vPars or VM kernel. Because the VSP can take the vPars or VM processor away (for example, when a hardware interrupt occurs), the time spent running other vPars or VMs is reported for the state that the vPars or VM was in at the time the CPU was taken away. For more information about using `Glance`, see *glance*(1M).

Glance 4.6 or later is supported running on a VSP or vPars or VM; however, certain measurements might be applicable in a particular context or report limited results. For example, measuring CPU utilization on the VSP reports all the time spent running in vPars or VMs as "system time"; to receive "user time" or "nice time" for a given vPars or VM, you must run Glance in those vPars or VM. Similarly, memory-related faults, or system calls for vPar or VM are not visible from Glance running in the VSP. Glance also offers a number of virtualization-related measurements. Note that Glance refers to virtual environments as logical systems.

HP Instant Capacity with Integrity VM guests

In an Integrity VM environment, Instant Capacity software provides meaningful functionality only on the VSP; it does not run on a VM (also known as a guest). In particular, Instant Capacity commands report an error if you attempt to run the commands on a VM guest. You can neither run a GiCAP Group Manager on a guest nor can specify a guest in the host list for a GiCAP group member.

In the case of vPar, Instant Capacity commands are supported on the VSP OS. However, on the vPar OS, you cannot execute Instant Capacity commands directly to activate or deactivate the cores. For an activation operation, first activate the cores on the VSP OS using the `icapmodify` command and then run the `vparmodify` command to complete the activation of the cores on the vPar OS. Similarly, for a deactivation operation, run the `vparmodify` command on the vPar OS and then run the `icapmodify` command on the VSP OS.

iCAP commands issued from the OA activate or deactivate cores only in the VSP. The `vparmodify` command must be run in the VSP to move the core to and from a vPar. If there is only one core in the VSP and the remaining cores are assigned to vPars, a deactivation request from the OA fails.

TiCAP is consumed after the core is active in either the VSP or vPar. If TiCAP is being used, to stop consuming TiCAP, you must deactivate the core from the vPar and the VSP.

4 Upgrading the VSP from earlier versions of Integrity VM

This chapter describes how to upgrade the VSP from an older version. You must know the following before upgrading to a newer version:

- vPars and Integrity VM v6 supports guests running HP-UX 11i v3 and HP-UX 11i v2 (starting from v6.1.5). It does not support OpenVMS guests.
- Integrity VM v4.3 and earlier versions used VIO interfaces and Legacy DSF's for mass storage. These are not supported on vPars and Integrity VM v6 and later releases.
  - You can use the `hpvmmodify` command to convert VIO interfaces to AVIO.
  - You can use the `hpvmdevtranslate` command to convert Legacy DSF files to Agile DSF.

NOTE: If you are upgrading from earlier versions of vPars (A5.x), see the detailed upgrade procedure documented in *Realize new workload migration and consolidation possibilities* white paper at **http://www.hpe.com/info/hpux-hpvm-docs/**.

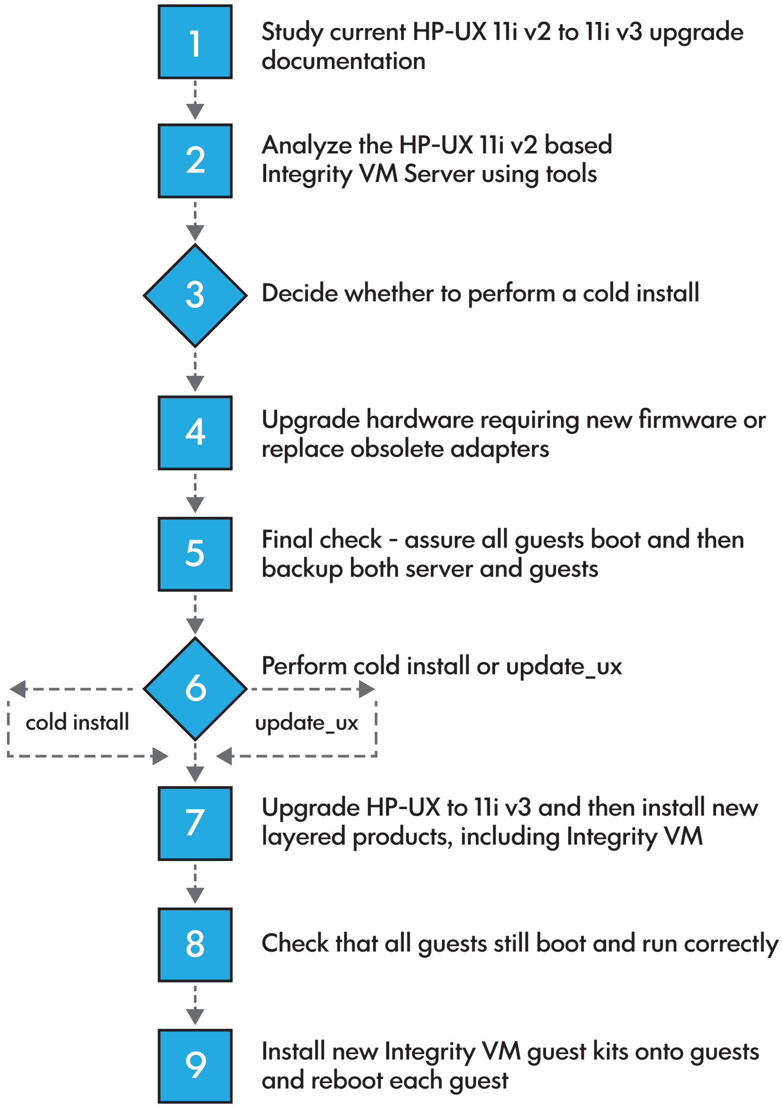
Upgrading VSP from Integrity VM v3.x to vPars and Integrity VM v6.4

The vPars and Integrity VM software requires the VSP to be running HP-UX 11i v3 operating system. Only HP-UX 11i v2 servers running Integrity VM Version 3.0 or Version 3.5 can be upgraded to the HP-UX 11i v3 vPars and Integrity VM Version 6.4 release. If you are upgrading the VSP from Integrity VM v4.0 or later to vPars and Integrity VM v6.4, see “Upgrading earlier versions of the VSP and VM guests to vPars and Integrity VM v6.4” (page 45).

HP-UX 11i v3 supports many features that are backward compatible with 11i v2, allowing 11i v2 applications to run without modifications. The primary aim of this section is to provide direction to the administrator performing the upgrade of the VSP to ensure that all configured VMs (guests) boot and run after completing the upgrade to 11i v3.

Figure 4 (page 39) shows a flowchart of the upgrade procedure from 11i v2 to 11i v3.

Figure 4 Upgrade procedure



Firstly, the administrator must identify subsystems on the 11i v2 Integrity VM server that are incompatible with or that are not supported on 11i v3. Some incompatibility issues can be exposed by tools, and others are found in referenced documents. The most common update problems are caused by the following:

- Unsupported hardware adapters or firmware.
- Memory and system disk space requirements (HP-UX 11i v3 has increased both of these.).
- Obsolete or unsupported storage multipath solutions.
- Layered products requiring an 11i v3 compatible version.

Studying the current HP-UX 11i v2 to HP-UX 11i v3 update documentation

The first stage of upgrading an Integrity VM v3.0 or v3.5 server to vPars and Integrity VM v6.4 server is to review the following HP-UX 11i v3 operating system update documents:

- HP-UX 11i v2 to 11i v3 Mass Storage Stack Update Guide available at **http://www.hpe.com/info/hpux-core-docs-11iv3**
- Read Before Installing or Updating Guide available at **http://www.hpe.com/info/hpux-core-docs-11iv3**

- HP-UX 11i v3 Installation and Update Guide available at **http://www.hpe.com/info/hpux-core-docs-11iv3**
- HP-UX 11i Version 3 Release Notes available at **http://www.hpe.com/info/hpux-core-docs-11iv3**
- Serviceguard Specific Documentation available at **http://www.hpe.com/info/hpux-serviceguard-docs**

For a general reference covering the features and hardware supported in HP-UX 11i v3, and to become familiar with the information before starting the upgrade, read the quickspecs document available at **http://www.hpe.com/info/quickspecs**.

Analyzing HP-UX 11i v2 based Integrity VM server

Analyzing HP-UX 11i v2 based Integrity VM server is the most important stage of the Integrity VM server upgrade. During this analysis, it is important to discover incompatible hardware and software subsystems, if any.

HP-UX 11i v3 uses a mass storage model called the agile device reference model, for naming and identifying devices. The 11i v2 model is called the legacy device reference model. The agile device model uses worldwide device identifiers (WWIDs) to identify devices. The WWID is a device attribute that is independent of the location of the device in a SAN or in an adapter or controller access path. Therefore, the agile device names are persistent with respect to changes in the access path, and can utilize multiple paths through a single device name.

The legacy devices require multiple device names to access the same device through multiple paths. Many Integrity VM customers use multipath solutions such as SecurePath, LVM PV-link, which allows them to use a single device name to access all paths. Some of these 11i v2 multipath solutions continue to work, while others must be removed. After the upgrade is completed, replace the existing multipath device with the agile device name, with the inherent multipath support.

NOTE: If you are using third party multipathing solution (EMC PowerPath or similar), it is necessary that it supports agile DSF. For more information about the support, see the appropriate vendor documentation.

Analyze each layered product to determine the upgrade impact:

- No change - Layered product is compatible.
- Delete or reinstall - Layered product requires a new version to work on 11i v3.
- Delay upgrade – Layered product needs a new version that is not yet released.

For more information about the layered product, see the HP-UX 11i v3 documentation available at **http://www.hpe.com/info/hpux-core-docs-11iv3**.

Running the HP-UX msv2v3check tool

The HP-UX `msv2v3check` command reviews all mass storage controllers and devices on your system for HP-UX 11i v3 compatibility and support. The `msv2v3check` tool is free software available on the **http://www.hpe.com/support/softwaredepot** website. Go to this website, search for msv2v3check, and download this free tool.

The `msv2v3check` command examines only mass storage controllers (host bus adapters) and devices for HP-UX 11i v3 compatibility and support. This includes the following:

- Ultra160 SCSI (C8xx) host bus adapters and attached HPE supported SCSI devices.
- Ultra320 SCSI (MPT) host bus adapters and attached HPE supported SCSI devices.
- Serial Attached SCSI (SAS) host bus adapters and attached HPE supported SAS devices.
- Smart Array RAID (CISS) host bus adapters and attached HPE supported RAID devices.

- Fibre Channel (FCD/TD) host bus adapters and attached HPE supported Fibre Channel devices.
- HPE supported SCSI disk enclosures and arrays.
- HPE supported Fibre Channel disk enclosures and arrays.

The `msv2v3check` command creates the log file `/var/adm/msv2v3check/mmddyy_hhmm` that contains all notes, warnings, and error messages from an invocation of `msv2v3check`, where `mmddyy_hhmm` represents the month, day, year, hour, and minute the `msv2v3check` utility is started.

After the `msv2v3check` utility is completed, a validation result is displayed that indicates the number of errors and warnings detected on your system configuration:

- An error is a critical message that indicates that your system does not support HP-UX 11i v3 in its current configuration. Do not ignore this message.
- A warning indicates a task that might require user action, for example, upgrading the firmware on a disk device, or manually reviewing the firmware of a Fibre Channel disk array.

Review all warnings and make the necessary corrections before upgrading to HP-UX 11i v3.

For more information about supported I/O drivers, devices, adapters, see the documentation available at **HPE Manuals**.

Determining HP-UX 11i v3 memory and system disk requirements

The memory requirements of vPars and Integrity VM vary depending on the number and size of VMs supported by the Integrity VM server. When upgrading from an 11i v2 Integrity VM server, carry out the following steps to determine the amount of memory required for the 11i v3 Integrity VM server:

1. When your 11i v2 Integrity VM server is running at peak load, use the Integrity VM `hpvmstatus -s` command to find out the available memory.
2. If the available memory is less than 1 GB, then the server may require additional memory to run the same load with 11i v3 and vPars and Integrity VM v6.4. Before upgrading, add the appropriate amount of memory to ensure that there is at least 1 GB of memory available during peak load.

NOTE: Different operating environments have different minimum memory requirements.

Determining version requirements for HP-UX OE and vPars and Integrity VM

When upgrading from an earlier release, support for specific guest types or backing stores might have been removed. For support information, see *HP-UX vPars and Integrity VM v6.4 Release Notes* at **http://www.hpe.com/info/hpux-hpvm-docs**.

Table 9 (page 41) lists the HP-UX 11i v2 to HP-UX 11i v3 supported OE server upgrades.

Table 9 Supported operating environments

Original 11i v2 operating environments	New 11i v3 operating environments
Foundation OE	Base OE
Technical Computing OE	Base OE
Enterprise OE	Virtual Server OE
Mission Critical OE	Data Center OE

For more information about HP-UX OE, see “Introduction” (page 11).

NOTE: Many software subsystems require upgrades on the 11i v2 Integrity VM server before upgrading to HP-UX 11i v3. Integrity VM must be upgraded to v3.0 or v3.5 before beginning the HP-UX upgrade. Other layered products, such as Serviceguard, must be upgraded before upgrading the operating system to 11i v3. Analyze each layered product for the required upgrades.

Remove the older HP Integrity Virtual Machines Manager product before upgrading to vPars and Integrity VM Version 6.4. After installing vPars and Integrity VM v6.4, install the latest Integrity Virtual Server Manager and HP-UX GUID Manager products. If you are upgrading an Integrity VSP from 11i v2 to 11i v3 and are using Veritas file systems and volumes, update to Veritas v5.0 and become familiar with the Veritas 5.0 installation guide available at **http://www.hpe.com/info/hpux-LVM-VxVM-docs**.

Deciding whether to perform a cold-install or an update

The preferred method for upgrading an HP-UX 11i v2 based Integrity VSP to an 11i v3 based VSP is to use the `Update-UX` program. The `update-ux` command takes as input the 11i v3 OE depot. The `update-ux` command strives to maintain all your current user, storage, and network configurations. There are some 11i v2 multipath solutions that are not compatible with 11i v3. In most cases, the multipath conversion is to use the agile devices on 11i v3 in place of the device names that the multipath solutions invented. The `Update-UX` program also strives to retain the volume definitions. This is helpful, because a cold-install most likely changes all the device names requiring a mapping of devices to volumes and to guests.

To choose a cold-install over an `update-ux` update is the ease with which you can immediately return to the 11i v2 environment. The `update-ux` path changes the original 11i v2 system configuration making a restore from backups the only way to return to the original 11i v2 system. The cold-install can and must be given separate disks to use allowing the original 11i v2 system disks to remain unchanged. Because the original disks can remain unchanged, the need to back up the 11i v2 based Integrity VSP is minimal.

NOTE: Hewlett Packard Enterprise recommends a complete back up of both the Integrity VSP and guests before updating.

Whether you choose `update-ux` or a cold-install upgrade, as the administrator you must study the documentation that covers the differences between HP-UX 11i v2 and HP-UX 11i v3. To obtain information about potential upgrade problems, you must also run the HP-UX `msv2v3check` tool.

Upgrading required hardware and firmware upgrades

While still running on 11i v2, perform all the hardware and firmware upgrades that are supported on 11i v2 and that are needed for 11i v3. This allows the administrator to verify that all the guests are fully functional with the changes before upgrading to 11i v3.

Performing a cold-install or an update

If you choose the cold-install upgrade path, it means the administrator is taking the responsibility for fully configuring the 11i v3 Integrity VSP to be functionally equivalent to the 11i v2 Integrity VSP configuration. vPars and Integrity VM v6.4 provides the `hpvmdevtranslate` utility to assist in mapping the legacy devices used by guests on the 11i v2 VSP to the new 11i v3 agile devices.

The `hpvmdevtranslate` utility produces the script `/var/opt/hpvm/common/hpvm_dev_convert`. This script must be reviewed and edited before running it to make the conversions. Device conversions that cannot be made are listed as comments labeled `ERROR:`. The administrator is responsible for determining the conversion of the ERROR lines. The `hpvmdevtranslate` utility translates only devices that provide unique WWIDs.

After evaluating your 11i v2 Integrity VSP and performing appropriate backups, carry out the following steps with the `hpvmdevtranslate` utility as part of a cold-install:

1. Choose the system disks that are to be used for the 11i v3 VSP and mark them as reserved disks:

   `# hpvmdevmgmt -a rdev:device_name`

2. Back up and collect all relevant configuration from the 11i v2 VSP.
3. Back up the `/var/opt/hpvm` directory, so that you can easily restore it to the 11i v3 system after the cold-install.

   NOTE: DRD can be used to clone an HP-UX system image to an inactive disk for recovery. For information about DRD, see the dynamic root disk documentation available at **http://www.hpe.com/info/drd-docs**.

4. Verify that all current guests that run on 11i v2 can boot and run successfully. Guests that cannot boot on 11i v2 cannot be expected to boot after the upgrade to 11i v3.
5. After verifying the guests, back up all relevant configuration data for each guest for a potential return to 11i v2.
6. Shut down the Integrity VM guests gracefully by logging into each one and shutting it down.
7. Shut down the Integrity VSP.
8. Using the HP-UX cold-install procedure, install the appropriate 11i v3 OE using the selected system disks. For information about performing a cold-install, see *HP-UX 11i v3 Installation and Update Guide*.
9. Remove any blocking layered products that might block the Integrity VM installation. See "Installing vPars and Integrity VM" (page 22) for a list products.
10. Remove layered products that might cause problems or that require a new 11i v3 compatible version after the HP-UX 11i v3 upgrade.
11. Determine the order of installation of layered products, including vPars and Integrity VM v6.4 (BB068AA), so that all dependencies are met. For example, if VERITAS is used to provide backing storage for guests, install it before Integrity VM.
12. Install all 11i v3 compatible layered products that are required for equivalent functionality to the 11i v2 VSP.
13. Install vPars and Integrity VM Version 6.4 on the 11i v3 VSP. For more information about installing vPars and Integrity VM, see "Installing HP-UX vPars and Integrity VM" (page 22).
14. Stop Integrity VM using `/sbin/init.d/hpvm stop`.
15. Using the appropriate recovery tool, restore the 11i v2 `/var/opt/hpvm` directory over the existing 11i v3 `/var/opt/hpvm` directory on the 11i v3 VSP.
16. Start vPars and Integrity VM using `/sbin/init.d/hpvm start`.
17. Run the translator:

   `# hpvmdevtranslate -a /var/opt/hpvm/common/hpvm_mgmtdb_pre1131`

18. Edit the script, `/var/opt/hpvm/common/hpvm_dev_convert`, taking note of ERROR lines and commenting out the exit line that prevents the running of the script.
19. Continue with the remaining 11i v3 Integrity VSP configuration until the host is functionally equivalent to the former 11i v2 Integrity VSP.

If you choose the update path:

1. Create a recovery image.
2. Verify that all the current guests that run on 11i v2 can boot and run successfully. Guests that cannot boot on 11i v2 cannot be expected to boot after the update to 11i v3.
3. After verifying the guests, back up all relevant configuration data for each guest for a potential return to 11i v2.
4. Install the latest `Update-UX` bundle from the OE media.

5. Update the OS or OE from the HP-UX 11i v3 OE media using the `update-ux` command. For example:

   ```
   # swinstall -s /dev/dvd Update-Ux
   update-ux -s /dev/dvd HPUX11i-VSE-OE BB068AA
   ```

   NOTE: There is an `update-ux` option, `-p`, which can be used to preview and update task by first running the session through the analysis phase.

   If you are updating from the VSE-OE depot, specify the following:

   ```
   # swinstall -s my.server.example.com:/OEdepot/path Update-UX
   update-ux -s my.server.example.com:/OEdepot/path HPUX11i-VSE-OE BB068AA
   ```

6. Remove any blocking layered products that might block the Integrity VM installation. See “Installing vPars and Integrity VM” (page 22) for a list of products.
7. Remove layered products that might cause problems or that require a new 11i v3 compatible version after the HP-UX 11i v3 update.
8. Determine the order of installation of layered products, including vPars and Integrity VM v6.4 (BB068AA), so that all dependencies are met. For example, if VERITAS is used to provide backing storage for guests, install it before Integrity VM.
9. Install vPars and Integrity VM Version 6.4 on the 11i v3 VSP.
10. Update non-OE applications from the Application media using the `swinstall` command.

    For example, if you plan to install Integrity Virtual Server Manager, switch to the AR disk and specify the following:

    ```
    # swinstall -s my.server.example.com:/Ardepot/path VMMGR
    ```

11. Create the recovery image.

Verifying vPars or VM after installing layered products

Follow these steps after installing layered products:

1. Start and stop each guest, one at a time, and ensure that they boot to their OS.
2. To resolve guest booting problems, see the guest troubleshooting section, “Appendix B” (page 301).
3. Upgrade each guest with the new guest kit.
4. Ensure there are no network issues.
5. If the guest OS is no longer supported, upgrade the guest OS.

NOTE: When Integrity VM is stopped either with the `/sbin/init.d/hpvm stop` command or as a result of removing or updating the version of Integrity VM on the VSP, messages of the following form might be logged in the `/var/opt/hpvm/common/command.log` file:

```
ERROR|host|root|Unable to communicate with the FSS agent
```

The messages, which are a result of interactions with the performance metrics processes `scopeux` and `perfd`, are normally transient and stop after about a minute. Approximately 60-70 messages might be generated in that time. You can clear this condition by either rebooting the VSP or by stopping and restarting the metrics collection processes.

To stop and restart the `perfd` process, use the following commands:

```
# /sbin/init.d/pctl stop
# /sbin/init.d/pctl start
```

To stop and restart the `scopeux` process, use the following commands:

```
# /sbin/init.d/ovpa stop
# /sbin/init.d/ovpa start
```

Troubleshooting upgrade issues

After you upgrade to 11i v3, examine the following issues:

- Mass storage issues

  The vPars and Integrity VM v6.4 release supports the use of both legacy and agile devices within guests. It is not necessary to convert guests to use strictly agile devices. If, however, problems occur with guests using multipath solutions that are based on legacy devices, change the backing device to use the equivalent agile device. For information about mass storage compatibility issues, see the documentation available at:

  HP-UX 11i v3 Manuals.

- Platform issues

  For 11i v3 platform support, see the following matrix:

  HP-UX Integrity Server Support Matrix

- Serviceguard issues

  For information about the Storage Multi-Pathing choices in HP-UX Serviceguard environments, see the Serviceguard website:

  HP Serviceguard Solutions

Upgrading earlier versions of the VSP and VM guests to vPars and Integrity VM v6.4

This section describes the process of updating an earlier version of the VSP to vPars and Integrity VM v6.4.

For example, to update the VSP and VM guests from v4.3 to v6.4:

1. Migrate VM guests to an alternate VSP or perform an orderly shutdown of all Integrity VM guests on the v4.3 VSP.

   If you have a VSP established as an OVMM target host, Hewlett Packard Enterprise recommends that you migrate the VM guests to that VSP. If OVMM target VSP does not exist, perform an orderly shutdown of the VM guests. For example, perform one of the following steps:

   - Migrate an existing VM guests to an alternate VSP (v4.3, v6.1, v6.2, v6.3, or v6.3.5):

     ```
     VSP -> hpvmmigrate -P VM name  -o -h Target VSP
     ```

   - Perform an orderly shutdown of all Integrity VM guests on the v4.3 VSP:

     ```
     VM -> shutdown -h -y 0

     SHUTDOWN PROGRAM
     12/17/12 09:51:23 PDT

     Broadcast Message from root (ttyp1) Mon Dec 17 09:51:23...
     SYSTEM BEING BROUGHT DOWN NOW!!!

     ...
     ```

2. Mount DVD HP-UX 11i v3 March 2016 ISO Image or locate the March 2016 Depot Server:

   ```
   VSP ->  kcmodule fspd=unused
   VSP -> kcmodule fspd=loaded
   VSP ->  mount /tmp/HP-UX_11i_v3_DC-OE_Core_1_2actualDVDname.iso /dvdrom
   VSP -> bdf
   Filesystem          kbytes   used   avail %used Mounted on
   /dev/vg00/lvol3   2097152 231152 1851488  11% /
   /dev/vg00/lvol1   2097152 371040 1712696  18% /stand
   /dev/vg00/lvol8   10485760 1449064 8973992  14% /var
   ```

```
/dev/vg00/lvol7   10485760 3075176 7352720  29% /usr
/dev/vg00/lvol6   20971520 11325968 9570272  54% /tmp
/dev/vg00/lvol5   10485760 5039496 5403832  48% /opt
/dev/vg00/lvol4   10485760  21152 10382856   0% /home
/dev/fspd1     75359147535914   0 100%/dvdrom
```

3. Run the `update-ux` command on the v4.3 VSP:

```
VSP -> update-ux -s /dvdrom
======= Mon Dec 17 21:14:05 PDT 2012 BEGIN update-ux

NOTE:   Output is logged to '/var/adm/sw/update-ux.log'
  * Obtaining some information from the source depot.
  * Copying an SD agent from the source depot
  * Installing the Update-UX product
  Current update-ux version: 11.31.22
  Source depot update-ux version: 11.31.22
  * Running the new version of update-ux
  * Installing the SW-GETTOOLS product
  * Configuring the SW-GETTOOLS product
  * Installing the SD filesets to be used for the update
  * Installing the SWM filesets needed to perform OE update
NOTE:   Running swm

...
```

4. Verify Integrity VM software after `update-ux` on the VSP:

```
VSP -> swlist -l product | grep -i B.06.40
HPVM              B.06.40  Integrity VM
VMAGENT           B.06.40  HP Resource Allocation Agent for Integrity VM
vmGuestLib        B.06.40  Integrity VM vmGuestLib
vmGuestSW         B.06.40  Integrity VM vmGuestSW
vmKernel          B.06.40  Integrity VM vmKernel
vmProvider        B.06.40  WBEM Provider for Integrity VM vmProvider
vmVirtProvider    B.06.40  Integrity VM vmVirtProvider
VSP -> swlist | grep -i B.06.40
BB068AA         B.06.40      HP-UX vPars & Integrity VM v6
VirtualBase     B.06.40     Base Virtualization Software
VSP -> swlist -l product | grep -i avio
AVIO-GVSD   B.11.31.1603  HPVM Guest AVIO Storage
AVIO-HSSN   B.11.31.1603  HP AVIO LAN HSSN Host Driver
AVIO-HVSD   B.11.31.1603  HPVM Host AVIO Storage Software
AVIO-IGSSN    B.11.31.1603  HP AVIO LAN IGSSN Guest Ethernet Driver
```

NOTE: The `what` string output of HPVM product will have the version as `HPVM_B.06.40`.

5. Boot and update the VM guest software after updating the VSP:

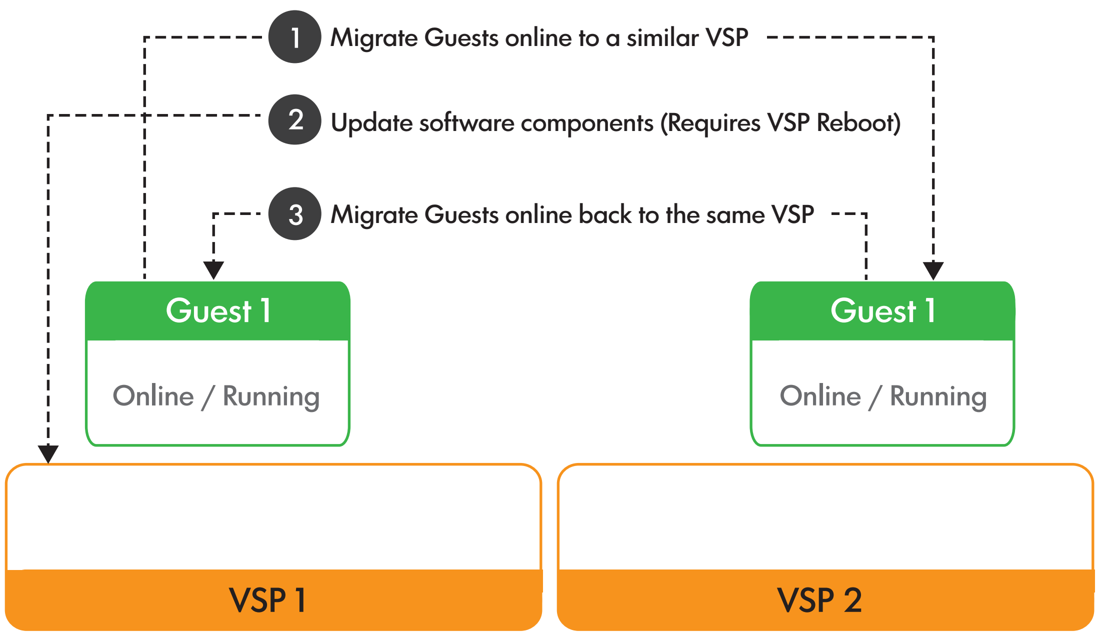
```
VM -> hpvminfo -S
HPVM Guest information
Version: HPVM B.06.40 LR ccipf opt Wed Nov 25 2015 14h08m47s IST
My partition ident: 722ce8ce-e118-11e0-9210-d8d3856a822a
Server partition ident: 5a8cc5cd-4096-11df-837f-1bece9967508
Server hostname: abc15.domain.com
Server physical ident: 5a8cc5cd-4096-11df-837f-1bece9967508
VM -> scp VSP:/opt/hpvm/guest-images/hpux/11iv3/hpvm_guest_depot.11iv3.sd /tmp/.
hpvm_guest_depot.11iv3.sd  100%  13MB 13.4MB/s 13.4MB/s  00:01
VM -> swinstall -x autoreboot=true -s /tmp/hpvm_guest_depot.11iv3.sd \*
* Software selections:
   VirtualBase,r=B.06.40,a=HP-UX_B.11.31_IA,v=HP
   AVIO-GVSD.GVSD-KRN,r=B.11.31.1603,a=HP-UX_B.11.31_IA,v=HP,fr=B.11.31.1603,fa=HP-UX_B.11.31_IA
   AVIO-GVSD.GVSD-RUN,r=B.11.31.1603,a=HP-UX_B.11.31_IA,v=HP,fr=B.11.31.1603,fa=HP-UX_B.11.31_IA
   AVIO-IGSSN.IGSSN-KRN,r=B.11.31.1603,a=HP-UX_B.11.31_IA,v=HP,fr=B.11.31.1603,fa=HP-UX_B.11.31_IA
   AVIO-IGSSN.IGSSN-RUN,r=B.11.31.1603,a=HP-UX_B.11.31_IA,v=HP,fr=B.11.31.1603,fa=HP-UX_B.11.31_IA
   vmGuestLib.GUEST-LIB,r=B.06.40,a=HP-UX_B.11.31_IA,v=HP,fr=B.06.40,fa=HP-UX_B.11.31_IA
   vmProvider.VM-PROV-CORE,r=B.06.40,a=HP-UX_B.11.31_IA,v=HP,fr=B.06.40,fa=HP-UX_B.11.31_IA
 * Selection succeeded.
VM -> hpvminfo -S
HPVM Guest Information
Version: HPVM B.06.40 LR ccipf opt Wed Nov 25 2015 14h08m47s IST
My partition ident: dabe1bea-0237-11e4-a20c-0017a4770010
Server partition ident: 40ad6216-339b-11e0-be85-eab7c570a466
Server hostname: xyz.domain.com
Server physical ident: 40ad6216-339b-11e0-be85-eab7c570a466
```

```
VSP - —> hpvmstatus
[Virtual Machines]
Virtual Machine Name VM# Type OS Type State #VCPUs #Devs #Nets Memory
==================== === ==== ======= =====  ===== ===== ===== ======
guestName             3   SH   HPUX    On(OS)  1     1     0    2048 MB
hpvmstatus: WARNING (test): The configuration file must be updated.
VSP —> hpvmmodify -F -P guestName
```

NOTE: Similar set of steps can be followed to upgrade 6.x version of the product to the latest version.

Figure 5 (page 47) illustrates the steps to be followed to upgrade 6.x version.

Figure 5 Upgrading a VSP using the Online Guest Migration process



Rolling back to the earlier installed version of Integrity VM

If you must roll back to a previous version of Integrity VM, this section provides the information needed to perform the rollback. The preferred method for rolling back to a previously installed version of Integrity VM is to restore the system image that was backed up before installing the current version of Integrity VM on the VSP. Because this is not always possible for all users the following method must work.

The VSP and guest configuration files are stored at `/var/opt/hpvm`. Because configuration files for newer versions of Integrity VM are not normally compatible for earlier versions of Integrity VM, a copy is made of the contents of `/var/opt/hpvm` to the `/var/opt/hpvm/backup` directory (except the ./guest-images and ./backups directories). If it is required, it is possible to revert to the earlier version of Integrity VM using the backups directory and the following process:

1. Ensure you have the installation media for the version of Integrity VM that was installed before v6.4.
2. Before you stop Integrity VM, ensure all guest types are of same type either all VMs or all vPars.

   NOTE: This is applicable only if you are rolling back to a version prior to 6.2.

3. Stop Integrity VM (`/sbin/init.d/hpvm stop`).
4. Remove Integrity VM v6.4 software (This causes a system reboot).

   ```
   # swremove -x autoreboot=true BB068AA
   ```

5. Move the `/var/opt/hpvm` area aside:

   ```
   # mv /var/opt/hpvm /var/opt/hpvm_6.4
   ```

6. Install the earlier installed version of Integrity VM following the directions for installing Integrity VM in this manual for that version. This also causes a system reboot.
7. After the system is back up, log in, and stop Integrity VM (`/sbin/init.d/hpvm stop`).
8. Restore the earlier Integrity VM environment:

   ```
   # cd /var/opt/hpvm_4.3/backups; tar -cpf - | cd /var/opt/hpvm; tar -xpf -
   ```

9. Start Integrity VM.

5 CPU and Memory

Configuring CPU resources for VM guests

Processor virtualization

VM guests are configured with virtual processors. A vCPU is a virtualized schedulable entity. Virtual processors are mapped to physical CPU, cores as a part of VM guest scheduling. For the purpose of this discussion, the term “physical CPU” refers to a processing entity on which a software thread can be scheduled. Each vCPU is independently scheduled as a single thread of execution on a physical CPU, subject to the entitlements discussed in “vCPU entitlements” (page 49). The scheduling of all vCPUs belonging to a guest and across guests is independent of each other. This helps in maximizing the utilization of physical CPU resource across many vCPUs belonging to different VM guests.

Each VM guest has at least one vCPU. Use the `hpvmcreate -c number_vcpus` command to specify the number of virtual CPUs that the VM guest can use. The maximum vCPU count that can be set for a VM guest is 32. If you do not specify the number of vCPUs, the default is 1.

For example, to set the new VM guest `vmguest1` to have two vCPUs, enter the following command:

```
# hpvmcreate -P vmguest1 -c 2
```

A running VM guest cannot use more vCPUs than the number of physical CPU cores on the VSP system. Do not set the number of vCPUs higher than the physical number of CPU cores, as this can prevent the VM guest from starting.

You can change the number of enabled CPUs in VM guests running HP-UX, using the `hpvmmgmt -c num` command from within the guest OS. This command sets the number of enabled virtual CPUs to the number indicated by `num` (up to the number of CPUs the guest is booted with), and disables the others. Disabled virtual CPUs do not show up in the guest when you run commands such as `top` or `GlancePlus`, and do not consume resources on the VSP. However, disabled virtual CPUs still appear on the VSP, for example when you run the `hpvmsar` command.

NOTE: HP Integrity VM does not support running real-time applications on the guest. Scheduling and precise timing properties that can be relied upon on physical hardware are not guaranteed to be preserved in a VM guest. In particular, changing the `hires_timeout_enable(5)` HP-UX tunable might not have the desired effect.

---

vCPU entitlements

Entitlement is the amount of processing power guaranteed to VM guest for each virtual CPU. When you create a VM guest, you can use the `hpvmcreate -e`command to specify the entitlement as a percentage, a value between 5% and 100%. If you do not specify the entitlement, the VM guest receives 10% minimum entitlement and 100% maximum entitlement by default.

The minimum entitlement of a VM guest means that each VM guest vCPU is guaranteed at least the specified % CPU of the physical CPU on which it is running (the physical CPU processing power left after all the interruptions have been serviced).

Similarly, the maximum entitlement of a VM guest means that each VM vCPU can use the specified maximum % CPU of the physical CPU on which it is running. It is recommended to keep the maximum entitlement of each vCPU at default that is 100% because lower values can have performance implications.

When the VM guest starts, the VSP ensures that minimum % CPU is available for every running VM to receive its entitlement. If sufficient physical CPU resources are available on the VSP system, a VM guest can receive more processing power than its minimum entitlement and can go to a maximum of 100%. When there is contention, each VM (or rather vCPU) is proportionally limited in such a way that entitlements for all VMs (vCPUs) that want to use their share are satisfied.

For VM guest with multiple virtual CPUs, the entitlement is guaranteed on each vCPU in the VM's configuration. For example, if a VM guest has four vCPUs, and the entitlement is set at 12%, the VSP ensures that the equivalent of at least 48% of one physical CPU is available to that guest. The vCPUs are distributed in such a way that each vCPU of each guest runs on a different physical CPU.

Based on the availability of physical CPU's processing power and memory resources, NUMA-aware Resource Allocator algorithm finds physical CPUs possibly closer to each other to associate vCPUs of VM guest and binds each vCPU of it to a unique physical CPU. So, we avoid one vCPU without physical CPU processing power because of other vCPU bound to the same physical CPU.

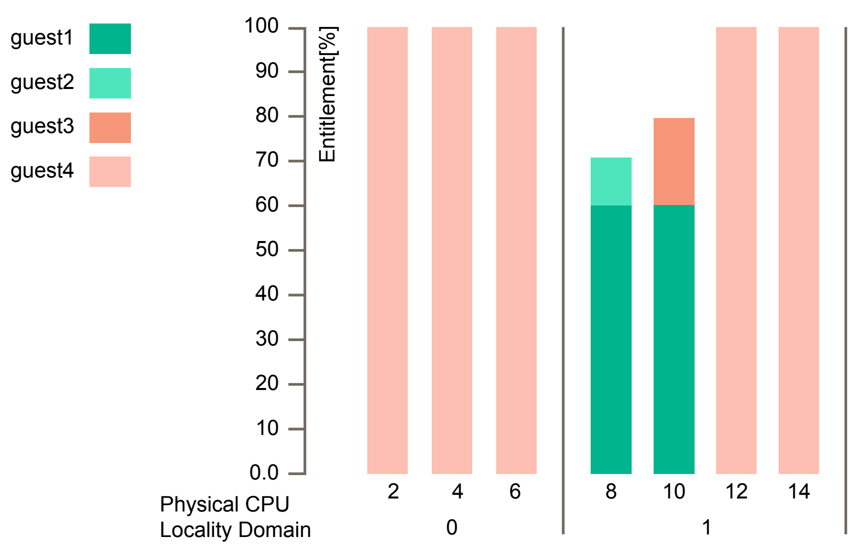
Ensure that the entitlement of each VM guest does not prevent the other VMs from obtaining sufficient processor resources to allow multiple VM guests to run at the same time. The sum of all entitlements across all active VM guests cannot be more than 100% for any physical processor.

To illustrate the above, assume the following system configuration details:

- VSP has 8 physical cores.
- User runs Guest1 with 2 vcpus at 60% minimum.
- User runs Guest2 with 1 vcpu at 10% minimum.
- User runs Guest3 with 1 vcpu at 20% minimum.
- User runs Guest4 with 5 vcpus as a VPAR guest.

Following Figure 6 captures the physical cores vs % entitlement usage for each of the four guests as per the entitlements described above. In this example, CPU 0 is assigned to VSP pool and it is not shown as it is not assigned to any of the four guests running.

Figure 6 Entitlements vs vCPU



Dynamically changing the entitlements

While you cannot add or remove CPUs to and from a VM guest dynamically, you can change the vCPU entitlement of the vCPUs that are already configured. You can use the `hpvmmodify` command to change the entitlement.

Transforming VM guest to a vPar

For better guest performance, you can transform a VM guest offline to a vPar. Use the `hpvmmodify -x vm_type=vpar` command to transform a VM guest to a vPar. For more information about transforming VM guest to a vPar, see “Transformation between VM and vPar” (page 243).

Hyperthreading for VM guest

Hyperthreading is not supported for VMs. Therefore, individual VMs will not show any hyperthreading capability. Even if hyperthreading and the `lcpu_attr` tunable are turned ON in the VSP, the number of vCPUs in a VM cannot be more than the number of physical CPU-cores on the system.

MCAs on VM guests

MCA’s (Machine Check Abort's) are the highest priority interruptions among a class of Itanium processor interruptions.

They indicate an unexpected hardware condition where one or more processors need immediate intervention to normal operation. Based on the scope and severity of the problem MCA's are categorized into several categories.

- Scope of an MCA
    - When observed problems can be isolated to a single processor, the MCA is categorized as local. In some situations, it is possible for multiple processors to encounter local MCA's simultaneously.
    - MCA’s caused by problems which affect the entire system are termed as Global MCA’s.
- Severity of an MCA

  When an MCA is encountered, Itanium processor hardware, system hardware, and system software work together to isolate and if possible, correct the error so that the normal operations can be resumed. Based on the success of this operation, MCA’s are termed as
    - Recoverable

      The faulty code is either corrected or terminated; system operation is resumed.

      OR
    - Non-recoverable

      Normal system operation cannot be safely continued; further actions depend on the exact nature of the problem. The system will be rebooted.

All the four combinations of these are possible.

Integrity VMs work with emulated virtual Itanium processors (vCPUs). Consequently, all error conditions that such virtual processors encounter are handled by the Virtual Machine Monitor.

MCAs encountered by emulated vCPUs are always categorized as non-recoverable; the VM is rebooted and necessary log files are generated.

If such a problem is encountered, you must gather diagnostic data and contact HPE Support. For more information, see “Appendix B” (page 301).

Configuring CPU resources for vPars

The CPU resource configured for vPar is the physical CPU on the VSP. The physical CPUs allotted to a vPar are dedicated to that vPar alone and are not shared with either the VSP or any other vPar or VM guest running on the VSP. Hence, the concept of entitlement does not apply to vPar cores. You can specify a maximum of (total VSP cores –1) for a single vPar.

The following example shows a VSP with 16 cores, of which 4 CPUs are reserved for 4 1-CPU vPars.

```
# hpvmstatus
```

```
[Virtual Machines]
Virtual Machine VM # Type OS Type State   #VCPUs #Devs #Nets Memory
Name
=============== ==== ==== ======= ======= ====== ===== ===== =======
VPAR4           4    VP   HPUX    Off     1      1     1     2048 MB
GUEST5          5    SH   HPUX    Off     1      1     1        2 GB
GUEST7          7    SH   HPUX    Off     1      1     1        2 GB
VPAR3           3    VP   HPUX    Off     1      1     1     2048 MB
VPAR1           1    VP   HPUX    Off     1      1     1     2048 MB
VPAR2           2    VP   HPUX    Off     1      1     1     2048 MB
GUEST6          6    SH   HPUX    Off     1      1     1        2 GB
```

```
# hpvmstatus -s
```

```
[HPVM Server System Resources]

 …
 Total number of operable system cores = 16
 CPU cores allocated for VSP = 1
 CPU cores allocated for vPars and VMs = 15
 CPU cores currently in use or reserved for later use = 4
 …
 Available CPU cores for a virtual partition = 11
 …
```

```
# vparstatus -A
```

```
[Available CPUs]:  11

…
```

The `mpsched` command on the VSP shows that 16 CPUs are still available as all vPars are currently DOWN.

```
# mpsched -s
```

```
System Configuration
=====================

Locality Domain Count: 4
Processor Count      : 16

Domain    Processors
------    ----------
0           0  2  4  6
1           8 10 12 14
2          16 18 20 22
3          24 26 28 30
```

```
# hpvmstart -p 1
```

```
…
hpvmstart: Successful start initiation of vPar or VM 'VPAR1'
```

Now that the vPar is started, one core (ID: 22) from the VSP is dedicated to running the vPar ‘VPAR1’. This can be confirmed by the absence of one VSP core in the `mpsched` command output.

```
# mpsched -s
```

```
System Configuration
=====================

Locality Domain Count: 4
Processor Count      : 15
```

```
Domain     Processors
------     ----------
0           0  2  4  6
1           8 10 12 14
2          16 18 20
3          24 26 28 30
```

Online CPU migration

vPars v6.1 and later supports online migration of CPUs. This means you can add and delete CPU from a live vPar without having to reboot it. During addition, free CPUs from the vPar and Integrity VM guest pool are added to the vPar. When deleted from the vPar, the CPUs return to the vPar and Integrity VM guest pool. You can use the `vparmodify` command to change the number of CPU cores assigned to the vPar.

The vPars v6 product supports dynamic CPU addition and deletion. The selection criteria for CPU addition are performed from within the VSP based on LORA CPU OLD policies. The selection criteria for CPU deletion is performed from within the HP-UX instance that is target of the CPU deletion. The following criteria are used to select the CPU cores to be deleted:

- Only cores in the default pset (see `psrset(1M)`) can be dynamically removed.
- The monarch core can never be deleted.
- If the default pset does not contain enough cores to satisfy a full request to reduce the number of cores, the processor assignments will remain unchanged.
- Processors can be moved to the default pset and then deleted.

For more information about illustration of the feature, see “Online CPU migration for vPar” (page 267).

Transforming vPar to a VM guest

For better flexibility, you could transform a vPar into a VM guest offline. Use the `hpvmmodify -x vm_type=shared` command to transform a VM guest into a vPar. For more information about transforming vPar to a VM guest, “Transformation between VM and vPar” (page 243).

Hyperthreading for vPars

Hyperthreading is supported in individual vPars. By default, the VSP has the hyperthreading (firmware setting) ON in the npartition or server, whereas the `lcpu_attr` tunable is OFF in the VSP. The `setboot` command, when run on an individual vPar shows that HT is ON. You can turn on `lcpu_attr` in an individual vPar using the `kctune` command. By default, `lcpu_attr` is OFF in the vPar (default behavior of HP-UX). Note that even when `lcpu_attr` is OFF in the VSP, each vPar can have its individual `lcpu_attr` enabled to use the hyperthreading functionality in the vPar.

Handling Local MCA

For a general overview of MCAs on Itanium based systems, see “MCAs on VM guests” (page 51).

vPars v6.x operate with actual processor and related hardware. Consequently, MCAs encountered by vPars are handled by the hardware, VSP (host) operating system, and system virtualization components, working together.

On a system running vPars v6.0, any MCA encountered in an individual vPar (or the VSP) results in a system crash that brings down all of the vPars. Starting vPars v6.1.5, a certain class of recoverable, local MCAs caused by a CPU in an individual vPar are isolated to that vPar and do not impact other running vPars.

The vPar OS first tries to automatically recover from such MCAs without bringing down the vPar (APR supported by HP-UX). If that is not possible, the individual vPar goes through a crash dump and is rebooted to recover from the error. Diagnostic dump files known as `tombstones` are generated. These files must be sent to Hewlett Packard Enterprise for analysis.

The type of MCAs recovered typically includes user process register file errors, kernel process register file errors, TLB errors and so on affecting a single vPar. In all other cases of local MCAs affecting individual vPars or any type of local MCA affecting VSP cores and any global MCAs, a server or nPartition crash occurs impacting VSP and all vPars. In most cases, a VSP core dump is also generated. In all cases, MCA logfiles are generated in the standard locations, depending on the platform.

You must be aware of the following behavior:

- If a CPU core experiences an excessive number of MCAs from which the vPar recovered either through APR or through rebooting the vPar, system firmware or diagnostics might deconfigure or deactivate the CPU. In this case, when the vPar reboots, it will not contain a deactivated or deconfigured CPU core, and the MCA error records belonging to the affected CPU core might not be available in the `/var/tombstones` directory.
- If another MCA (of any type) occurs on any other CPU core when recovery of an earlier MCA is not yet completed, this might cause the server or partition to be reset.
- If you stop or reset a vPar before it completely boots up after processing a local MCA, it might lead to the server or partition being reset, depending on the platform. On Superdome 2, this might also result in the nPartition status being displayed as MCA, even though the vPar has actually recovered from the MCA.
- When a local MCA affecting an individual vPar cannot be contained or isolated, it triggers a server or nPartiton reset. In most cases, this manifests as an INIT received by the VSP resulting in a VSP crash dump and reboot. Hence, if there is an unexpected crash of the VSP indicating a transfer of control, verify the system firmware logs to determine if there was an MCA that caused this. The VSP crashdump itself might not have any information about the MCA.

Reserved resources and resource over-commitment

HP-UX vPars and Integrity VM allows the reservation of the resources for VMs and vPars. Reservations imply that a resource will be available when it is needed, with the assurance that a VM or vPar can boot at any time.

The reserved resources setting is managed for each individual VM and vPar and is set using the `resources_reserved` attribute (managed with the `-x` option of the `hpvmcreate` and the `hpvmmodify` command). The default behavior of the `vparcreate` command is to set `resources_reserved` to true when a vPar is created. However the `hpvmcreate` command does not reserve resources by default when creating VMs or vPars. The `resources_reserved` attribute can be managed using the `hpvmmodify` command. Resources that are reserved include memory, CPU, and I/O devices. If a resource is assigned to a VM or vPar that has the `resources_reserved = true`, that same resource cannot be assigned to a different VM or vPar that also has `resources_reserved = true`. It is also not possible to assign a resource to a vPar or VM that has `resources_reserved = true`, if that resource is not currently available.

For example, if all the CPUs have been assigned to other reserving VMs or vPars, then it is not possible to assign CPUs to any additional reserving VMs or vPars. It is possible to assign resources to non-reserving VMs and vPars, however, it is not possible to boot them (because the resources assigned to that VM or vPar are reserved by other VMs or vPars).

Handling faulty CPU

On VSP with HP System Fault Management (HP SFM) software installed, if a faulty CPU is encountered, a CPU deletion request is raised on the host kernel.

If the CPU identified for deletion happens to be a non-Monarch CPU of the running vPar, then it can be dynamically deleted from the vPar. If the CPU is the Monarch CPU of the vPar, then the CPU cannot be deleted.

Deleting a CPU on the VSP does not impact the running VM guests, as the entitlements are adjusted dynamically by the HPVM scheduler with the other remaining physical CPUs that are present in the vPar and Integrity VM guest pool.

Configuring memory for VM guests

Memory virtualization

When a VM guest is started, the memory from the vPars and Integrity VM pool is allocated and presented to the guest as if it were private, physical memory. Each VM guest is provided with a virtualized physical address space called guest-physical memory. The guest operating system manages this guest-physical memory in exactly the same way the operating system manages physical memory on an HPE Integrity system. The VMM manages the mapping of guest-physical memory to real-physical memory on the VSP. Any interaction of the guest operating system with its memory management entities such as page tables and translation look-aside buffers are intercepted by the VMM, controlling access to physical memory management structures. Maximum memory supported for a VM guest is 256 GB.

Overhead memory for VM guests

Each VM guest has a memory overhead depending on its size. A rough estimation of the individual guest memory overhead can be done using the following formula:

```
Guest memory overhead = cpu_count * guest_mem * 0.4% + 64M
```

where,

guest_mem

  is the VM guest memory size (in MB).

cpu_count

  is the number of vCPUs for VM.

For example,

For a VM with 4 vCPUs and 16GB memory, the overhead is 320MB,

For the same 16GB VM with 1 vCPU, it is ~128M.

When you create a 16GB 4vCPU VM, additional 320MB is used up by the VM. For larger VM guests, if the overhead memory computed above is greater than 4G, then overhead memory has been reduced by 50% or capped at 4G based on the following new formula:

```
Guest memory overhead = cpu_count*guest_mem*0.4% + 64M
```

If (Guest memory overhead >4G)

```
Guest memory overhead = MAX (4G, Guest memory overhead/2)
```

For example,

For a VM with 32 vCPUs and 256GB memory, the new overhead memory computed is 16GB instead of 32GB.

This memory is taken from the vPar and Integrity VM guest pool. Note that, there might be some amount of memory taken up from the VSP memory when a guest is started. However, that is in most cases negligible compared to the vPar and Integrity VM guest overhead memory taken up from the vPar and Integrity VM guest pool.

This overhead memory is prereserved during the start of the guest.

The `hpvmstatus -s` command output displays additional information taking into account the required guest memory overhead and guests with reserved resources.

Dynamic memory

Dynamic memory is an optional feature of Integrity VM that allows you to change the amount of physical memory in use by a VM without rebooting the VM.

When dynamic memory is enabled for an Integrity VM, it starts up with a range for memory size; the range specifies a potential maximum and absolute minimum value. At any given time, based on actual system usage, the amount of memory in use by the guest will be within this range.

Ensure to note the following:

- The upper limit of the memory required for the guest must be available on the VSP.
- At run time, memory cannot be increased beyond the upper limit. To increase the limit, you must shut down the VM guest and specify the required upper limit.
- If the Integrity VM guest is migrated online, the target must have the upper limit of specified memory available.

NOTE: Dynamic memory is not applicable for vPar.

To illustrate this feature, it allows a VM that is a Serviceguard node to be used as a standby server for multiple Serviceguard packages. When a package fails over to the VM, the VM memory can be changed to suit the requirements of the package before, during, and after the failover process.

To use dynamic memory, the VM must have the VirtualBase software installed, as described in "Installing VirtualBase on a vPar or VM Guest" (page 28).

For more information about managing Integrity VM dynamic memory, see "Managing dynamic memory from the VSP" (page 253).

Configuring memory for vPars

Memory allocation

When a vPar is started, the memory from the vPars and Integrity VM pool is allocated and presented to the vPar as if it were private, physical memory. vPar memory is not virtualized, so, there is no additional virtualization overhead involved in handling vPar memory. There is no hard limit on the maximum memory configuration for vPars. Barring some overhead memory, the whole of available memory that is present in the vPar and Integrity VM pool can be used by a single vPar.

Overhead memory for vPar

Each vPar has a memory overhead depending on its size and is constant for a given vPar memory configuration, irrespective of the CPU count.

```
vPar memory overhead = vpar_mem * 0.4% + 64M
```

where, `vpar_mem` is the vPar memory size (in MB).

For example, a 16G, 8 CPU vPar, would have a memory overhead of 128M. In this example, even if it became a 16G 16 CPU vPar, the memory overhead would remain 128M.

On large configuration vPars, overhead memory is capped to a maximum value of 320MB. For example, a 2.5TB vPar guest will consume only 320MB of overhead memory unlike 11GB in the previous releases.

Online memory migration

Starting HP-UX vPars and Integrity VM v6.2 release, online addition and deletion of memory is supported. This means that you can add and delete memory from a live vPar without rebooting the vPar.

NOTE: The vPar OS must have the PHKL_43308 patch installed to use this feature. This patch will be automatically installed as a part of the HP-UX 11i v3 March 2013 update on the vPar. If you are running an HP-UX 11i v3 Version prior to March 2013 update on your vPar, you must install this patch.

For vPars with Direct IO devices, this feature is currently limited to blade servers and is not supported on Superdome2 systems.

When memory is added to a live vPar:

- The VSP configures and presents the requested memory from the vPar and Integrity VM memory pool to the vPar.
- The HP-UX kernel in the vPar discovers and integrates the new memory pages. Subsequently, applications can use the new memory.

NOTE: The VSP attempts to obtain memory from the vPar and Integrity VM memory pool, based on the most favourable NUMA characteristics of the vPar. There are no manual controls to change memory selection.

When memory is to be deleted from a live vPar:

- The HP-UX kernel in the vPar selects the memory pages to evacuate, and moves the contents to other available free pages and then frees those memory pages.
- The VSP marks the memory as free, returns the memory back to the vPar and Integrity VM memory pool, and this memory can be assigned to other vPars.

For more information about the online memory migration, see *Reconfiguring vPars v6 memory with zero downtime* at **http://www.hpe.com/info/hpux-hpvm-docs**.

Base and floating memory

In HP-UX, portions of the memory, that contain kernel code and certain kernel data structures cannot be evacuated. While allocating memory during boot or runtime, the HP-UX kernel needs to know in advance, the memory type to use for kernel data structures as they cannot be evacuated. To aid HP-UX kernel in this differentiation, vPars software sub divides memory into the following two types:

- base memory
- floating memory

NOTE: Base memory and floating memory is applicable only for vPars.

Base memory is used by the vPar HP-UX kernel for critical data structures. You can increase the amount of base memory of a live vPar but you cannot decrease it.

Floating memory is typically used for user applications. You can either increase or decrease floating memory from a live vPar.

For the list of command line options for base and floating memory configuration, see "Command options for base or floating memory configuration" (page 262).

NOTE: The floating memory that is deleted from one vPar can be allocated to another vPar as base or floating memory.

Allowed memory modification operations

Table 10 (page 58) lists the operations that are allowed for each memory type depending on the vPar state.

Table 10 Types of memory

vPar state	Base memory		Floating memory	
	ADD	**DELETE**	**ADD**	**DELETE**
Online	Allowed	Not Allowed	Allowed	Allowed
Offline	Allowed	Allowed	Allowed	Allowed

For more information about illustrations and command line options related to vPar online memory migration, see "An illustration of vPar online memory migration" (page 265).

Guidelines for base and floating memory configuration

The HP-UX kernel requires a certain percentage of total memory to be base memory for system performance and to ensure that there is adequate memory available for critical system needs. The following table lists the recommended minimal amount of memory that must be configured as base memory for some typical memory sizes.

Total Guest Memory	Minimum Base Memory
1 GB to 3 GB	1 GB
4 GB to 8 GB	1/2 of total memory
9 GB to 16 GB	4 GB
Over 16 GB	1/4 of total memory

⚠ **WARNING!** It is mandatory that the base and floating memory guidelines specified are adhered to. If the proportion of base to floating memory is too low, the vPar could experience a panic or hang. The vPar may not boot without sufficient base memory.

NOTE: Some workloads or vPar kernel configurations might require more base memory than what is recommended here.

The system administrator must configure enough base memory to allow the vPar to achieve required baseline application performance taking into consideration the following constraints:

- The kernel has more flexibility using base memory. The kernel restricts the use of floating memory to contents that it can later relocate if necessary. Hence, a system with all base memory would perform better compared to a system with the same amount of memory for system use but divided between base and floating memory.
- Some kernel sub-systems and applications do their allocations based on the amount of base memory discovered at system boot time. These subsystems or applications could allocate their caches based on the amount of base memory available to the kernel during boot time, and might not scale that cache when more base memory is made available later through online memory addition. Hence, the performance of a system that is booted with less base memory followed by online addition of base memory might not perform the same as a system configured with the sufficient amount of base memory prior to booting.
- On a system with heavy amount of memory utilization, the HP-UX kernel might take minutes or even hours to evacuate memory. Hence, it is advisable not to delete floating memory on a heavily loaded system.
- Depending on the system load, adding or deleting large amount of floating memory to or from a live vPar can take significant time. This can sometime result in Serviceguard heartbeat

failures on vPars configured as Serviceguard nodes. Hewlett Packard Enterprise recommends that a single large memory transaction be split into multiple smaller transactions.

For example, if a partition contains a large amount of floating memory, instead of deleting all the floating memory in one operation, it might help to split it into multiple smaller floating memory delete operations.

NOTE: Under very rare conditions, the kernel could consume some portion of floating memory during boot. In such a situation, the portion of floating memory consumed by the kernel will be converted to base memory. When that happens, the guest configuration file will be updated to reflect the increase in base memory and decrease in floating memory for that vPar.

Granularity and memory modification

Granularity refers to the unit size in which memory is assigned to all the vPars. The memory granule size is fixed at 64 MB. Hence, all memory operations are performed in multiples of this size. On a live vPar, a maximum of 255 granules can be specified per memory operation. Therefore, the maximum amount of memory that can be added to or deleted from a vPar in a single operation is 16,320 MB.

Memory is always migrated (either add or delete operation) in multiples of 64 MB granules. Hence if a memory migration operation is initiated where the requested memory is not a whole multiple of 64 MB, the actual memory considered for the operation will be round down to the previous granule size.

For example, if a request is made for deletion of 100 MB of memory, only 64 MB will be deleted. If a 257 MB deletion is requested, 256 MB will be deleted. Similarly, if a 100 MB memory addition request is made, 64 MB will be added and not 100 MB. To minimize any unintended changes, it is recommended to perform memory migration operations in terms of multiples of the granule size.

Memory allocation and usage for VMs and vPars—Implementation notes

vPar and VM memory is allocated in granules of 64 MB. If a vPar's or VM's memory size is not an even multiple of 64 MB, the API/CLI code rounds up, but hpvmapp rounds down. Consequently, there will be 64 MB of memory reserved for the vPar/VM, but not used by it, and the vPar/VM might have up to 64 MB to 1 KB of memory less than was allocated.

To work around this problem, set the vPar or VM memory size to a multiple of 64 MB.

Overhead memory calculations for a vPar and VM guests are different. A vPar always takes less or same overhead memory than a VM guest with same configurations. If a vPar guest is converted to a VM guest and if there is no sufficient memory to accommodate increased overhead memory, the modified guest may not start.

6 Storage devices

This chapter describes vPar and Integrity VM storage and explains how to configure and use vPar and Integrity VM guest storage. The way you configure and manage vPar and VM guest storage affects the way vPar and VM guest perform. To benefit most, learn how the VSP makes storage available to vPars and VM guests.

Storage goals

To successfully configure and manage virtual storage, it is helpful to understand the basic goals of the vPars and Integrity VM storage subsystem, including:

- Storage utilization
- Storage availability
- Storage performance
- Storage security
- Storage configurability

Storage utilization

The main purpose of vPars and Integrity VM is to increase system resource utilization on Integrity servers. The vPar and VM guest storage subsystem meets this goal by permitting multiple vPars and VMs to share a variety of physical storage adapters and devices that are available on an Integrity server. Furthermore, the vPars and Integrity VM storage subsystem allows a single storage LUN on the VSP to be carved up into smaller entities that can be used as separate individual disks or DVDs on the virtual platform.

Storage availability

Like HPE Integrity servers, it is expected that VMs and vPars have different storage device types available for use. The vPar and VM guest storage subsystem allows a guest OS to use disks, DVDs, tapes, and media changers. Additionally, the way that virtualization abstracts the physical hardware provides a common supportable interface with which a guest OS can interact. Because a guest OS accesses only vPars and Integrity VM virtual hardware, it can use physical hardware that it does not support on an Integrity server.

Storage performance

Each release of the vPar and Integrity VM product strives to improve performance. Performance is improved in each release by lowering costs of virtualization, exploiting new features in the VSP, and tuning operating systems for the virtual platform. At the same time, vPars and Integrity VM provides more virtualization choices to VSP administrators, so that they can find the best balance between virtualization and performance to meet their needs.

Storage security

To ensure that multiple vPars and VMs can run on one physical machine without each accessing the resources that belong to the others, the VSP isolates each VM and vPar. Using vPar and Integrity VM commands, the VSP administrator determines the physical storage resources that each VM and vPar can access. This storage isolation is maintained by the vPar and VM guest storage subsystem through DMA boundary checks on each vPar/VM I/O operation, thereby ensuring that one VM or vPar does not access the memory of another.

Storage configurability

VSP administrators expect the vPars and VM guests to be as easily configurable as HPE Integrity servers. The vPar and VM guest storage subsystem allows for easy changes to the storage devices through vPars and Integrity VM commands. Using these commands, the VSP administrator dynamically adds, deletes, and modifies storage devices on VMs and vPars. Guest administrators can change some storage, limited in scope by the VSP administrator, using the virtual console.

Storage architectures

The vPars and Integrity VM guest storage subsystem provides three types of storage architectures:

- Shared I/O
- Attached I/O
- NPIV

Shared I/O

The shared I/O architecture is a means by which a vPar and VM guest accesses an entirely virtualized storage subsystem provided by the VSP. The VSP emulates a HPVM proprietary hardware device to the vPar or VM guest. The vPar or VM guest storage subsystem interacts with the VSP to complete I/O operations to the VSP storage entity. This abstraction enables the VSP administrator to share physical VSP storage hardware across multiple vPars or VMs and to allocate that storage at sub-LUN levels.

The individual storage LUNs are shared by dividing a VSP LUN into smaller parts such as logical volumes or files. Each of these sub-LUN VSP entities can then be used as media for separate virtual storage devices. The vPars and VM guests access the virtual storage devices as real storage devices, with no knowledge that the virtual storage media is actually a sub-LUN VSP entity.

The way the virtual storage media is accessed by the vPar or VM guest storage subsystem allows them to share physical VSP storage adapters. All virtual storage media is accessed through user-defined interfaces on the VSP. The VSP maintains complete control of the physical hardware and handles the vPar or VM guest I/O operations just as it would be handled for any other user application. Thus, just as hardware is shared among normal applications running on the VSP, vPar and VM guest I/O is shared across the physical storage as well.

The shared I/O architecture also provides for whole LUNs to be virtualized. While this does not increase storage utilization, it does provide higher storage availability. Because the LUN is virtualized, the guest OS need not support the physical VSP LUN. It is sufficient to support the virtualized version of VSP LUN. Thus, by using shared I/O a vPar or VM guest can run with any physical hardware that is supported by the VSP.

Finally, all vPar or VM guest I/O requests in shared I/O are processed by virtual adapters. A virtual adapter is an emulation of a proprietary adapter type that a special driver in the guest OS accesses. The virtual adapter uses internal vPar or VM guest storage subsystem calls to handle communication of vPar or VM guest I/O to the virtual devices. This connection between the virtual adapter and the virtual devices must not resemble anything in an HPE Integrity server system. It is emulated so that the vPar or VM guest does not know the difference.

Attached I/O

Attached I/O allows a vPar or VM guest visibility to the real device and its properties. In this architecture, the vPar or VM guest storage subsystem attaches a LUN path on the VSP to a virtualized storage adapter. The LUN can be a DVD, tape, or media changer.

The main difference between shared I/O and attached I/O is the degree to which a physical storage subsystem is virtualized. In shared I/O, an entire storage subsystem is virtualized. Therefore, all physical adapters on the VSP and all the storage connected to those adapters can be shared among vPars and VM guests. In attached I/O, only the storage adapter is virtualized. Therefore, only the VSP physical storage adapters are shared.

To provide the vPar or VM guest with complete control over attached devices, the vPar and VM guest storage subsystem interprets I/O requests from the guest device drivers into I/O requests that can be completed by the VSP storage subsystem on the behalf of vPar or VM guests. In the process, the VSP storage subsystem sends all the actual data and responses back to the vPar or VM guest device drivers.

NPIV devices

NPIV is a fibre channel technology that allows you to create multiple virtual Fibre Channel ports over a single physical Fibre Channel port on the VSP. These are then allocated to vPars or VM guests on the VSP. With NPIV, a vPar or VM guest discovers SAN devices on its own, just the way it is done on a physical server.

For more information about NPIV and the steps to configure NPIV, see "NPIV with vPars and Integrity VM" (page 99).

vPar and VM guest storage implementations

This section describes the implementations of the vPar and VM guest storage architectures.

vPar and VM guest storage adapters

The AVIO storage adapter is a high performance virtual storage adapter used by vPars and VM guests with paired OS drivers in the guest and host. The AVIO virtual storage adapter supports up to 128 non-NPIV and 2048 NPIV storage devices. AVIO leverages storage stack features from the VSP to provide optimal storage manageability in the guest.

NOTE: For optimal performance, you must take care to ensure that the versions and patch levels of both the guest and host AVIO storage drivers are synchronized.

vPar and VM guest storage devices

vPar and Integrity VM supports a variety of virtual, attachable, and NPIV devices. Disk and DVD-ROM devices support several virtual media types (see "Virtual devices" (page 62)). Physical tapes, media changers, and CD or DVD burners are attachable. They can be used to backup data directly from a vPar or VM guest (see "Attached devices" (page 63)).

With all the three storage implementations, the maximum transfer size can be 1 MB for any guest operating system.

Virtual devices

Table 11 (page 62) lists the virtual disk types supported by vPar and Integrity VM guest.

Table 11 Virtual disk types

Virtual disk type	Backing storage device
Virtual Disk	VSP disk, include Veritas DMP DFSs and cluster DSFs
Virtual LvDisk	VSP LVM or VxVM logical volume
Virtual FileDisk	VSP VxFS file

Table 12 (page 63) lists the virtual DVD-ROM types supported.

Table 12 Virtual DVD-ROM types

Virtual DVD type	Backing storage device
Virtual DVD	Disk in a VSP physical DVD drive
Virtual FileDVD	ISO file on a VSP VxFS file system
Virtual NullDVD (empty)	VSP physical DVD drive or VxFS directory

Attached devices

vPars and Integrity VM supports a suite of attached devices on HP-UX 11i v2 and HP-UX 11i v3 guests to complete data backups from a vPar or VM guest. vPars and Integrity VM attaches these devices using a special pass-through capability built into the AVIO storage driver on the host. With this pass-through mechanism, vPar or VM I/O requests are sent through the virtual storage subsystem to the physical device. The device responses are sent to the AVIO storage driver, which sends the responses to the vPar or VM. The vPar or VM guest has visibility to all the data and responses. Hence, support for the attached physical device must be provided by the guest OS.

Attached devices include:

- CD/DVD burners
- Media changers
- Tape devices

Attached devices allow sharing of tapes, changers, and burners among multiple guests and the host. Attached I/O supports USB 2.0 DVD burners.

NPIV devices

For more information about NPIV devices, see “NPIV with vPars and Integrity VM” (page 99).

Configuring vPar and VM guest storage

This section explains how to plan and set up vPar and VM guest storage.

Storage considerations

When you configure storage for a vPar or VM guest, consider the following:

- Storage supportability
- Storage performance
- Storage multipath solutions
- Storage management
- Storage changes
- Virtual storage setup time

Storage supportability

Before you configure vPar or VM guest storage, ensure that the VSP storage can be supported by the vPar or VM guest by ensuring the following:

- All the VSP storage available for use by a vPar or VM guest must meet support requirements for the Integrity server and OS version that comprises the VSP. If the physical storage is not supported by the VSP, it is not supported for use by a vPar or VM guest.
- All the VSP storage available for use by a vPar or VM guest must be connected with a supported adapter and driver type. For more information about the list of supported types,

see the *HP-UX vPars and Integrity VM Release Notes* at **http://www.hpe.com/info/hpux-hpvm-docs**.

If the physical storage is not connected with one of the supported adapter and driver types, it cannot be used by a vPar or VM guest. Use the `ioscan` command to display the VSP storage that is connected to adapters and drivers.

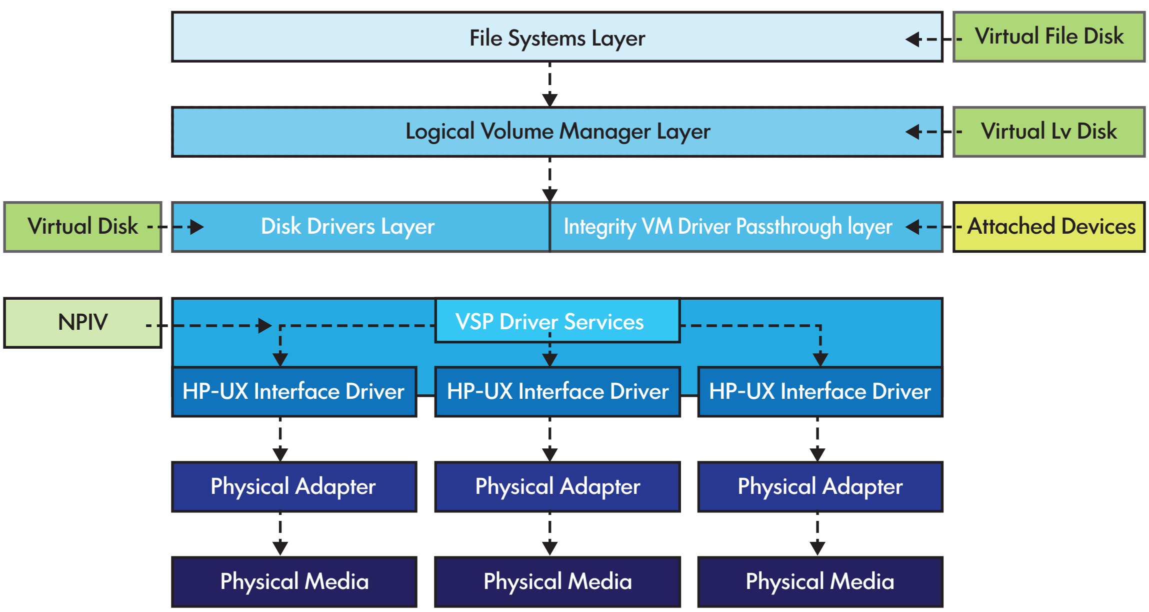
- Any VSP attachable device available for use by a vPar or VM guest must be supported by the guest OS to which it is attached. If the physical device is not supported by the guest OS, the device cannot be attached to the vPar or VM guest.

Storage performance

To meet the performance requirements of applications running in vPars and VM guests, consider the potential performance of each type of storage device.

Different types of virtual media have different effects on the performance of the virtual device because they communicate differently with the VSP to complete vPar or VM guest I/O operations. To understand the effect of the virtual device type on potential performance, consider the vPar and VM guest storage I/O stack illustrated in Figure 7 (page 64).

Figure 7 Storage I/O stack



For a virtual I/O operation to be completed, the I/O has to travel round trip between the virtual storage adapter and the VSP physical storage device. The longer the path is, the longer it takes for virtual I/O to be completed. As shown in Figure 7 (page 64), a virtual I/O operation must traverse each software layer in order, from where it originates to the physical media. For example, a virtual I/O operation for a Virtual FileDisk must traverse any logical volume managers the file system is on and the disk drivers that control the whole disk. Therefore, in general, the higher the virtual media is in the VSP I/O stack, the slower it operates.

The simplified I/O stack in Figure 7 (page 64) does not completely illustrate all the choices that can affect the performance.

- Performance of different software layers differs.
- The interfaces to each software layer are different, allowing Integrity VM different ways to send I/O through the layers.

  For example, whole disks can achieve higher throughput rates than logical volumes and file systems.
- The I/O layer might have features to help performance increase beyond a lower layer.

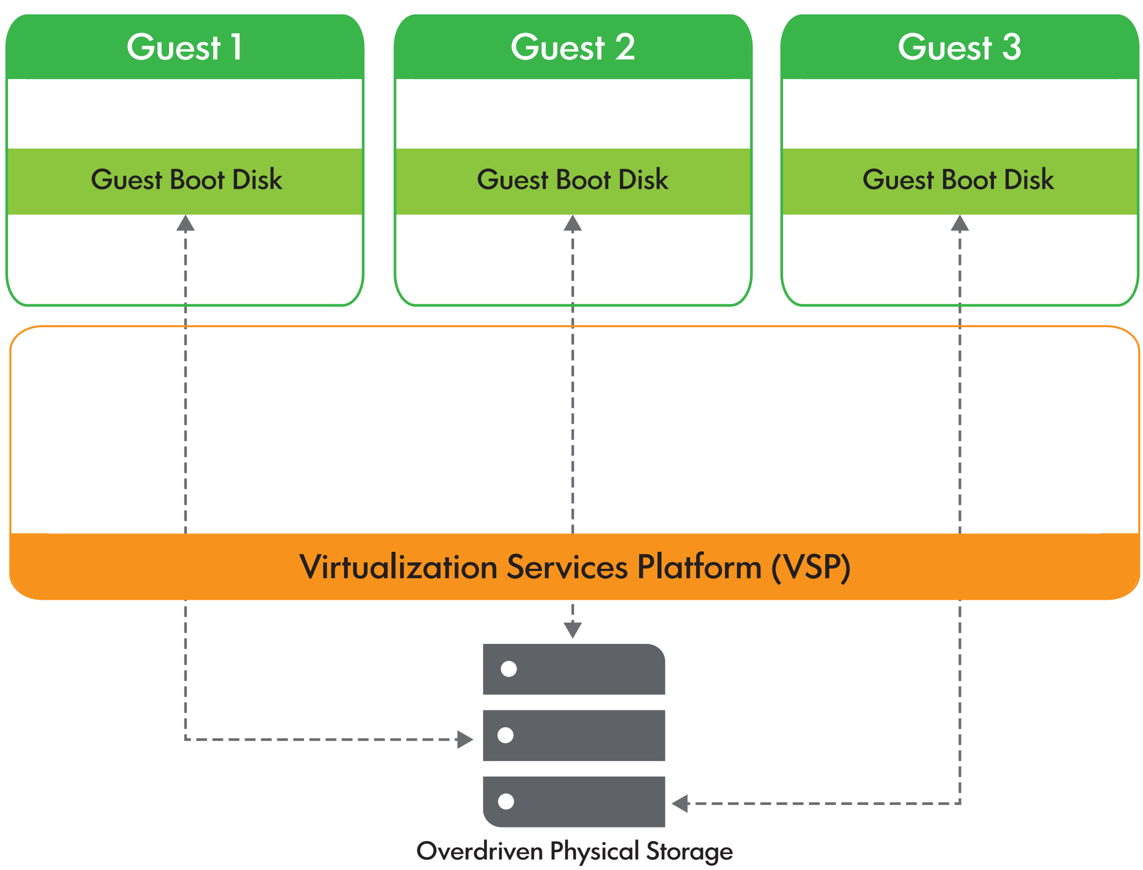
  For example, the file cache of the file system might help a Virtual FileDisk perform better on some I/O workloads than the other virtual device types, which have no such caching.

For more information about tuning performance at each software layer on the VSP, see the vPars and Integrity VM white papers at **http://www.hpe.com/info/hpux-hpvm-docs**.

When you configure virtual devices, consider how the virtual media maps to the physical storage. All virtual media connects to a piece of physical media in the data center. You can ensure the best performance by understanding the impact of the physical storage and the way I/O accesses it.

It is important to know where the virtual media is located on physical storage devices. With vPars and Integrity VM, a single physical disk might be sliced into logical volumes or files. Slicing up physical disks increases utilization, but it can affect the performance of the physical device. The guest OS treats the virtual disk as a whole disk, not as a part of a physical one. Over-slicing physical storage can overload the ability of a physical device to handle virtual I/O that is meant for whole disks. Figure 8 (page 65) shows a common mistake of overdriving physical storage with multiple guest OS boot disks, which are often I/O intensive.

Figure 8 Overdriving physical storage hurts performance



You can provide workloads that the physical devices can handle for all the virtual devices layered on top of them. You can use the performance tools on the VSP, such as *sar(1M)*, to see how the physical storage is keeping up with the virtual device demands.

The way the virtual media I/O gets to the physical storage backing is also an important consideration. As shown in Figure 7 (page 64), all virtual I/O goes through a general VSP I/O services layer that routes the virtual I/O to the correct VSP interface driver. The interface driver then controls the physical I/O adapter to issue virtual I/O to the physical storage device. By load balancing across these physical adapters, virtual I/O bottlenecks can be eliminated at the physical hardware layers, thereby increasing performance. Load balancing can be done by using a multi-pathing solution on the VSP. For help with selecting a multipath solution for a virtual media type, see "Storage multipath solutions" (page 66).

The performance of attached devices is largely determined by the type of physical device attached to the VM. Tapes, media changers, and CD or DVD burners are inherently slow devices, not significantly impacted by the software overhead of vPar and Integrity VM.

Storage multipath solutions

vPars and Integrity VM virtual devices support the built-in multi-pathing of the HP-UX 11i v3 VSP, which is enabled by default to provide improved performance, load-balancing, and higher availability for vPars and VM guests. Starting with v6.3, vPars and Integrity VM virtual devices also support Veritas DMP devices. Currently, there are no multipath solutions supported for the attachable device types of tapes, media changers, and CD or DVD burners.

For non-NPIV devices, there are no multiple paths to virtual devices inside a VM. The following are the reasons for the support of multi-pathing only on the VSP for non-NPIV based AVIO backing stores:

- The VSP is the only place where all virtual I/O can be properly load balanced for the best overall performance. A single VM cannot account for all the other vPar or VM I/O with which it is competing on the VSP (see Figure 7 (page 64)).
- Running a multipath solution in a vPar or VM guest does not provide any high availability for a virtual device. Virtual connections between virtual adapters and their devices are never lost until an `hpvmmodify` command is used to disconnect them. The only connection ever lost is the ability of a virtual device to access its own virtual media through the VSP. Errors in communication to the virtual media are properly emulated as media errors sent to the guest OS, not as path failures.
- The VSP does not return specific errors to Integrity VM for hardware path failures. vPars and Integrity VM does not detect such events and does not pass them to the vPar and VM guest.

For NPIV devices, multi-pathing products run on the vPar or VM guest and not on the VSP. For more information about multi-pathing for NPIV devices, see Section (page 105).

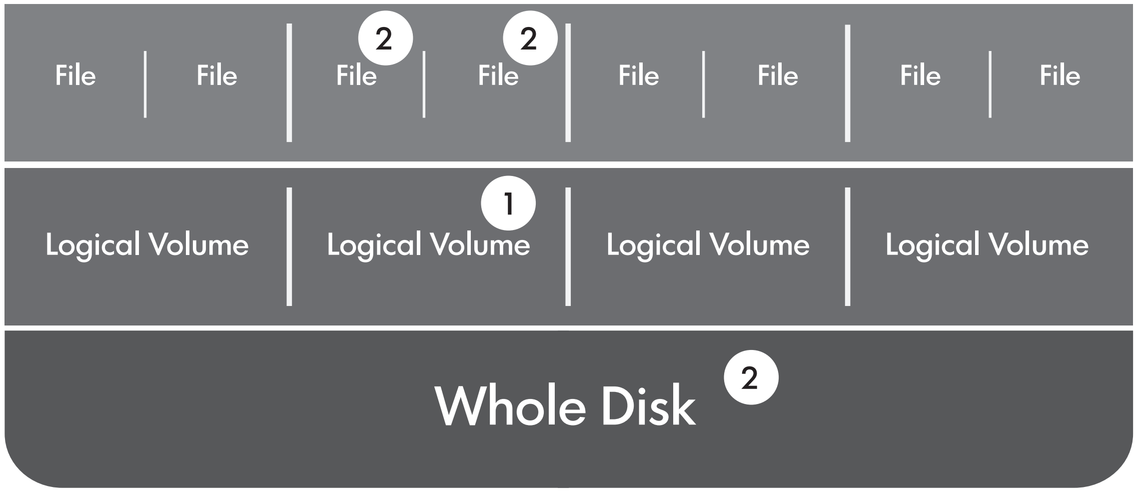
Storage management

Before you decide how to divide VSP storage, consider the impact on the management of the storage subsystem.

A VSP administrator manages vPar or VM storage to make sure virtual media is allocated safely. This begins with understanding the VSP I/O stack and knowing from where the virtual media is being allocated. When you share a physical backing storage device among VMs, potential conflicts are not always obvious. For example, if you use a file in a file system on a whole disk as a backing store, the raw whole disk device itself cannot also be used as a backing store.

Figure 9 (page 67) shows an example of a VSP I/O stack as it applies to a single LUN.

Figure 9 Sub-LUN storage allocation example



The VM is allocated a logical volume from the LUN for a Virtual LvDisk.

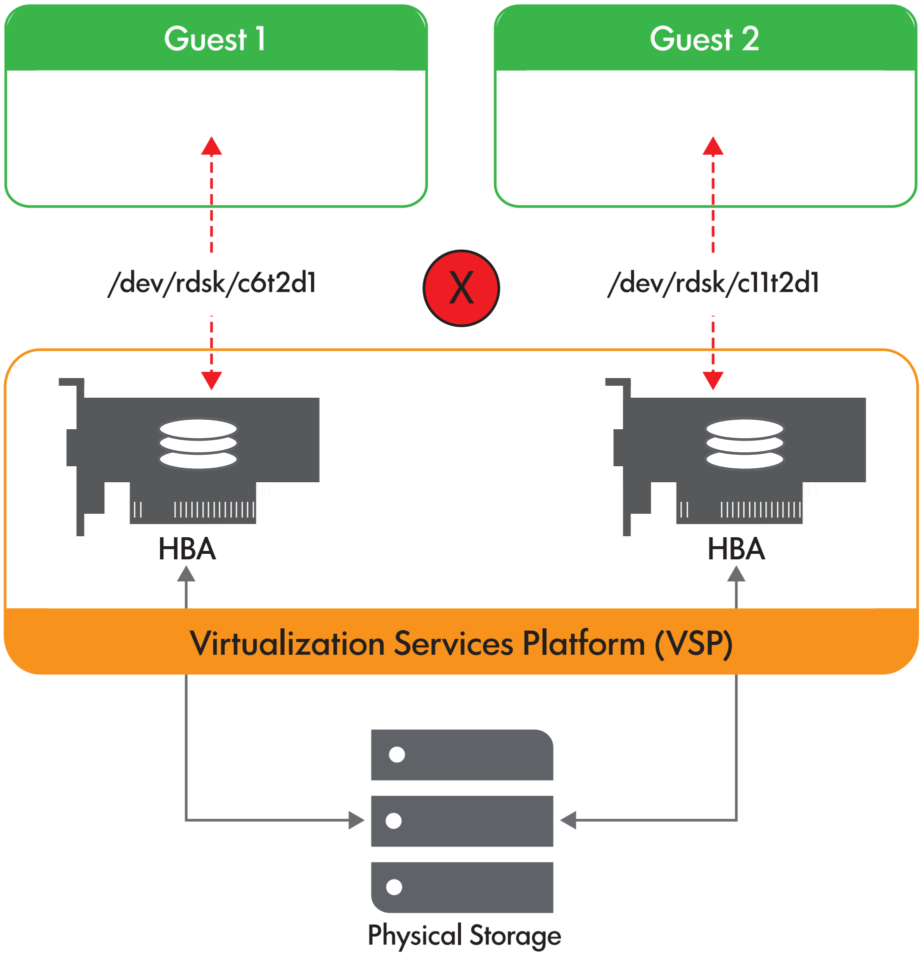
- The logical volume that has been allocated is labeled **1**.
- The parts of the disk that cannot be allocated are labeled **2**.

Those parts that are no longer available include the files that were on the logical volume and the whole disk that makes up part of the volume group. If any of these parts are allocated for other virtual devices, data on the Virtual LvDisk can get unintentionally over-written.

Those parts that are still available for reallocation include other logical volumes that are on the disk, and files that are on those logical volumes. These pieces can be allocated without the problem of data getting damaged because they do not overlap with the Virtual LvDisk.

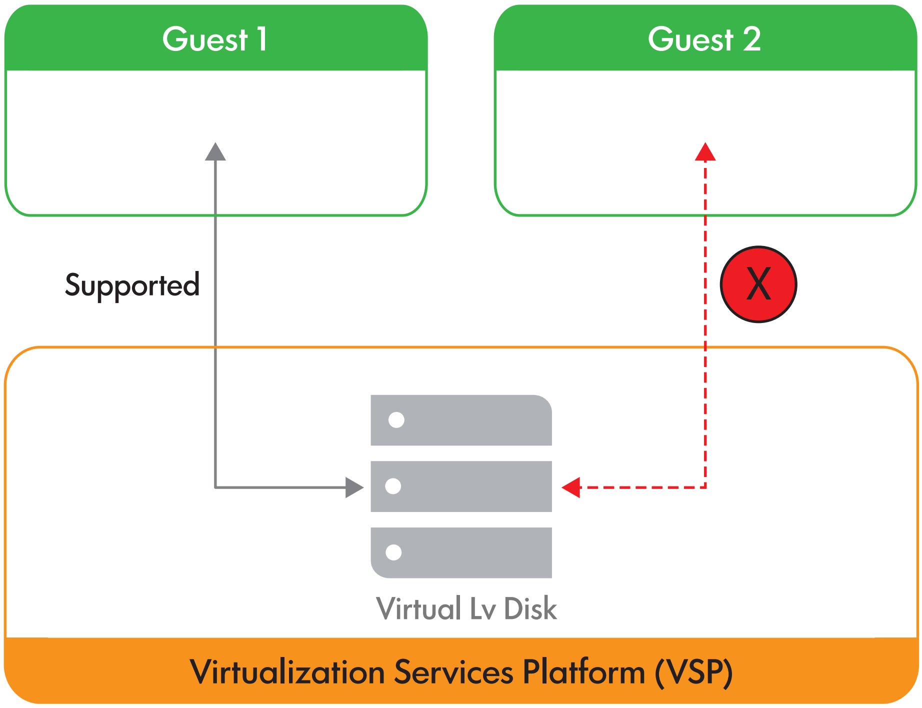
You must avoid whole LUN collisions, beyond avoiding sub-LUN collisions. The same storage resource, virtual, or attached, cannot be specified more than once to the same VM. HP-UX 11i v3 supports both legacy per-path device files (for example, `/dev/rdsk/c6t2d0`) and agile non-path specific device files (for example, `/dev/rdisk/disk`). As shown in Figure 10 (page 68), there may be more than one legacy device file that points to the same physical storage device, while there is only one agile device file for a given physical storage device. Starting vPar and Integrity VM v6.0 onwards, only agile DSFs must be used to configure guest backing stores. Adding virtual devices to the guest using legacy device files or starting a guest that contains backing stores specified using legacy files will fail.

Figure 10 Bad multipath virtual media allocation



Also, the same storage resource, virtual or attached, cannot be simultaneously shared between VMs, unless otherwise specifically exempted. Figure 11 (page 69) shows a Virtual LvDisk being shared across VMs, which is not supported.

Figure 11 Bad virtual device allocation



As these examples illustrate, it is important to know where storage is allocated from to avoid data getting damaged with vPars, VMs, or even the VSP. Management utilities such as the HP SMH utility allows you to track disk devices, volume groups, logical volumes, and file systems. You can use these utilities to annotate devices so that VSP administrators can know the vPars or VMs that are using each VSP storage device.

To show each disk only once, management utilities consolidate multipath devices into one disk. When you are dividing up the disk, you must use all the parts of a single disk on a single VM. Allocating different parts of the same disk to different VMs makes it difficult to manage and to isolate problems.

When an LVM volume group is deactivated, the physical volumes used by that storage is designated as unused by HP-UX system administration tools such as HP SMH. This is also true for Integrity VM storage management. As a result, these physical volumes are not automatically protected from use by VMs as virtual disks.

You can resolve this problem in one of the following ways:

- If the volume group is to remain deactivated, the VSP administrator can manually add the physical volume as a restricted device using the `hpvmdevmgmt` command.
- After activating the volume group, run the `hpvmhostrdev` command so that the VSP storage management database is updated accordingly.

An HP-UX system administrator can deactivate a volume group using the `vgchange` command. It can also be deactivated, if it is a shared LVM (SLVM) volume group whenever the associated Serviceguard cluster is reconfigured, or the VSP system is rebooted. You must verify that all SLVM volume groups are activated after a VSP reboot or Serviceguard cluster reconfiguration.

vPars and Integrity VM checks the present physical configuration when you create a vPar or VM using the `hpvmcreate` command. If the vPar or VM uses backing stores that are not available, the vPar or VM is created, and warning messages provide additional details. If you use the `hpvmstart` command to start a vPar or VM that requires physical resources that are not available on the VSP, the vPar or VM is not allowed to start, and error messages provide detailed information about the problem.

Some devices must be restricted to use by the VSP and to each guest (for example, boot devices and swap devices). Devices can be restricted using the `hpvmdevmgmt` command. For more information about sharing and restricting devices, see "Restricting VSP devices" (page 273).

Any alternate boot device for a vPar or VM guest must be set with the same care that you would use on a physical system. If the primary boot device fails for any reason, a vPar or VM set to `autoboot` attempts to boot from devices in the specified boot order until either an option succeeds or it reaches the EFI Shell. You must make sure that any specified boot options and the boot order, are appropriate for the guest. For more information about the `autoboot` setting, see Table 22 (page 151).

Storage changes

Depending on how you set up storage for a vPar or VM guest, the resulting configuration can be more or less difficult to change.

The ability to change virtual media depends on the type of virtual media used. Whole disks are not normally adjustable in terms of size, but some storage enclosures might permit the adjustment of a LUN without losing the data of that LUN. Logical volumes are adjustable without losing any data. Finally, files can be changed easily with VSP file system commands.

Changes to virtual media can take place on the VSP only after the virtual device that uses the media is removed from the active vPar or VM. The `hpvmmodify` command denies the attempts to change virtual devices that have I/O active on them. After an active vPar or VM guest is allocated virtual media for a virtual device, that vPar or VM guest owns that media and can access it any time. VSP administrators must coordinate with guest administrators about active VM changes, if the two roles are served by different individuals.

This coordination might also be necessary for attached I/O devices. After a VSP device is attached to the vPar or VM guest, it is controlled and owned by that vPar or VM guest. Modifications to the attached device, such as changing a tape, can be done physically without detaching the device from the vPar or VM guest. However, such changes must be coordinated with the VSP administrator, especially if the guest administrator has no physical access to the device attached to the vPar or VM guest.

All types of virtual storage devices can be added and removed dynamically from vPars or VMs. That is, virtual disks, virtual DVDs, tapes, media changers, and CD or DVD burners are all hot-swappable.

Starting with vPars and Integrity VM v6.3, virtual storage adapters can be added dynamically to a vPar or VM guest. Dynamic deletion of a virtual adapter from a vPar or VM guest is supported starting with vPars and Integrity VM v6.3.5. For more information about addition and deletion of storage adapters, see "Dynamic addition of storage adapters" (page 71) and "Dynamic deletion of storage adapters" (page 71).

PCI OLRAD operations on Storage IO card with active vPars or VM guests

Starting with vPars and Integrity VM v6.3, a PCI Online Replacement of an I/O card on the VSP is supported without bringing down active vPars or VM guests that may be using resources backed by the card being considered for replacement. This is done as long as no critical resources in the vPar or VM guest are impacted.

PCI Online Deletion of IO cards on the VSP is not supported if there are active guests using resources backed by the card being considered for deletion.

For more information about PCI OLR and the associated restrictions, see "PCI OLR support on VSPs" (page 173).

PCI Online Addition of IO cards on the VSP is supported if there are active vPars or VM guests on the VSP. After the device is added online, backing stores seen through the new I/O card or NPIV HBAs backed by the new IO card can be added online to the vPars and VM guests. For more information about PCI Online addition, see "Dynamic addition of storage adapters" (page 71).

Dynamic addition of storage adapters

Starting with v6.3, vPars and Integrity VM storage adapters can be dynamically added to a running vPar or VM guest. This is in addition to the existing ability to add new LUNs behind an existing virtual adapter. This capability is available with both HPVM AVIO Storage adapters and HPVM NPIV Storage adapters. In the case of AVIO Storage adapters, the feature allows addition of storage capacity without guest downtime. With NPIV Storage adapters, it allows online addition of storage capacity and online addition of redundant paths to an existing NPIV LUNs.

For more information about dynamic addition of IO devices to vPars and VM guests, see "Dynamic I/O for vPars and Integrity VM guests" (page 269).

Dynamic deletion of storage adapters

Starting with v6.3.5, both NPIV and non-NPIV AVIO storage Host Bus Adapters (HBAs) and LUNs behind them, may be dynamically deleted. Several enhancements have been made for dynamic management of non-NPIV HBAs and LUNs visible below it.

- Dynamic removal of the last LUN behind a non-NPIV AVIO HBA

  Prior to vPars and Integrity VM v6.3.5, it was not possible to remove the last LUN behind non-NPIV AVIO HBAs. `hpvmmodify(1M)` has been enhanced to allow this operation.

  Note that, when the last LUN is removed from non-NPIV AVIO HBAs, the corresponding HBA is also automatically removed.

- Removing storage device special files within vPars and Integrity VM

  Prior to vPars and Integrity VM v6.3.5, it was necessary to manually run `rmsf(1M)` within the vPars and Integrity VM guest to close the device and remove the storage device special files before the corresponding device could be deleted. This manual action is no longer required; when a LUN is deleted, the `hpvmmodify` command automatically checks for device usage, and, if not in use, closes it and cleans up the corresponding device special file.

- Dynamic removal of a non-NPIV HBA and all LUNs behind it

  `hpvmmodify(1M)` has been enhanced to delete a non-NPIV AVIO HBA along with all LUNs visible under it, in a single operation.

For example,

```
# hpvmmodify -P guestname -d hba:avio_stor:0,2
```

Deletes the non-NPIV AVIO HBA at PCI bus-0, device-2, along with all LUNs behind it.

AVIO Storage adapters (both NPIV and non-NPIV) can be dynamically deleted from vPars or Integrity VM guests only if the devices visible through the virtual adapter are not SYSTEM CRITICAL or DATA CRITICAL to the vPar or Integrity VM guest.

For more information about dynamic deletion of I/O devices, see "Dynamic I/O for vPars and Integrity VM guests" (page 269).

NOTE: You cannot delete the first AVIO storage HBA that a vPar or a Integrity VM guest is configured with while the vPar or Integrity VM guest is online. The AVIO storage HBA that gets configured first can only be deleted while the vPar or Integrity VM guest is offline.

For example,

A vPar or a VM guest's AVIO storage HBAs are at (3,0) and (3,1). In this case the HBA at (3,0) cannot be removed dynamically.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name  VM #  Type  OS Type    State        #VCPUs  #Devs  #Nets  Memory
 guest1                                  1        SH      HPUX         On (OS)       2                  2
  1            4 GB

#hpvmstatus -p1 -d
[Storage Interface Details]
disk:avio_stor:3,0,0:disk:/dev/rdisk/disk52
hba:avio_stor:3,1,0x100000110A03002A,0x100000110A03002B:npiv:/dev/fcd3

#hpvmmodify -p1 -d disk:avio_stor:3,0,0:disk:/dev/rdisk/disk52

hpvmmodify: A Dynamic IO deletion operation has been initiated for this VM or vPar. Check hpvmstatus output or
 syslog for completion status.

#hpvmstatus -p1 -V
[Dynamic I/O Interface Details]
IO OLAD operation in progress        : none
IO OLAD current operation argument : none
IO OLAD last operation completed    : LUN Delete
IO OLAD last operation argument     :
Device type                              : disk
Adapter type                            : avio_stor
Ioscan format                           :0/0/0/0.0.0
Bus                                          : 0
Device                                     : 0
Function                                  : 0
Target                                     : 0
Lun                                          : 0
Physical Device                       : /dev/rdisk/disk52

IO OLAD last operation status     : failed_guest
IO OLAD last operation error      :  Cannot delete the first storage IO device

Suppose, later a dynamic addition of a AVIO storage HBA is done at slot (2,4).

#hpvmstatus -p1 -d
[Storage Interface Details]
hba:avio_stor:2,4,0x100000110A03000A,0x100000110B03000A:npiv:/dev/fcd5
disk:avio_stor:3,0,0:disk:/dev/rdisk/disk52
hba:avio_stor:3,1,0x100000110A03002A,0x100000110A03002B:npiv:/dev/fcd3
```

In the above case slot wise (2,4) is first AVIO storage HBA. However, AVIO storage HBA at slot (3,0) still cannot be deleted as this was the first configured AVIO storage HBA when the vPar or a VM guest was booted and AVIO storage HBA (2,4) is added later. Later,when the vPar or a VM guest with above configuration is rebooted or restarted, AVIO storage HBA at slot (2,4) gets configured first and hence, it cannot be removed dynamically.

Notifying guest OS of changes in guest storage configuration

With HP-UX 11i v3, the AVIO storage vPar and VM guest driver can receive events asynchronously from the VSP whenever the underlying storage, such as LUN or target changes state, for example, when a new LUN or target is added or deleted or when the size of a LUN changes.

The asynchronous event generation occurs in addition to any notifications issued using the SCSI programming model, such as CHECK CONDITION on a subsequent I/O. When the AVIO storage driver on the vPar or VM guest detects the events, it takes appropriate actions, such as discovering the new targets. For example, if new targets are added using the `hpvmmodify -a` command, then the vPar or VM guest driver automatically detects the new device without the manual scan. The vPar or VM guest automatically detects any modification to the underlying backing storage.

To avoid damage to vPar or VM guest, you must change the underlying backing storage on an existing vPar or VM guest device when it is not running. If the change is to a running vPar or VM guest, the administrator must ensure that the change will not adversely affect the health of the running environment. Although, HP-UX vPars and Integrity VM does check to determine if the device is in use, those checks are not reliable, because the vPar or VM guest might or might not be using the device at the time it is checked.

Backing storage can be adversely affected if the actual storage or access path is modified directly by an HP-UX server command, for example, by removing a file backing store or unmounting the file system. If the devices being changed are a result of some SAN reconfiguration, you must run the `ioscan` command on the VSP before attempting the change with the `hpvmmodify` command. If the backing storage is changed by remapping a different wwid to an existing dsf using `scsimgr replace_wwid -D dsf`, you must run the `hpvmdevmgmt -I` command. If the backing storage is SAN presented as a different device and the change is done using `io_redirect_dsf -d old_dsf -n new_dsf`, the vPar or VM guest must be modified using the `hpvmmodify` command to reference the new disk in place of the old disk.

NOTE: When a SLVM LV is configured as a backing store for a vPar or VM guest, any changes made to the LV size, will be automatically propagated to the vPar or VM guest only if it is running on the cluster node configured as the `server` for the volume group to which the LV belongs. Restarting the guest reflects the modified LV size.

Virtual storage setup time

Some virtual devices take longer to set up than others. Whole disks are very easy to set up because they require nothing more than a character device file. This is usually created automatically when the VSP system is booted.

Logical volume creation is simple. Logical volumes are used widely on HP-UX systems. The Veritas Enterprise Administrator, the HP-UX Logical Volume Manager commands, or the SMH can be used to create logical volumes. With experience, you can use logical volume commands quickly.

Creating files for virtual devices is not hard, but takes time. Files are usually placed on top of logical volumes, so you might have to create a logical volume first.

To create empty files for virtual disks, use the `hpvmdevmgmt` command (see “Managing the device database” (page 270)).

To create ISO files from physical CD or DVD media for use in virtual DVDs, use the *mkisofs* or the *dd* utility.

NPIV brings in ease of storage provisioning because storage presentation does not have to be a two-step process (first, presenting the LUNs to the VSP and then assigning each one to the vPar and VM guest). With NPIV HBAs, storage provisioning for a vPar or VM guest is the same as for a standalone system. This differentiates it from legacy AVIO storage.

Sample script for adding multiple devices at once

If using NPIV is not an option to add 256 AVIO storage devices to a vPar or VM guest, Hewlett Packard Enterprise recommends that you use the `hpvmcreate` and `hpvmmodify` commands to add multiple devices at a time using multiple `-a` options. Adding multiple devices at a time takes less time than adding them one at a time, with one device per call to `hpvmcreate` command and then one device per call in subsequent calls to `hpvmmodify` command.

You can add any number of devices at a time up to the supported limit. However, you might find that adding multiple devices at a time per call to `hpvmmodify` command not only takes less time than adding all of them at once, but also using one particular number of devices at a time provides better `hpvmmodify` command performance than others. For example, if you are adding a total of 256 disks, adding 64 at a time might provide better performance than adding 8 at a time and better performance than adding 128 at a time. The best number to use might vary depending on many factors including how many total devices you are adding.

For more information about the sample script for adding multiple devices, see Appendix C (page 307).

Setting up virtual storage

When you add or modify a virtual device, you must enter a resource statement (*rsrc*). The resource statement can specify either virtual network devices (as described in "Creating virtual and direct I/O networks" (page 122) and *hpvmresources*(5)), or virtual storage devices.

The resource statement specifies the virtual storage device that will be seen by the vPar and VM guest and how it maps to the physical storage device on the VSP.

The following is an outline of a complete resource statement for specifying a virtual storage device:

VM-vpar-storage-specification:*VM-Host-storage-specification*

where:

VM-vpar-storage-specification

defines where and what storage is seen in the vPar and VM guest (see "Storage specification" (page 74)).

VM-Host-storage-specification

defines where and how the vPar and VM guest storage is supplied on the VSP (see "VSP storage specification" (page 75)).

For examples of how to construct resource statements, see "Storage resource statements" (page 76).

Storage specification

All virtual storage is addressed from virtual PCI buses. The vPar and VM guest virtual platform contains 8 PCI buses. Each PCI bus has 8 slots into which virtual PCI adapters can be placed. An AVIO storage adapter supports up to 128 devices and provides high performance and guest storage manageability.

A VSP administrator specifies this virtual adapter using the following:

device:avio_stor:*pcibus*,*pcislot*,*target*

where:

- *device* is one of the following:

  `disk`, `dvd`, `tape`, `changer`, `burner`, or `hba`

- *pcibus* is an integer from 0-7.

  It represents the PCI bus number for the virtual device.

- *pcislot* is an integer from 0-7.

  pcislot also referred to as the *pcidevice*, represents the PCI slot number for the virtual device. A PCI function number is not specified. It is implicitly zero because the virtual storage adapter supports only a single channel.

- *target* is an integer from 0–127 for AVIO. This is applicable only for non-NPIV backing stores. All supported non-NPIV storage device types can share the same virtual AVIO adapter by specifying the same PCI bus and slot numbers.

  All targets connected to a vPar and VM guest are single LUN devices. That is, all virtual devices are emulated as single LUNs. All virtual LUN numbers are implicitly zero and therefore not specified.

A virtual adapter can only be added to a vPar or VM guest if it has a device connected to it, with the exception of NPIV HBAs, where you can add a NPIV HBA to a vPar or VM guest without presenting LUNs to it.

Not all device types are virtualized. Disk and DVD devices are virtual device types, whose virtual media comes from the VSP. Tapes, changers, and burners are physical VSP devices. For these attached devices, the physical IDs do not determine their place on the virtual bus.

NOTE: Certain PCI slots are used by vPars and VM guests for special devices. You can use the `hpvmstatus -P <guest_name> -V` command to get a list of reserved slots. For more information about dynamic addition of IO devices to vPars and VM guests, see "Dynamic I/O for vPars and Integrity VM guests" (page 269).

VSP storage specification

Each vPar and VM guest storage device is backed by some VSP storage entity. A VSP entity is defined on the VSP with a system file, which is used by vPars and Integrity VM and the VSP operating system in processing I/O to and from that storage entity.

A VSP administrator specifies these storage entities using the following specification:

storage:*location*

where:

- *storage* is one of the following:

  `disk`, `lv`, `file`, `null`, `attach_path`, or `npiv`.

  The selection of storage type defines what VSP system files apply. For example, `lv` implies the use of logical volume character device files.

  For virtual devices, the selection of VSP storage determines what type of virtual media the virtual device uses. For example, the selection of `lv` for a virtual disk, makes it a Virtual LvDisk to the VM.

  A VSP storage entity can only be used for one VM device type at a time. For example, a VSP CD or DVD drive cannot be used for a Virtual DVD and an attached burner at the same time.

- *location* is a VSP system file.

  The file permissions on the VSP system file or HW path for attach_path devices are not honored by vPars and Integrity VM. vPar and VM guest device types that support write operations can still do so using a VSP system file marked read only. Backing stores provided as virtual disks can be written to regardless of the file permission settings on the backing store. A backing store provided as a virtual DVD is always read-only. Attached devices do not consider file permissions when backing up data.

  More than one VSP system file might point to the same VSP storage entity. For example, if multiple paths to storage are present on the VSP, more than one disk system file can point to the same disk. Different VSP system files change how I/O is routed to the vPar or VM storage resource, but the system files point to the same storage entity. Therefore, different system files cannot constitute different vPar or VM guest storage resources. A given storage resource can only be specified once to a given vPar or VM guest. Therefore, only one VSP system file per VSP storage entity can be provided to a vPar or VM guest (see "Storage management" (page 66)).

Not all virtual device types support all VSP storage types (see "vPar and VM guest storage implementations" (page 62)). The next section discusses the VM storage resource statements.

Storage resource statements

This section provides information about formulating complete valid resource statements for vPar and VM guest storage devices.

To specify a storage device for a vPar or VM guest, use a complete valid resource statement with the `hpvmcreate` or `hpvmmodify` command. The resource statement is a combination of the vPar and VM guest resource specification (described in "Storage specification" (page 74)) and the VSP Storage Specification (described in "VSP storage specification" (page 75)). This section provides examples of complete resource statements for each of the following types of virtual storage devices:

- Virtual disks
- Virtual LvDisks
- Virtual FileDisks
- Virtual DVDs
- Virtual FileDVDs
- Virtual NullDVDs
- Attachable Devices

NOTE: For more information about resource statement for an NPIV HBA, see "Configuring an NPIV HBA (vHBA)" (page 100).

A vPar or VM guest can have up to 256 non-NPIV devices (number of virtual and attached devices).

The minimum size of a virtual storage resource is 512 bytes for virtual disk and 2048 bytes for a virtual DVD.

Do not specify the same storage resource, virtual or attached, for the same vPar or VM guest more than once (see "Storage management" (page 66)). Unless otherwise noted, storage resources, virtual or attached, cannot be simultaneously shared by vPars and VM guests.

The resource statements in the following subsections do not contain vPar or VM guest hardware addressing. The PCI bus, PCI slot, and AVIO target numbers are optional.

Virtual Disks

A Virtual Disk is an emulated AVIO disk whose virtual media comes from a VSP disk LUN. The VSP disk LUN is specified using a character device file. The character device file is owned by the HP-UX `esdisk` driver.

Virtual Disk resources cannot be shared simultaneously across active vPars and VM guests (except in certain cluster configurations, as indicated in this document). Virtual Disk resources can be changed dynamically among active vPars and VM guests.

To prevent virtual media conflicts that can result in data damage, a proper accounting of how the VSP whole disks are allocated for use by Virtual Disks needs to be done, as described in "Storage management" (page 66).

The following is the Virtual Disk resource statement form:

`disk:avio_stor::disk:`*`/dev/rdisk/diskX`*

where `/dev/rdisk/diskX` is an HP-UX `esdisk` character device file.

These device files can be located for a VSP LUN using the `ioscan` command.

For example:

```
# ioscan

# ioscan -NfunC disk

disk        64000/0xfa00/0x10   esdisk    CLAIMED     DEVICE
HP      HSV210
/dev/disk/diskX   /dev/rdisk/diskX
```

These system files are installed and removed using the `insf` and `rmsf` commands, respectively. Device files are created automatically by the VSP for any storage it identifies during boot. New devices connected or created after boot time, may require the use of `ioscan` and `insf` commands to create the new device files. To remove old device files for storage, use the `rmsf` command.

A VSP disk LUN can also be specified using the corresponding Cluster DSF in the virtual disk resource statement.

The following is the Virtual Disk resource statement form using cDSF:

`disk:avio_stor::disk:`*`/dev/rcdisk/diskX`*

where `/dev/rcdisk/diskX` is an HP-UX `esdisk` character cluster device special file.

These device files can be located for a VSP LUN using the `ioscan` command.

For example:

```
# ioscan

# ioscan -NfunC disk

disk     15  64000/0xfa00/0x4  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME

 /dev/cdisk/diskX   /dev/disk/diskY   /dev/rcdisk/diskX  /dev/rdisk/diskY
```

The `/dev/rdisk/diskY` is the corresponding device special file name for Cluster device special file name `/dev/rcdisk/diskX`. The mapping between the two can be viewed using the `ioscan` command.

For example:

```
# ioscan -m cluster_dsf


Cluster DSF            Persistent DSF        Legacy DSF(s)
=============================================================
/dev/rcdisk/diskX      /dev/rdisk/diskY      /dev/rdsk/c?t?d?
```

For example:

```
# cmsetdsfgroup -n <VSP-node1> -n <VSP-node2>
```

NOTE: If the VSP node is removed from the cDSF group then the cluster device special files must be removed manually from that node.

These cluster device special files are installed and created using the `cmsetdsfgroup(1M)` command. See *cmsetdsfgroup*(1M) manpage.

Cluster devices special files can be used as a backing store by the guests only if the Cluster DSF (cDSF) feature is enabled on the VSP, that is, the VSP is a part of Cluster DSF group.

Before using cDSF as backing store, confirm that if the VSP is part of Cluster DSF group.

```
# hostname
hpidm01-3
# cmsetdsfgroup -q
```

```
bones
hpidm01-3
#
```

Virtual LvDisks

A Virtual LvDisk is an emulated AVIO disk whose virtual media is provided by a raw VSP logical volume. To specify a VSP logical volume, use a character device file. The character device file is owned by either LVM or VxVM.

Virtual LvDisks cannot be shared simultaneously across active vPars and VM guests. Virtual LvDisk resources can be changed dynamically between active vPars and VM guests (see "Using vPars and Integrity VM storage" (page 91)).

Logical volumes can be created using the `sam` utility or the Veritas Enterprise Administrator. Alternatively, logical volumes can be created using the commands available with the volume manager. All logical volumes are created on whole disks. The sizes of the logical volumes come from the space available from their respective volume group types; the logical volume size can be increased without loss of data in the volume. The character devices for the logical volumes are created by their respective volume managers at the time the logical volume is created. Also to avoid file system corruptions for the VSP and guest, use only raw logical volumes that do not contain VSP file systems, and are not currently mounted on the VSP.

To prevent data from getting over written, keep an account of logical volumes for Virtual LvDisks. To make accounting easier, all logical volumes within a volume group can be assigned to a single guest. When logical volumes are configured this way, you only have to keep track of the volume groups to prevent media conflicts. For more information about tracking virtual media allocation, see "Storage management" (page 66).

If you are using LVM, the following is the Virtual LvDisk resource statement form:

`disk:avio_stor::lv:`*`/dev/vg_name/rlvol_name`*

`vgdisplay -v`

where *`/dev/vg_name/rlvol_name`* is an LVM character device file for *`rlvol_name`* on *`vg_name`*. To view the LVM character device file name, enter the following command:

```
# vgdisplay -v
VG Name                     /dev/lvrackA
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      4
Open LV                     4
Max PV
Cur PV                      1
Act PV                      1
Max PE per PV               8683
VGDA                        2
PE Size (Mbytes)            4
Total PE                    8681
Alloc PE                    8192
Free PE                     489
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0

   --- Logical volumes ---
   LV Name                     /dev/lvrackA/disk1
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  2048
   Allocated PE                2048
   Used PV                     1
```

```
   LV Name                     /dev/lvrackA/disk2
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  2048
   Allocated PE                2048
   Used PV                     1

   LV Name                     /dev/lvrackA/disk3
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  2048
   Allocated PE                2048
   Used PV                     1

   LV Name                     /dev/lvrackA/disk4
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  2048
   Allocated PE                2048
   Used PV                     1

   --- Physical volumes ---
   PV Name                     /dev/disk/disk237
   PV Status                   available
   Total PE                    8681
   Free PE                     489
   Autoswitch                  On
```

In this example, the Virtual LvDisk resource statement form is `disk:avio_stor::lv:/dev/lvrackA/rdisk2`.

To use VxVM, the following is the Virtual LvDisk resource statement form:

`disk:avio_stor::lv:`*`/dev/vx/rdsk/dg_name/v_name`*

where *`/dev/vx/rdsk/dg_name/v_name`* is a VxVM character device file for volume *`v_name`* on disk group *`dg_name`*. To view the VxVM character device file name, enter the following command:

```
# vxprint
Disk group: rootdg

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg rootdg       rootdg       -        -        -        -        -       -

dm disk01       c3t0d0       -        35562538 -        -        -       -

Disk group: VxvmTest1

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg VxvmTest1    VxvmTest1    -        -        -        -        -       -

dm disk01       c5t8d0       -        780564 -        -        -       -

v  vxvm_1       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_1-01    vxvm_1       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-01    vxvm_1-01    ENABLED  2048000  0        -        -       -

v  vxvm_2       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_2-01    vxvm_2       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-02    vxvm_2-01    ENABLED  2048000  0        -        -       -

v  vxvm_3       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_3-01    vxvm_3       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-03    vxvm_3-01    ENABLED  2048000  0        -        -       -
```

```
v  vxvm_4       fsgen        ENABLED  2048000  -        ACTIVE   -        -
pl vxvm_4-01    vxvm_4       ENABLED  2048000  -        ACTIVE   -        -
sd disk01-04    vxvm_4-01    ENABLED  2048000  0        -        -        -
```

To use VxVM, the Virtual LvDisk resource statement form is `disk:avio_stor::lv:/dev/vx/rdsk/VxvmTest1/vxvm_2`.

For information about multipath solutions for Virtual LvDisks, see "Storage multipath solutions" (page 66).

Virtual FileDisks

A Virtual FileDisk is an emulated AVIO disk, which uses the VSP file as a virtual media. The VSP file is specified using the absolute pathname to the file. The file can be on a VxFS file system locally mounted on the VSP or files located on an NFS-mounted file system. For more information about configuration and requirements that must be met before using an NFS mounted file as a VM or vPar file backing store, see "NFS-Mounted backing stores" (page 88).

Virtual FileDisks cannot be shared at the same time across active VMs. Virtual FileDisk resources can be changed dynamically between active vPars and VM guests (see "Using vPars and Integrity VM storage" (page 91)).

The file systems used for Virtual FileDisks must be managed to prevent data from getting corrupted. To help with accounting, Hewlett Packard Enterprise recommends that all files under a given directory be used with a single vPar and VM guest. Additionally, it might help to allocate file directories from complete logical volumes or whole disks to make the accounting even easier. For more information, see "Storage management" (page 66).

Following is the Virtual FileDisk resource statement form:

`disk:avio_stor::file:`*`/pathname/file`*

where *`/pathname/file`* specifies the VSP file used as virtual media.

A VxFS file system can be created on top of a whole disk or logical volume. For files over 2 GB, VxFS requires the file system be marked with a `largefiles` option. You can use the `mkfs` command to create the VxFS file systems directly. After the file systems are created, you can use the `mount` command to mount them onto the VSP file system. Alternatively, if you use logical volumes to create the file system, you can use the volume manager GUI such as HP SMH to create the file systems and their mount points, when the logical volumes are created. After the file system is mounted, you can create empty files for Virtual FileDisk using the `hpvmdevmgmt` command.

```
# mkfs -F vxfs -o largefiles /dev/disk/disk237
# mount /dev/disk/disk237 /fdev/frackA/
# hpvmdevmgmt -S 4G /fdev/frackA/disk1
```

In this example, the Virtual FileDisk resource statement form is `disk:avio_stor::file:/fdev/frackA/disk1`.

For more information about multipath options for a Virtual FileDisk device, see "Storage multipath solutions" (page 66).

NOTE: Each vPar or VM guest can support a maximum of 30 Virtual FileDisks.

Virtual DVDs

A Virtual DVD is an emulated AVIO DVD-ROM with virtual media that comes from a disc inside of a CD or DVD drive on the VSP. The VSP CD or DVD drive is specified using an HP-UX `esdisk` character device file.

While the Virtual DVD is read-only, the slowness of the physical VSP CD or DVD drives prohibits them from being shared across active vPars and VM guests. Thus only one active vPar and VM guest at a time must be given a particular Virtual DVD resource. Virtual DVD resources can be changed dynamically between active vPars and VM guests (see “Using vPars and Integrity VM storage” (page 91)).

Because the Virtual DVDs are read only, they do not require management to prevent conflicts writing to the device. However, to prevent sensitive information from being accessed by the wrong vPar or VM guest, ensure you know which vPar or VM guest currently owns the device before you load a CD or DVD. You can find this information on the VSP using the `hpvmstatus` command.

Following is the agile Virtual DVD resource statement form:

`dvd:avio_stor::disk:`*`/dev/rdisk/disk#`*

where `/dev/rdisk/disk#` is an HP-UX `esdisk` character device file for a VSP CD or DVD drive.

Typically, the HP-UX `esdisk` character files is already created before booting the VSP. If they are not, they can be created and managed using the `ioscan`, `insf`, and `rmsf` utilities. For example,

```
# ioscan -NfunC disk

disk        7   64000/0xfa00/0x6    esdisk    CLAIMED      DEVICE
TEAC       DW-224E
                /dev/disk/disk7   /dev/rdisk/disk7

# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: TEAC
         product id: DW-224E
               type: CD-ROM
               size: 4300800 Kbytes
   bytes per sector: 2048
```

In this example, the Virtual DVD resource statement form is `dvd:avio_stor::disk:/dev/rdisk/disk7`.

For a vPar and VM guest to recognize a Virtual DVD, physical media must be present inside the VSP CD or DVD drive. If media is not added at vPar and VM guest start time, it can be inserted into the VSP CD or DVD drive after the vPar and VM guest is already up. A rescan by the guest OS picks up the new media and adds the Virtual DVD to the vPar and VM guest.

If the VSP Administrator requires control of the VSP CD or DVD drive claimed by a vPar or VM guest but has no media for the VSP CD or DVD drive, then a Virtual NullDVD must be specified (see “Virtual NullDVDs” (page 83)). Physical media can then be inserted into the VSP CD or DVD drive and become virtual media for a Virtual DVD using the `hpvmmodify` command or the virtual console's `insert` command (see “Guest administrator” (page 91)).

After the Virtual DVD is in the vPar or VM guest, the VSP CD or DVD drive is locked. The VSP CD or DVD drive is automatically unlocked when the vPar or VM guest is shut down. The VSP CD or DVD can also be changed while the vPar or VM guest is up using the `eject` command of the virtual console. After ejected, the Virtual DVD turns into a Virtual NullDVD and the VSP CD or DVD drive unlocks. After you place physical media in the VSP's CD or DVD drive, use the virtual console's `insert` command to turn a Virtual NullDVD back to a Virtual DVD, relocking the VSP CD or DVD drive.

CAUTION: If the Virtual DVD drive of the guest is backed by a CD or DVD-ROM in the VSP that is either an enclosure DVD-ROM or is assigned via vMedia, then the following exceptions apply:

- A vPar or VM guest configured with a virtual DVD that is backed by such a CD or DVD device in the VSP fails to start up if the device is disconnected when the vPar or VM is being started.
- For such CD or DVD-ROMs, a media eject operation works like a drive disconnect, and hence, the media eject operation succeeds irrespective of its usage by the VSP or by any of the active vPars or VM guests.

Such devices can be identified by looking for “Virtual CD-ROM” or “Virtual DVD-ROM” in the device description provided by the `ioscan` command.

For example, the ioscan output for an enclosure DVD-ROM:

```
#ioscan -funC disk
Class   I  H/W Path     Driver  S/W State   H/W Type     Description
==========================================================================
disk    4  255/1/0.0.0  sdisk   CLAIMED     DEVICE       HP Virtual DVD-ROM
             /dev/dsk/c21t0d0   /dev/rdsk/c21t0d0
```

Most physical VSP CD or DVD devices on HPE Integrity servers have only one path to them, as multipath software is not available on the VSP for them.

Virtual FileDVDs

A Virtual FileDVD is an emulated SCSI DVD, which uses a VSP ISO file as virtual media. The VSP ISO file is specified using the absolute pathname to the ISO file. The file has to be on a VxFS file systems locally mounted on the VSP. NFS file systems are not supported for Virtual FileDVDs.

Following is the Virtual FileDVD resource statement form:

`dvd:avio_stor::file:`*`/pathname/file.ISO`*

where *`/pathname/file.ISO`* specifies the VSP ISO file to use as virtual media.

You can create a VSP ISO file using the `mkisofs` utility or by using the `dd` command to copy CD or DVD media to a file. The VxFS file system should be enabled to support `largefiles`, because ISO files tend to be over 2 GB in size. All the ISO files that are useful to a guest OS should be placed in the same directory to take advantage of dynamic changes using the virtual console (see “Modifying storage devices” (page 95)). The ISO files must be marked with proper permissions; they must not be world writable. For example,

```
# ls -l /var/opt/hpvm/ISO-images/hpux

total 26409104
-rw-r--r-- 1 root sys 3774611456 Jul 11 :59 0505-FOE-OE.iso
-rw-r--r-- 1 root sys 4285267968 Jul 11 17:05 0512-FOE.iso
-rw-r--r-- 1 root sys 3149987840 Jul 11 18:42 0603-FOE-D1.iso
-rw-r--r-- 1 root sys 29978624 Jul 11 18:51 0603-FOE-D2.iso
```

In this example, the Virtual FileDVD resource statement form is:
`dvd:avio_stor::file:/var/opt/hpvm/ISOimages/hpux/0603-FOE-D1.iso`.

Virtual FileDVDs, such as all files, can take advantage of the multipath options with which the file system is created. For more information, see “Storage multipath solutions” (page 66).

Virtual FileDVDs are read-only and are shareable across active VMs. You can use the `hpvmdevmgmt` command to mark them as sharable.

To prevent media conflicts, you must manage Virtual FileDVDs (see “Storage management” (page 66)). You can know the location of the file system directory where the ISO file resides using the virtual console of the guest. To simplify accounting, you can allocate file directories from complete logical volumes or whole disks.

A Virtual FileDVD reverts to its original resource statement when the guest shuts down or reboots. Therefore, after you install a guest from multiple CDs or DVDs, you must reload the Virtual FileDVD when the guest reboots to complete the installation. Stop the automatic EFI reboot and insert the CD or DVD using the appropriate `IN` and `EJ` commands. When the media is loaded, you can proceed with the installation.

NOTE: The `hpvmmodify` command might fail to change a Virtual FileDVD if the device is modified by the virtual console. The `hpvmstatus` command displays the current status of the Virtual FileDVD, which might not be in its original resource state. To see the original resource statement, required by the `hpvmmodify` command to change a Virtual FileDVD, use the `hpvmstatus -D` command.

Virtual NullDVDs

A Virtual NullDVD is an emulated SCSI DVD-ROM with no virtual media present. The next media selection might come from a VSP CD or DVD drive or VSP ISO file, depending on how the Virtual NullDVD is configured. After the next media is selected, the Virtual NullDVD turns into either a Virtual DVD (see “Virtual DVDs” (page 80)) or a Virtual FileDVD (see “Virtual FileDVDs” (page 82)) device. As such, a Virtual NullDVD is a transitory state of an empty virtual DVD type.

The choice of how to configure a Virtual NullDVD depends on the access that the VSP administrator gives to the guest administrator. Virtual DVD changes can be initiated from the virtual console (see “Guest administrator” (page 91)). All virtual DVD changes by the guest administrator are constrained by the actions of the VSP administrator.

If the VSP administrator gives access to the guest administrator to load and unload physical media on the VSP CD or DVD drive, the Virtual NullDVD can be set up with the following form of the resource specification:

`dvd:avio_stor::null:`*`/dev/rdisk/disk#`*

where *`/dev/rdisk/disk#`* is an HP-UX `esdisk` character device file that points to the VSP CD or DVD drive.

This is the same as setting up a Virtual DVD (see “Virtual DVDs” (page 80)), except that the VSP CD or DVD might not contain media. The media is expected to come from the guest administrator, who should have access to the VSP to make such physical media changes. For example,

```
# ioscan -NfunC disk

disk       7   64000/0xfa00/0x6    esdisk    CLAIMED     DEVICE
TEAC      DW-224E
             /dev/disk/disk7   /dev/rdisk/disk7
# diskinfo /dev/rdisk/disk7

SCSI describe of /dev/rdisk/disk7:
             vendor: TEAC
         product id: DW-224E
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0
```

In this example, the Virtual NullDVD resource statement is `dvd:avio_stor::null:/dev/rdisk/disk7`.

If the VSP administrator does not want to give rights to the guest administrator to access the VSP CD or DVD drive, you can set up a Virtual NullDVD to a file system directory containing the ISO files that the guest administrator wants to access. Following is the resource statement form:

`dvd:avio_stor::null:`*`/pathname`*

where */pathname* is the file system directory where the ISO files are located.

This is the same as setting up a Virtual FileDVD (see “Virtual FileDVDs” (page 82)), except that the file is not specified. By specifying a file directory, the guest administrator can choose the ISO files to use from the virtual console. The file directory must be a locally mounted VxFS file system. NFS file systems are not supported. If the ISO files are world writable, they are not available from the virtual console for the ISO files listed.

```
# ls -l /var/opt/hpvm/ISO-images/hpux

total 26409104
-rw-r--r-- 1 root sys 3774611456 Jul 11 :59 0505-FOE.iso
-rw-r---r-- 1 root sys 4285267968 Jul 11 17:05 0512-FOE.iso
-rw-r---r-- 1 root sys 3149987840 Jul 11 18:42 0603-FOE-D1.iso
-rw-r---r-- 1 root sys 29978624 Jul 11 18:51 0603-FOE-D2.iso
```

The Virtual NullDVD resource statement form is
`dvd:avio_stor::null:/var/opt/hpvm/ISO-images/hpux/`.

You can configure the Virtual NullDVD to be sharable or have multipath options. If the Virtual NullDVD device is configured to use the VSP CD or DVD device, it is not sharable and no multipath options are available. If the Virtual NullDVD is configured to use a file system directory, it is sharable and you can use multipath options (see “Storage multipath solutions” (page 66)). To mark the directory sharable across VMs, you can use the `hpvmdevmgmt` command. For example,

```
# hpvmdevmgmt -m gdev:/var/opt/hpvm/ISO-images/hpux/:attr:SHARE=YES
```

For more information about using the `hpvmdevmgmt` command, see “Managing the device database” (page 270).

Virtual NullDVDs require no additional management beyond that required for the Virtual DVD (see “Virtual DVDs” (page 80)) or Virtual FileDVD (see “Virtual FileDVDs” (page 82)) types.

Attachable devices

vPars and Integrity VM allows you to attach physical VSP backup device types to vPars or VM guests. VSP backup device types are tapes, media changers, and CD or DVD burners. These devices are specified on the VSP using their respective lunpath hardware path (displayed only in `ioscan` with the `-N` option). For more information about how to find lunpath hardware path for a given physical device, see “Finding the lunpath hardware path” (page 85).

The guest OS running on the vPar and VM guest has full control over an attached physical device.

Following are the resource statement forms for attached devices depending upon the device type:

- For magnetic tape:

  `tape:avio_stor::attach_path:`*lunpath_hardware_path*

- For media changers:

  `changer:avio_stor::attach_path:`*lunpath_hardware_path*

- For CD or DVD burners:

  `burner:avio_stor::attach_path:`*lunpath_hardware_path*

The following example shows the resource specifier for an attached tape device:

`tape:avio_stor:0,4,0:attach_path:0/7/1/1.0x500104f00048b29e.0x0`

For more information about attached I/O support and configuration, see “Attached device support” (page 85).

As with virtual devices, attached devices can be attached and detached dynamically across active vPars or VM guests (see “Using vPars and Integrity VM storage” (page 91)). Also, while the device is being attached to a vPar or VM guest, it cannot be opened by the VSP at the time of or during attachment.

Because tapes, media changers, and CD or DVD burners are not virtualized, media changes with these must be done physically. Therefore, all media changes with attached devices must be done by individuals with access to that physical storage. Changes to attached devices might require the device to be unlocked from an active guest OS. Attached devices remain in the last lock state the guest OS put it in when the device is detached or the VM is shut down. Empty devices are attached and are not locked.

Multipath solutions are not available for attached devices on the VSP. Multipath products are not supported in the vPar or VM guest.

Manage attached devices to prevent the wrong vPars and VM guests from viewing sensitive information. You can find the vPars or VM guests that are currently using attached devices using the `hpvmstatus` command.

Attached device support

Attached devices allow sharing of tapes, changers, and burners among multiple guests and host, support for USB 2.0 DVD burners and improves performance.

To identify USB CD or DVD devices, use the `ioscan -funN` command.

NOTE: vPars and VM guest might do four to six calls to `open()` on a DVD when accessing it and `hpvmcreate` or `hpvmmodify` command might take more than a minute to complete when there is no media in the drive. Example commands that appear to hang are:

```
# hpvmcreate -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmcreate -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
```

Finding the lunpath hardware path

To obtain the lunpath hardware path to configure an attached device, use the `ioscan` command with the `-m lun` option. For example, in this case of a tape having two paths the `ioscan` output is:

```
# ioscan -m lun /dev/rtape/tape1_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
======================================================================
tape      1  64000/0xfa00/0x0   estape  CLAIMED     DEVICE       online  STK     T9940B
             0/1/1/1.0x500104f00048b29d.0x0
             0/7/1/1.0x500104f00048b29e.0x0
                      /dev/rtape/tape1_BEST    /dev/rtape/tape1_BESTn
                      /dev/rtape/tape1_BESTb   /dev/rtape/tape1_BESTnb
```

You can use the `ioscan` command to find the device special file corresponding to a lunpath hardware path. For example, in the previous case, to find the device special file for lunpath hardware path 0/7/1/1.0x500104f00048b29e.0x0, run the following `ioscan` command:

```
# ioscan -kfnNH 0/7/1/1.0x500104f00048b29e.0x0

Class      I   H/W Path  Driver               S/W     State     H/W Type      Description
=========== ======================== ====== ======== =========== ===========
lunpath  21  0/7/1/1.0x500104f00048b29e.0x0  eslpt   CLAIMED   LUN_PATH      LUN path for tape1
```

The DSF for tape1 is /dev/rtape/tape1_BEST*.

Sharing an attached device

Attached devices can be shared among multiple vPars and VM guests in a VSP using a single physical HBA port (initiator) or multiple physical HBA ports (initiators) in the VSP. To share a tape device:

1. Identify the tape devices:

   ```
   # ioscan -funNC tape
   Class     I  H/W Path  Driver S/W State   H/W Type     Description
     ===================================================================
   ```

```
  tape      5  64000/0xfa00/0x1  estape   CLAIMED     DEVICE       HP      Ultrium 3-SCSI
                      /dev/rtape/tape5_BEST    /dev/rtape/tape5_BESTn
                      /dev/rtape/tape5_BESTb   /dev/rtape/tape5_BESTnb
  tape      6  64000/0xfa00/0x3  estape   CLAIMED     DEVICE       STK     T9840B
                      /dev/rtape/tape6_BEST    /dev/rtape/tape6_BESTn
                      /dev/rtape/tape6_BESTb   /dev/rtape/tape6_BESTnb
```

2. This system has two tape drives. Identify the lunpaths:

```
# ioscan -m lun /dev/rtape/tape5_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
======================================================================
tape      5  64000/0xfa00/0x1   estape  CLAIMED     DEVICE       online  HP      Ultrium 3-SCSI
              0/5/0/0/0/0.0x500110a0008b9de2.0x0
                       /dev/rtape/tape5_BEST    /dev/rtape/tape5_BESTn
                       /dev/rtape/tape5_BESTb   /dev/rtape/tape5_BESTnb

# ioscan -m lun /dev/rtape/tape6_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
======================================================================
tape      6  64000/0xfa00/0x3   estape  CLAIMED     DEVICE       online  STK     T9840B
              0/4/1/0.0x500104f0004732d9.0x0
              0/4/1/1.0x500104f0004732d9.0x0
              0/4/1/0.0x500104f0004732da.0x0
              0/4/1/1.0x500104f0004732da.0x0
                       /dev/rtape/tape6_BEST    /dev/rtape/tape6_BESTn
                       /dev/rtape/tape6_BESTb   /dev/rtape/tape6_BESTnb
```

Device `tape5` is connected to the VSP using a single HBA port (initiator). It has one lunpath through initiator (0/5/0/0/0/0). Device `tape6` is connected to the VSP using two HBA ports (initiators). It has four lunpaths through two initiators (0/4/1/0 and 0/4/1/1).

Example 2 Example of sharing a tape device using a single initiator (single lunpath):

```
# hpvmmodify -P guest1 -a tape:avio_stor::attach_path:0/5/0/0/0/0.0x500110a0008b9de2.0x0
# hpvmmodify -P guest2 -a tape:avio_stor::attach_path:0/5/0/0/0/0.0x500110a0008b9de2.0x0
# hpvmdevmgmt -l gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0

0/5/0/0/0/0.0x500110a0008b9de2.0x0,lunpath1:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape5_BESTn:guest1,guest2:0x01.0x00.0x03.0x500110a0008b9de1_lunpath1

# hpvmdevmgmt -m gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0:attr:SHARE=YES
# hpvmdevmgmt -l gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0

0/5/0/0/0/0.0x500110a0008b9de2.0x0,lunpath1:CONFIG=gdev,EXIST=YES,SHARE=YES,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape5_BESTn:guest1,guest2:0x01.0x00.0x03.0x500110a0008b9de1_lunpath1
```

The `hpvmdevmgmt -m` command can also take the following form:

```
# hpvmdevmgmt -m gdev:lunpath1:attr:SHARE=YES
```

where "lunpath1" is the vPars and Integrity VM- generated alias for the hardware path. The vPar and VM guest-generated alias of the form "lunpath#" can be used as shorthand in device management commands, but it cannot be used in the `hpvmcreate` or `hpvmmodify` commands.

Example 3 Example of sharing a tape device using different initiators (different lunpaths):

1. Add different paths to each vPar and VM guest:

   ```
   # hpvmmodify -P guest1 -a tape:avio_stor::attach_path:0/4/1/0.0x500104f0004732d9.0x0
   # hpvmmodify -P guest2 -a tape:avio_stor::attach_path:0/4/1/1.0x500104f0004732d9.0x0
   ```

   Note that the two lunpath hardware paths in the previous example are through two different initiators (0/4/1/0/ and 0/4/1/1/).

2. List the attributes of each path (Note the value of the `AGILE_DSF` attribute is the same for both lunpaths.):

   ```
   # hpvmdevmgmt -l gdev:0/4/1/0.0x500104f0004732d9.0x0

   0/4/1/0.0x500104f0004732d9.0x0,lunpath3:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
   /dev/rtape/tape6_BESTn:vme01,guest1:0x01.0x00.0x03.0x500104f0004732d8_lunpath3

   # hpvmdevmgmt -l gdev:0/4/1/1.0x500104f0004732d9.0x0

   0/4/1/1.0x500104f0004732d9.0x0,lunpath4:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
   /dev/rtape/tape6_BESTn:guest2:0x01.0x00.0x03.0x500104f0004732d8_lunpath4
   ```

3. List the attributes of the parent tape DSF:

   ```
   # hpvmdevmgmt -l gdev:/dev/rtape/tape6_BESTn
   /dev/rtape/tape6_BESTn:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACH,SHARE_LUNPATHS=NO:
   lunpath3,lunpath6,lunpath5,lunpath4:0x01.0x00.0x03.0x500104f0004732d8
   ```

4. Modify the `SHARE_LUNPATHS` attribute:

   ```
   # hpvmdevmgmt -m gdev:/dev/rtape/tape6_BESTn:attr:SHARE_LUNPATHS=YES
   ```

   NOTE: The `SHARE_LUNPATHS` and `SHARE` attributes take effect only after running the `hpvmstop` command.

5. Relist the attribute of the parent tape DSF:

   ```
   # hpvmdevmgmt -l gdev:/dev/rtape/tape6_BESTn

   /dev/rtape/tape6_BESTn:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACH,SHARE_LUNPATHS=YES:
   lunpath3,lunpath6,lunpath5,lunpath4:0x01.0x00.0x03.0x500104f0004732d8
   ```

Patch dependency

Table 13 (page 88) lists the patch dependencies for the AVIO attached devices.

Table 13 Patch dependencies for AVIO attached devices

Patch Number	HP-UX Version	VSP	Guest	Notes
PHKL_38604	11i v3	Yes	Yes	Hard[1] dependency for guest, and soft[2] dependency for VSP.
PHKL_38605	11i v3	Yes	No	Soft dependency on VSP.
PHKL_38750	11i v3	Yes	Yes	Recommended patch.

1 Enforced during `swinstall`.

2 Required only if attached devices are configured. No enforcement using `swinstall`.

NFS-Mounted backing stores

vPars and Integrity VM supports NFS-mounted backing stores for use as root file system (that is, boot), swap, dump, and as data LUNs. The following configuration requirements apply for NFS mounted backing stores:

- NFS-mounted files cannot be used as file-backed virtual DVD drives.
- The maximum number of NFS-mounted backing stores per guest is four.
- NFS-mounted backing stores are supported only for HP-UX 11i v3 guests.
- The following NFS mount options must be used by the VSP when mounting an NFS file system housing the backing-store files of a guest:
  - NFS Version 3
  - TCP
  - Hard
  - IPv4 address or server host names mapping to IPv4 address
- The Integrity VSP (NFS client) and the NFS server systems must reside in the same IP subnet.
- OVMM is supported for VMs using NFS-mounted backing stores. For OVMM to be successful, both Integrity VSPs must mount the NFS file system housing the backing-store files of the guest using the identical syntax and mount options. Both the source and target VSPs must have the NFS file system mounted at the time of migration.

Following limitations apply to NFS-mounted backing stores:

- The use of symbolic links on the NFS server to redirect the location of the backing-store files of the guest is not allowed. However, symbolic links are still allowed inside the guest booted with an NFS backing store.
- Management of Integrity VM guests configured with NFS-mounted backing stores is not supported with the management applications. For more information about backing store requirements for individual products in the Matrix OE suite, see "Managing vPars and VMs using GUI" (page 277).

When creating NFS-mounted backing-store files, Hewlett Packard Enterprise recommends that you create these files locally on the NFS server, if possible. You can use either the `hpvmdevmgmt` command, if available on the NFS server, or the `dd` command. For example, to create an 80 GB file on an HP-UX NFS server as a guest backing store in the shared directory called `/export`, use either of the following commands:

```
/opt/hpvm/bin/hpvmdevmgmt -S 80G /export/vm1.boot
/usr/bin/dd if=/dev/zero of=/export/vm1.boot bs=1024K count=80000
```

If local access to the NFS server is not available, you can use these same commands on the VSP inside the NFS-mounted file system.

NOTE: Creating the backing-store files of the guest on an NFS client system (that is, VSP), can take significantly longer to complete than directly creating the backing-store files locally on the NFS server. Therefore, create backing-stores files of the guest directly on the NFS server, if possible.

Mapping AVIO storage devices on HP-UX guests

This section explains how to map an AVIO storage device on a vPar or VM guest to an `hpvmstatus` display on the Integrity VSP either at the EFI console or at the HP-UX operating system.

The following example shows the output of `hpvmstatus` from the Integrity VSP:

```
# hpvmstatus -P aviotest
[Storage Interface Details]
Guest                                Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
disk    avio_stor    0   2   0  22   0 disk      /dev/rdisk/disk7
```

The following statistics are displayed in this example:

- PciBus = 0
- PciDev = 2
- PciFtn = 0
- Addr (Target Id) = 22 (0x16)
- Lun = 0

NOTE: Addr (Target Id) is decimal in the `hpvmstatus` display, and PciFtn and LUN are always zero (0).

The vPar or VM guest EFI device path encodes PciBus, PciDev, and Addr (Target Id) from the `hpvmstatus` display:

```
                         PciDev
                            |
                            | PCIFtn
                  PciBus    | |       Addr(Target Id)
                     |      | |         |
                     V      V V         V
blk16 : Acpi(PNP0A03,0)/Pci(2|0)/Scsi(Pun16,Lun0)
```

PciFtn (PCI function) and LUN number are always zero (0). Addr (Target Id) becomes EFI Pun number and is displayed as a hexadecimal number.

Following are the two methods for mapping an HP-UX 11i v2 VM guest hardware path or HP-UX 11i v2 Device Special File (DSF) to an Integrity VSP `hpvmstatus` display:

1. `-e` option of the `ioscan` utility

   `ioscan-fne` displays the HP-UX hardware path/DSF and the EFI device path for the device. The HP-UX hardware path encodes the following from the `hpvmstatus` display:

   - PciBus
   - PciDev
   - Addr (Target Id)

   Addr (Target Id) is encoded as an HP-UX target ID and an HP-UX LUN ID in the HP-UX hardware path.

HP-UX target ID and HP-UX LUN ID are calculated from Addr (Target Id) in the `hpvmstatus` display using the following equations:

```
HP-UX tgt ID = Addr(Target Id) % 16
HP-UX lun ID = Addr(Target Id) / 16
```

Note the following example:

```
# ioscan -fne
                PciDev
                   | PCIFtn
                   | |(Addr(Target Id) % 16) <-> HP-UX tgt ID
            PciBus | | |(Addr(Target Id) / 16) <-> HP-UX lun ID
                 | | | | |
                 V V V V V
disk       49  0/0/2/0.6.1   esdisk     CLAIMED     DEVICE       HP      Virtual Disk
                            /dev/rdisk/disk7   /dev/rdisk/disk7
Acpi(PNP0A03,0)/Pci(2|0)/Scsi(Pun16,Lun0)
                     ^      ^ ^         ^
                     |      | |         |
                  PciBus    | PCIFtn   Addr(Target Id)
                            |
                     PciDev
```

In this example, exp1/exp2 represents the quotient from exp1 divided by exp2 (integer division), and exp1% exp2 finds modulo of exp1 divided by exp2 (that is, finds the remainder of an integer division).

2. `get_info` option of the `gvsdmgr` utility

   If you are using the HP-UX DSF, the following `gvsdmgr` option can be used to get the VSD LUN ID, which is the same as the Addr (Target Id) in the `hpvmstatus` display. The `gvsdmgr` utility displays VSD LUN Id as a hexadecimal number. The first nibble of VSD LUN ID becomes HP-UX LUN ID, and the second nibble becomes HP-UX target ID.

   The following example shows the `get_info` option with the `gvdsmgr` utility:

```
# gvsdmgr get_info -D /dev/gvsd0 -q lun=/dev/rdisk/disk7
Tue Oct  2 13:35:32 2007

Lun DSF                                           : /dev/rdisk/disk7
VSD LUN Id                                        : 0x16
Lun  Hardware path                                : 0/0/2/0.6.1
LUN State                                         : UNOPENED
```

The following is a method for mapping an HP-UX 11i v3 vPar or VM guest hardware path or HP-UX 11i v3 DSF to an Integrity VSP `hpvmstatus` display using the `ioscan` utility:

```
# ioscan -m dsf /dev/rdisk/c0t6d1
Persistent DSF Legacy DSF(s)
========================================
/dev/rdisk/disk22 /dev/rdisk/c0t6d1

# ioscan -m lun /dev/rdisk/disk22
Class I Lun H/W Path Driver S/W State H/W Type Health Description
======================================================================
disk 22 64000/0xfa00/0x1 esdisk CLAIMED DEVICE online HP Virtual Disk
0/0/2/0.0x16.1x0
/dev/disk/disk22 /dev/rdisk/disk22
/dev/disk/disk22_p1 /dev/rdisk/disk22_p1
/dev/disk/disk22_p2 /dev/rdisk/disk22_p2
/dev/disk/disk22_p3 /dev/rdisk/disk22_p3
```

An HP-UX 11iv3 lunpath hardware path displayed by the `ioscan` utility can be mapped to an `hpvmstatus` utility output as follows:

```
              PciDev
                   | PCIFtn
                   | | Addr(Target Id)
            PciBus | |  |     Lun
                 | | |  |     |
```

```
 V V V  V    V
0/0/2/0.0x16.1x0
```

Using vPars and Integrity VM storage

The following sections describe the roles of individuals accessing virtual storage, the commands they use, and some examples of using vPars and Integrity VM storage.

Storage roles

This section describes the roles of individuals in working with vPars or VM guests storage. Each role has different responsibilities in using vPars or VM guests storage. The roles might be played by one or more individuals depending on security requirements and skill sets. The three roles are:

- VSP administrator
- Guest administrator
- Guest user

For more information about creating vPar or VM guest administrator and operator accounts, see “Creating guest administrators and operators” (page 247).

VSP administrator

The VSP administrator is responsible for the proper configuration and maintenance of the VSP for running vPars and VM guests. As such, this person needs complete access to the VSP to install hardware and software. This person must also know about HP-UX system maintenance, hardware configuration, and setting up and using various software applications and tools.

The VSP administrator uses the following commands to manage vPar or VM guest storage devices:

Management function	Integrity VM command
Add, delete, manage, and modify vPar/VM storage devices.	`hpvmmodify` (see “Changing VM configurations” (page 154))
Display information about the storage devices for a vPar/VM.	`hpvmstatus` (see “Monitoring guests” (page 239))

After a resource is added or attached to a vPar or VM guest and it is online, the storage resource seen by the guest is owned by the guest administrator. That is, the guest OS may access that storage resource at any time. A deletion, detachment, or modification fails if any guest I/O is active on the resource. Dynamic storage changes on an active vPar or VM must be approved by the guest administrator.

Guest administrator

The vPar or VM Guest Administrator is responsible for the proper maintenance of a guest OS. The VSP administrator must provide the guest administrator access to the virtual console to control the vPar or VM. The guest administrator must know how to maintain the guest OS, install patches and applications, and set up security for the guest users of the guest OS. Additionally, vPar or VM guests storage requires you to:

- Install any specific guest OS patches required by vPars and Integrity VM for proper OS operation on the virtual platform.
- Review and understand any vPar or VM guests storage release notes that are specific to the guest OS.
- Work with the VSP administrator to complete virtual storage changes, including managing attached VSP devices.

The guest administrator uses the virtual console to modify virtual storage. The virtual console is used to change discs of a virtual DVD device type. All modifications are bound by the configurations created by the VSP administrator for the VM.

The virtual console commands are available from the vMP Main Menu, using the `hpvmconsole` command or by pressing **Ctrl+B** if you are already connected. The virtual console commands eject (`ej`) and insert (`in`) allow you to control the DVD device. Both commands provide submenus for displaying devices that are removable. Selecting options through the submenus completes the ejection or insertion process.

If the guest `hpvmconsole pc -cycle` command does not complete and restart the guest, enter **Ctrl+B** to interrupt the command and then press **Enter** to return to the virtual console. Exit the virtual console by entering the `X` command. At the VSP command prompt, enter the following command to start the guest:

```
# hpvmstart -P guestname
```

NOTE: If a guest hangs, attach the guest to the virtual console of the guest using the `hpvmconsole` command, then use **Ctrl+B** to enter the virtual console. Enter the `tc` command to reset the guest. The guest captures a memory dump of the machine state, which can be used later for offline diagnosis. Do not terminate the guest from the VSP or power down a hung guest using the virtual console. Doing so can corrupt the guest file system.

Management function	Integrity VM command
Eject a virtual DVD	`vMP> ej`
Insert a virtual DVD	`vMP> in`

NOTE: When a DVD without a disk in the drive is added to a guest, specify the backing store type `null`. For example,

```
# hpvmmodify -P guest -a dvd:avio_stor::null:/dev/rdisk/disk#
```

Run `ioscan` on the booted guest if the guest is running HP-UX.

If an empty DVD drive is given the backing store type `disk`, the following example shows the result:

```
# hpvmmodify -P testguest -a  dvd:avio_stor::disk:/dev/rdisk/disk31
hpvmmodify: WARNING (testguest): DVD or burner: '/dev/rdisk/disk31' currently has no disk.
This device may not show up or be usable by the guest when booted.
```

If a guest boots when configured with a DVD using the `disk` backing store type when there is no disk in the drive, the guest kit utility command `hpvmdevinfo` (available for HP-UX guests) might return the following results:

```
# hpvmdevinfo
hpvmdevinfo: Error converting (0,0,1): Error 0
Device  Bus,Device,Target  Backing Store  Host Device Name   Virtual Machine Device
Type                       Type                              Name
======  =================  =============  ================   ===========================
disk    [0,0,0]            disk           /dev/rdisk/c2t0d0  /dev/rdisk/c0t0d0
dvd     [0,0,1]            disk           /dev/rdisk/disk31  ??
```

The following phrases in the results indicate the problem of an empty DVD drive:

- The "Error converting (0,0,1): Error 0" message
- The "??" string in the field for the device name of the VM

Output appears for the `dvd`, because it is stored as part of the guest configuration on the VSP. However, because there is no disk in the drive, the drive itself is not virtualized as a device within the guest. Also, note that the DVD drive does not show up in `ioscan` output in the guest.

Guest user

The guest user runs applications on a guest OS. Access is provided and limited by the guest administrator. There are no Integrity VM storage requirements for application users of the guest OS.

There are no Integrity VM storage commands for application users in the guest OS. The guest users use Integrity VM storage on the guest OS the same way as they normally use storage on an HPE Integrity server. Any Integrity VM storage changes must be directed to the guest administrator, guest operator, or the VSP administrator.

Managing storage

This subsection describes ways to use the vPar or VM guests storage commands.

Adding virtual storage devices

A VSP administrator adds or attaches vPar or VM guests storage using the `hpvmstatus` and `hpvmmodify` or `vparmodify` commands. Virtual storage devices can be added or attached while the vPar or VM guest is online. The virtual storage adapter can have up to 128 devices (the number of virtual and attached devices).

To add or attach a virtual storage device to a guest:

1. Based on all the vPar or VM guests storage considerations, choose a storage device to add.
2. Based on the device type, set up and configure the VSP to form a valid resource statement. This includes accounting VSP resources to avoid future storage conflicts.
3. Use the valid resource statement with the `hpvmmodify` command to add or attach the vPar or VM guests storage device.

For more information about dynamic addition of IO devices to vPars and VM guests, see "Dynamic I/O for vPars and Integrity VM guests" (page 269).

The resource statement for adding a vPar or VM guests storage device does not require virtual hardware addressing. If the PCI bus, slot, and target numbers are not specified, vPars and Integrity VM automatically chooses the first position available for the device. For example:

```
# hpvmmodify -P myvmm -a disk:avio_stor::disk:/dev/rdisk/disk7
# hpvmstatus -P myvmm
..
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk7
```

NOTE: If the PCI bus, slot, and target numbers are not specified, then, vPars and Integrity VM automatically adds up to 15 targets behind a single HPVM AVIO Storage HBA and then moves on to a new virtual HBA for subsequent additions.

To add more than 15 devices behind a given AVIO virtual HBA, the target number must be explicitly specified.

To add an AVIO storage device at a specific PCI bus, slot, and target, specify the following:

```
host# hpvmmodify -P guest1 -a disk:avio_stor:0,5,0:disk:/dev/rdisk/disk11
```

NOTE: You can achieve higher guest performance for HP-UX 11i v3 guests older than the March 2011 release by configuring as many AVIO storage adapters as the number of virtual CPUs in the guest. The `pcibus`, `pcislot`, and `aviotgt` portions must be explicitly specified for each device. For example, a resource statement for a 4–vCPU guest takes the following form:

```
-a disk:avio_stor:1,0,0:disk:/dev/rdisk/disk1
-a disk:avio_stor:1,1,0:disk:/dev/rdisk/disk2
-a disk:avio_stor:1,2,0:disk:/dev/rdisk/disk3
-a disk:avio_stor:1,4,0:disk:/dev/rdisk/disk4
```

These are not the requirements for guests that are at the March 2011 or later releases.

NOTE: A DMP device can be added online to a vPar or a VM guest post v6.3.5. If you attempt the operation while the vPar or VM guest is online, the addition fails and the new device addition does not get saved to the guest configuration to be applied when the guest is next restarted. You have to repeat the device addition after the vPar or VM guest is shut down.

Deleting storage devices

A VSP administrator deletes or detaches vPar or VM guests storage using the `hpvmstatus` and `hpvmmodify` or `vparmodify` commands. vPar or VM guests storage devices can be deleted or detached dynamically. The vPar or VM guests storage adapter is automatically removed when the last vPar or VM guests storage device connected to the adapter is removed.

NOTE: AVIO virtual HBAs and devices configured under legacy AVIO virtual devices cannot be deleted (using the `hpvmmodify` or `vparmodify` commands) if the vPar or VM guest is at EFI.

To delete or detach a virtual storage device from a vPar or VM guest:

1. Use the `hpvmstatus` command to locate the resource to verify whether the vPar or VM is powered on. If the vPar or VM is on, consult with the guest administrator to obtain permission to remove the resource before proceeding.
2. Use the `hpvmmodify` command to delete or detach the resource.

The resource statement for deleting a vPar or VM guest storage device does not require virtual hardware addressing. For example:

```
# hpvmstatus -P myvmm
...
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk7
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
disk avio_stor 0 5 0 0 0 disk /dev/rdisk/disk11
# hpvmmodify -P myvmm -d disk:avio_stor::disk:/dev/rdisk/disk7
# hpvmstatus -P myvmm
...
[Storage Interface Details]

disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
```

To delete an AVIO storage device, specify the following:

```
host# hpvmmodify -P guest1 -d disk:avio_stor:0,5,0:disk:/dev/rdisk/disk11
```

To delete an NPIV HBA, specify the following:

```
host# hpvmmodify -P guest1 -d hba:avio_stor:0,2,0x50014C27FFFFF000,0x50014C2FFFFFF000:npiv:/dev/fcd0
```

Modifying storage devices

The VSP administrator or the guest administrator can modify a vPar or VM guest storage device. The VSP administrator can use the `hpvmstatus` and `hpvmmodify` commands to change the virtual media of virtual devices. The guest administrator uses the virtual console to change the virtual media of virtual DVDs. All attached devices are modified using physical VSP access.

When the VSP administrator uses the `hpvmstatus` and `hpvmmodify` commands to modify the virtual media of a virtual device, for the guest OS, this operation is a whole-disk replacement or a DVD removable media event, depending on the device type.

To modify the virtual media of a virtual device:

1. Use the `hpvmstatus` command to locate the virtual device resource to modify and to verify whether the VM is powered on. If the vPar or VM guest is on, consult with the guest administrator before proceeding to replace the virtual media.
2. Based on the vPar or VM guest storage considerations, choose a new virtual media type to add.
3. Based on the virtual media type, set up and configure the VSP to form a valid VSP storage specification. Take into account the other demands on VSP resources to avoid vPar or VM guest storage conflicts.
4. Use the VSP storage specification with the `hpvmmodify` command to modify the virtual device resource.
5. Verify that the old VSP resource is no longer in use by a vPar or VM guest.
6. When run on an active vPar or VM guest and with a storage device managed by `avio_stor` HBA, the vPar and VM guest must run the `gvsdmgr` command before using the modified backing store. For information about the `gvsdmgr` utility, see the HP-UX *gvsdmgr*(1M).

The resource statement for modifying a virtual device requires virtual hardware addressing (see “Storage specification” (page 74)). For example:

```
# hpvmstatus -P myvmm
...
```

```
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk7
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
# hpvmmodify -P myvmm -m disk:avio_stor:0,1,1:disk:/dev/rdisk/disk2
# hpvmstatus -P myvmm
...
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk2
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
```

To complete a DVD ejection and insertion, follow the virtual console menu. However, new media selections might require the help of the VSP administrator. Changes through the virtual console are not saved across guest OS reboots.

If the VSP administrator sets up a Virtual DVD for the vPar and VM guest, the virtual console eject and insert command unlock and lock the physical VSP CD or DVD drive. The `eject` command changes the Virtual DVD into a Virtual NullDVD in the vPar and VM guest, unlocking the VSP CD or DVD drive in the process. The physical media in the VSP CD or DVD drive can then be changed by the VSP administrator or the guest administrator if access is permitted. After the media is changed, the `insert` command can be used to change the Virtual NullDVD back into a Virtual DVD, locking the VSP CD or DVD drive and making the newly loaded media accessible by the vPar and VM guest. For example:

```
# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual DVD
               type: CD-ROM
               size: 665600 Kbytes
   bytes per sector: 2048
vMP> ej

                Ejectable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev    Pstore  Path
-------------------------------------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd     disk    /dev/rdisk/disk7

Enter menu item number or [Q] to Quit: 1
Confirm eject action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual NullDVD
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0

vMP>
```

After inserting a new disk on the VSP CD or DVD drive, enter the following:

```
vMP> in
Insertable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev
--------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd
```

```
Enter menu item number or [Q] to Quit: 1
Insertable File Backing Stores
Num     File
---------------------
[1]     /dev/rdisk/disk7

Enter menu item number or [Q] to Quit: 1
Confirm insertion action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual DVD
               type: CD-ROM
               size: 4300800 Kbytes
   bytes per sector: 2048
```

NOTE: Guest operating systems, applications, or configuration files sensitive to device names or hardware paths must be repaired after the move. Because HP-UX 11i v3 supports the agile device naming model, 11i v3 guest applications using agile device names are not affected as long as they are configured with disk backing stores.

If the VSP administrator sets up a Virtual FileDVD for the vPar and VM guest, the virtual console options to eject and insert are used to select among the ISO files provided in the file directory for the Virtual FileDVD. The `eject` command changes the Virtual FileDVD into a Virtual NullDVD device. The VSP administrator can add ISO files to and remove them from the file system directory for the Virtual FileDVD. After the ISO file directory is updated, use the `insert` command to view all the newly available ISO files in the directory and choose one to be used for a new Virtual FileDVD. It is not necessary to change the file directory between each eject and insert operation. The guest administrator can change the ISO files provided in the file directory without any VSP administrator interaction. For example:

```
# diskinfo /dev/rdisk/disk0
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual FileDVD
               type: CD-ROM
               size: 665600 Kbytes
   bytes per sector: 2048
vMP>ej
                Ejectable Guest Devices
Num  Hw-path     (Bus,Slot,Tgt) Gdev   Pstore Path
--------------------------------------------------------------------------------
[1]  0/0/1/0.7.0 (0,1,7)        dvd    file   /var/opt/hpvm/ISO-images/hpux/IOTdisc

Enter menu item number or [Q] to Quit: 1

Confirm eject action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
vm # diskinfo /dev/rdisk/disk0
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual NullDVD
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0
```

```
vMP> in
                Insertable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev
--------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd

Enter menu item number or [Q] to Quit: 1
Insertable File Backing Stores
Num     File
---------------------
[1]     0505-FOE.iso
[2]     0512-FOE.iso
[3]     0603-FOE-D1.iso
[4]     0603-FOE-D2.iso
[5]     IOTdisc

Enter menu item number or [Q] to Quit: 1
Confirm insertion action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk0
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual FileDVD
               type: CD-ROM
               size: 3686144 Kbytes
   bytes per sector: 2048
```

For attached devices, modifications are made physically on the device. The guest OS supplies commands for loading and unloading tapes using media changers. But loading new media into the media changer, changing tapes in standalone drives, and changing discs with CD or DVD burners are accomplished manually. This process requires cooperation between the VSP administrator and the guest administrator.

Troubleshooting Storage related problems

For more information about troubleshooting storage related problems, see “Storage” (page 292).

7 NPIV with vPars and Integrity VM

NPIV allows you to create multiple virtual Fibre Channel ports (vFCs) over one physical Fibre Channel port (pFC) on a VSP. To identify a virtual port, you must create the virtual port with a unique World Wide Name (WWN), just like the unique embedded WWN by which a physical port is identified.

Using the NPIV feature, you can allocate the vFC instances created over a physical port as resources to vPar and VM guests. The resource that is added to the vPar or VM is a virtual Host Bus Adapter or virtual HBA (vHBA). The vPar or VM guest then automatically discovers targets and LUNs behind the vHBA using the same mechanism used on a standalone system to discover targets and LUNs behind a physical HBA.

With the introduction of NPIV, vPars and VM guests can now support two kinds of devices:

- Legacy AVIO (shared I/O, attached I/O)
- LUNs visible with the vHBA (NPIV HBAs)

NPIV devices can co-exist with legacy AVIO devices in the same vPar or VM guest. Unlike legacy AVIO storage, the NPIV LUNs do not need to be visible to the VSP and therefore, the LUNs that the vPar or VM guest will discover behind the vHBA can be managed and provisioned the same way as on a standalone system.

NOTE: NPIV is supported only on HP-UX 11i v3 guests.

The same LUN cannot be presented to a vPar or VM guest as both an NPIV device and legacy AVIO device.

Benefits of NPIV

Following are some of the benefits of NPIV:

- Provides storage isolation between vPar or VM guests and the VSP, and among vPar or VM guests.
- Provides security and I/O traffic isolation by providing LUN masking and zoning capabilities similar to regular FC LUNs.
- Allows running of applications that require un-virtualized device access on the vPar.
- Allows monitoring the server and storage environment using charge back applications.
- Streamlines vPar and VM guest migrations.

For more information about NPIV and its benefits, see *HP-UX vPars 6.0 and Integrity VM 4.3 N_Port ID Virtualization (NPIV)* at **http://www.hpe.com/info/hpux-hpvm-docs**.

Dependencies and prerequisites

The NPIV functionality requires a hardware I/O stack, which explicitly supports NPIV from the HBAs through the interconnect modules and SAN fabric. NPIV is supported with Emulex, Qlogic FC cards, and Emulex CNA cards. For more information about supported HBAs, see *HP-UX vPars and Integrity VM v6.4 Release Notes* at **http://www.hpe.com/info/hpux-hpvm-docs**.

NOTE:

- NPIV is supported only with fabric topologies. It is not supported with arbitrated loop topologies where a FC host port is directly connected to the end device target port or via FibreChannel Hubs. You can use the `fcmsutil` command to determine whether the FC port is configured with fabric topology:

  ```
  # /opt/fcms/bin/fcmsutil /dev/fcd0 | grep "Topology"
  Topology = PTTOPT_FABRIC
  ```

- NPIV feature must be enabled on the FibreChannel Switch. By default, some FibreChannel Switches disable the NPIV feature.

NPIV — supported limits

Table 14 (page 100) lists the supported limits associated with NPIV in vPars and Integrity VM v6.4 on 11i v3 vPars and VM guests.

Table 14 NPIV supported limits in vPars and Integrity VM v6.4

Limit description	Supported limit
NPIV HBAs per vPar and VM guest	16
Number of NPIV HBAs per physical HBA	32
Number of paths supported per NPIV device	16
Number of LUNs per NPIV HBA	2048
Number of NPIV devices per vPar and VM guest	2048

NOTE: In configurations where multiple NPIV HBAs created on a single physical HBA are used by different vPars and VM guests, all the I/O from these vPars and VM guests share a single physical HBA, which can lead to performance bottlenecks in high I/O scenarios.

For a more balanced performance, Hewlett Packard Enterprise recommends that you spread NPIV HBAs for vPars and VM guests across multiple physical adapters.

Configuring an NPIV HBA (vHBA)

The overall configuration process for NPIV HBAs is the same as for AVIO. Starting v6.1, a new storage type called **`npiv`** was introduced for configuring NPIV HBAs.

The following sections describe how to determine whether an existing FC card on the VSP supports NPIV, how an NPIV HBA resource is specified, and how you can present storage devices to an NPIV HBA both, before and after a guest starts up.

Verifying whether VSP can support NPIV

Before creating an NPIV HBA, check the physical HBAs on the system to verify that they support NPIV. Run the `fcmsutil` command on a VSP Fibre Channel HBA:

```
/opt/fcms/bin/fcmsutil /dev/fcXXX
```

where /dev/fcXXX is the DSF (device special file) associated with the Fibre Channel port. It can be obtained from the `ioscan -kfnC fc` command:

```
# ioscan -kfnC fc

Class I   H/W Path     Driver S/W State H/W Type   Description
===================================================================
fc    0  0/2/0/0/0/0  fcd    CLAIMED    INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC Port
1)/dev/fcd0
```

```
fc    1  0/2/0/0/0/1  fcd    CLAIMED   INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC Port
2)/dev/fcd1
```

The following sample shows you whether NPIV is supported on the VSP:

```
# /opt/fcms/bin/fcmsutil /dev/fcd0
Vendor ID is = 0x1077
                           Device ID is = 0x2532
            PCI Sub-system Vendor ID is = 0x103C
                   PCI Sub-system ID is = 0x3263
                               PCI Mode = PCI Express x8
                       ISP Code version = 5.4.0
                       ISP Chip version = 2
                               Topology = PTTOPT_FABRIC
                             Link Speed = 4Gb
                     Local N_Port_id is = 0x010800
                  Previous N_Port_id is = None
            N_Port Node World Wide Name = 0x5001438002344785
            N_Port Port World Wide Name = 0x5001438002344784
            Switch Port World Wide Name = 0x200800051e0351f4
            Switch Node World Wide Name = 0x100000051e0351f4
              N_Port Symbolic Port Name = porti3_fcd0
              N_Port Symbolic Node Name = porti3_HP-UX_B.11.31
                           Driver state = ONLINE
                       Hardware Path is = 0/2/0/0/0/0
                     Maximum Frame Size = 2048
         Driver-Firmware Dump Available = NO
         Driver-Firmware Dump Timestamp = N/A
                                   TYPE = PFC
                         NPIV Supported = YES
                         Driver Version = @(#) fcd B.11.31.1103 Aug  2 2011
```

If NPIV is supported, running the command again with the new `npiv_info` option provides information about all the running virtual HBAs currently associated with this physical HBA:

```
# /opt/fcms/bin/fcmsutil /dev/fcd0 npiv_info
PFC Hardware Path               = 0/0/0/5/0/0/0
PFC DSF                         = /dev/fcd0
PFC Class Instance              = 0
PFC Driver state                = ONLINE
PFC Port WWN                    = 0x5001438001459910
PFC Node WWN                    = 0x5001438001459911
PFC Switch Port WWN             = 0x201400051ef06bd3
PFC Switch Node WWN             = 0x100000051ef06bd3

FlexFC Virtual Fibre Channel (VFC)
----------------------------------
Maximum Supported FlexFC VFC    = 16
Number Active FlexFC VFC        = 0

HPVM Virtual Fibre Channel (VFC)
----------------------------------
Maximum Supported HPVM VFC      = 48
Number Active HPVM VFC          = 1

The following provides the list of VFC(s) associated with this PFC:

Type                            = HPVM VFC
VFC Index                       = 17
VFC Guest ID                    = 0x4
VFC Port WWN                    = 0x50014c2000000007
VFC Node WWN                    = 0x50014c2800000023
VFC Driver state                = ONLINE
VFC DSF                         = /dev/fcd6
VFC Class Instance              = 6
```

Specifying an NPIV HBA resource

An NPIV resource is specified using the following format:

`devicetype:adaptertype:bus,device,vWWP,vWWN:storage:device`

where:

devicetype
: The virtual device type as seen in the vPar. For NPIV, this will be **hba**.

adaptertype
: The adapter type as seen in the vPar. For NPIV, the adaptor type is **avio_stor**.

bus
: The PCI bus number for the virtual device; can range 0 to 7.

device
: The PCI slot number for the virtual device; can range 0 to 7.

vWWP
: A valid (64 bit), unique (virtual) Port WWN that is assigned to the NPIV HBA. This is analogous to the unique Port WWN that is associated with physical HBAs.

vWWN
: A valid (64 bit), unique (virtual) Node WWN that is assigned to the NPIV HBA. This is analogous to the unique Node WWN that is associated with physical HBAs.

storage
: The physical storage type in the host. For NPIV, this is **npiv**.

device
: The physical device in the host corresponding to the virtual device. For NPIV, this corresponds to the device special file for the physical port on which the virtual NPIV instance is created.

NOTE: Certain PCI slots are used by vPars and VM guests for special devices. You can use the `hpvmstatus -P <guest_name> -V` command to get a list of the reserved slots.

Finding and using WWNs

Using NPIV HBAs generate virtual WWNs. Administrators are responsible for tracking WWNs and guaranteeing their uniqueness across the Storage Area Network (SAN). You can allocate and manage unique WWNs for NPIV HBAs using the HP-UX GUID Manager, which is a client-server based product. Using this application ensures that you do not perform this task manually. GUID Manager is integrated with vPars and Integrity VMs to support NPIV, and it is also integrated with HPE Integrity Virtual Server Manager to support managing the WWN database.

IMPORTANT: Hewlett Packard Enterprise recommends using the HP-UX GUID Manager to allocate and maintain vWWPs and vWWNs. For more information about HP-UX GUID Manager, see the *HP-UX GUID Manager Administrator Guide* at **http://www.hpe.com/info/hpux-hpvm-docs**.

Creating and managing NPIV HBA

Adding NPIV HBA resources

An NPIV HBA resource can be specified while creating the vPar or VM guest, or after the vPar or VM guest is created. And, NPIV HBA can also be added to a vPar or VM guest while it is online. For more information about the resource string format for an NPIV resource, see “Specifying an NPIV HBA resource” (page 102).

IMPORTANT: Before creating an NPIV HBA, ensure that the physical HBAs on the system support NPIV.

Example 4 Create an NPIV HBA using the GUID server for WWNs

Create a vPar named vPar1 with 4 virtual CPUs and an NPIV HBA created on `/dev/fcd0` using the GUID server to assign port and node WWNs. You can also use the `vparstatus -a` command to display NPIV capable fiber channel devices.

```
vparcreate -P vPar1 -c 4 -a hba:avio_stor::npiv:/dev/fcd0
```

```
vparcreate -P vpar1 -c 4 -a hba:avio_stor:,,,:npiv:/dev/fcd0
```

NOTE: It is optional to use the commas. The comma is an alternate way to get WWNs from the GUID server. If you use the commas, ensure that you specify 3 commas.

Example 5 Create an NPIV HBA manually specifying WWNs

Add an NPIV HBA created on `/dev/fcd1` using a virtual port WWN of 0x50060b00006499b9 and virtual node WWN of 0x50060b00006499ba to the vPar named vPar1. Obtain the port and node WWNs from your storage administrator or other source.

```
vparmodify -P vPar1 -a hba:avio_stor:,,0x50060b00006499b9,
0x50060b00006499ba:npiv:/dev/fcd1
```

In the resource string, you can skip the bus and slot numbers for an NPIV HBA. VSP picks the next available bus and slot number for the NPIV HBA. However, you cannot skip the virtual node WWN and port WWN if the VSP is not configured to obtain WWNs from a GUID server.

NOTE: The `vparmodify` or `hpvmmodify` command cannot be used to change any attribute of the NPIV HBA after it is created.

Viewing NPIV resources

The `vparstatus` or `hpvmstatus` command output includes the NPIV HBA in the I/O details for vPars and VM guests that have NPIV HBAs configured.

Example 6 Sample to determine NPIV HBA details for a vPar named vPar1

```
vparstatus -P Vpar1 -d
[Virtual Partition Devices]

[Storage Interface Details]
disk:avio_stor:0,0,0:avio_stor:/dev/rdisk/disk31
disk:avio_stor:0,0,1:lv:/dev/vg_on_host/rlvol3
hba:avio_stor:0,4,0x50060b00006499b9,0x50060b00006499ba:npiv:/dev/fcd1
hba:avio_stor:1,3,0x50060b00006499a0,0x50060b00006499a8:npiv:/dev/fcd0

[Network Interface Details]
network:avio_lan:0,1,0xF2AF8F8647BF:vswitch:vswitch1:portid:1
network:avio_lan:0,5,0x569FC1F96205:vswitch:vswitch1:portid:3

[Misc Interface Details]
serial:com1::tty:console
```

In this example, the port WWNs are:

- 0x50060b00006499b9
- 0x50060b00006499a0

These WWNs must be used for LUN masking or fabric zoning.

Deleting configured NPIV HBAs

The `vparmodify` or `hpvmmodify` command is used to delete an NPIV HBA from a vPar or VM guest.

Example 7 Deleting an NPIV resource

```
vparmodify -P vPar1 -d hba:avio_stor:,,0x50060b00006499b9,
0x50060b00006499ba:npiv:/dev/fcd1
```

For the relevant vPar or VM guest, you can use this syntax by copying it from the I/O details of the `status` command output and pasting it where required.

With NPIV HBAs, new LUNs presented or unpresented to the virtual HBA are automatically detected by the guest.

Configuring storage for a vPar or VM guest with NPIV HBAs

You can assign storage for a vPar or VM guest with an NPIV HBA either before or after it starts up. In both cases, the guest boots if a non-NPIV boot device is configured. If it does not have a boot device, the guest boot halts at EFI.

To configure storage for a guest with an NPIV HBA:

1. Start the guest.

   After the guest starts, the virtual port instance corresponding to the NPIV HBA assigned to the guest logs into the FC fabric to which the physical HBA is connected.
2. Obtain the port WWN assigned to the NPIV HBA using the `vparstatus` or `hpvmstatus` command on VSP.
3. Note the port WWN number, and work with the storage administrator to get the required storage provisioned.

   The storage administrator must use the storage management utility corresponding to the storage device from which the administrator plans to provision storage, and then create LUNs of the required capacity.
4. Present the LUNs to the port WWN corresponding to the NPIV HBA.

Installing the guest image on NPIV disks

After the LUNs are presented to the NPIV HBA, the vPar and VM guest image can be installed on an NPIV device. After the NPIV device is enumerated and selected at EFI shell, installing vPar or VM guest images on it is same as the HP-UX installation process.

Identifying NPIV HBAs and devices in a guest

To identify an NPIV HBA from the set of HBAs in the guest, use the `ioscan` command.

Example 8 Identifying NPIV HBAs and devices in a vPar

```
# ioscan -kfNd gvsd
Class     I  H/W Path  Driver S/W State   H/W Type     Description

ext_bus   0  0/0/0/0   gvsd   CLAIMED     INTERFACE    HPVM AVIO Stor Adapter
ext_bus   1  0/0/4/0   gvsd   CLAIMED     INTERFACE    HPVM NPIV Stor Adapter
ext_bus   4  0/1/3/0   gvsd   CLAIMED     INTERFACE    HPVM NPIV Stor Adapter
```

NOTE: The `ioscan` output listing the NPIV devices in the guest is the same as a similar listing of SAN LUNs in a native host.

Legacy DSFs are not supported for NPIV devices, hence, the `ioscan` command displays NPIV devices only if the `-N` option is used.

Example 9 Identifying NPIV HBAs and devices in a guest by specifying hardware path

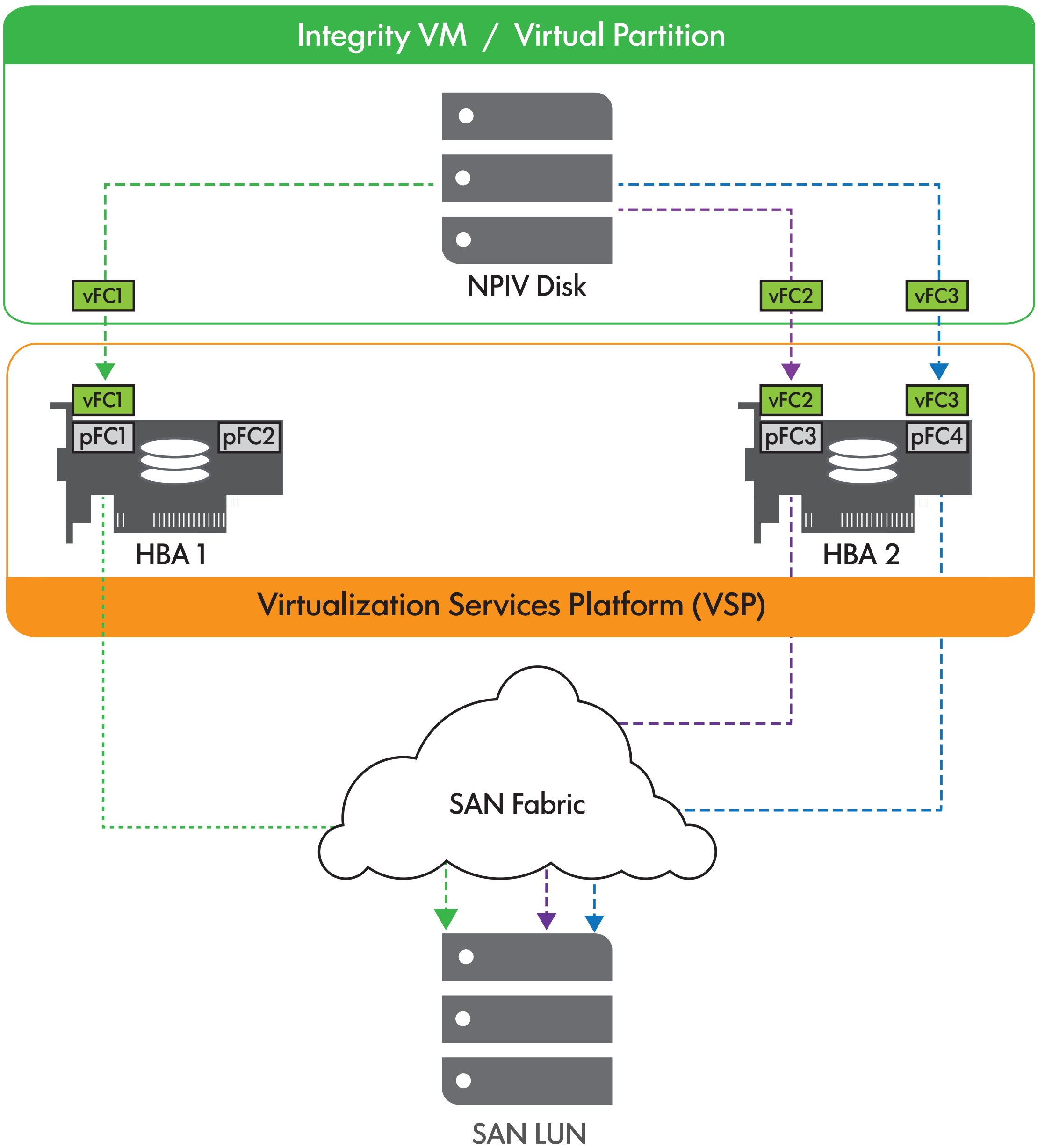
```
# ioscan -kfNH 0/0/4/0
Class     I  H/W Path  Driver S/W State   H/W Type     Description
===================================================================
ext_bus   1  0/0/4/0   gvsd    CLAIMED     INTERFACE    VPAR NPIV Stor Adapter
tgtpath   3  0/0/4/0.0x207000c0ffda0287                    estp    CLAIMED     TGT_PATH     Virtual Storage
HBA target served by gvsd driver, target port id 0x105ef
lunpath   5  0/0/4/0.0x207000c0ffda0287.0x0                eslpt   CLAIMED     LUN_PATH     LUN path for ctl1
lunpath   8  0/0/4/0.0x207000c0ffda0287.0x4001000000000000  eslpt   CLAIMED    LUN_PATH    LUN path for disk7
lunpath   9  0/0/4/0.0x207000c0ffda0287.0x401d000000000000  eslpt   CLAIMED    LUN_PATH    LUN path for disk8
tgtpath   4  0/0/4/0.0x247000c0ffda0287                    estp    CLAIMED     TGT_PATH     Virtual Storage
HBA target served by gvsd driver, target port id 0x104ef
lunpath   6  0/0/4/0.0x247000c0ffda0287.0x0                eslpt   CLAIMED     LUN_PATH     LUN path for ctl2
lunpath  11  0/0/4/0.0x247000c0ffda0287.0x4001000000000000  eslpt   CLAIMED    LUN_PATH    LUN path for disk7
lunpath  12  0/0/4/0.0x247000c0ffda0287.0x401d000000000000  eslpt   CLAIMED    LUN_PATH    LUN path for disk8
```

Configuring multiple paths for NPIV devices

For NPIV devices, multi-pathing products run on the vPar or VM guest and not on the VSP. Multiple paths to an NPIV device can be configured by presenting it to multiple NPIV HBAs created on different FC ports on the VSP.

Figure 12 (page 106) shows a possible NPIV configuration that provides multiple paths to the NPIV device in the guest. In this example, the NPIV disk has 3 paths, one through HBA1 and one each through each of the two ports on HBA2.

Figure 12 Multi-pathing with NPIV devices



NOTE: Having multiple paths to an NPIV device through the same physical HBA port on the VSP does not fetch the benefits of multi-pathing because all paths will be using the same physical port for IO traffic and thereby not provide any redundancy.

All aspects of native multi-pathing for regular FC device on a physical host is applicable to an NPIV device seen in the guest. For more information about native multi-pathing on HP-UX, see *HP-UX 11i v3 Native Multi-Pathing for Mass Storage* at **http://www.hpe.com/info/hpux-hpvm-docs**.

NPIV pools

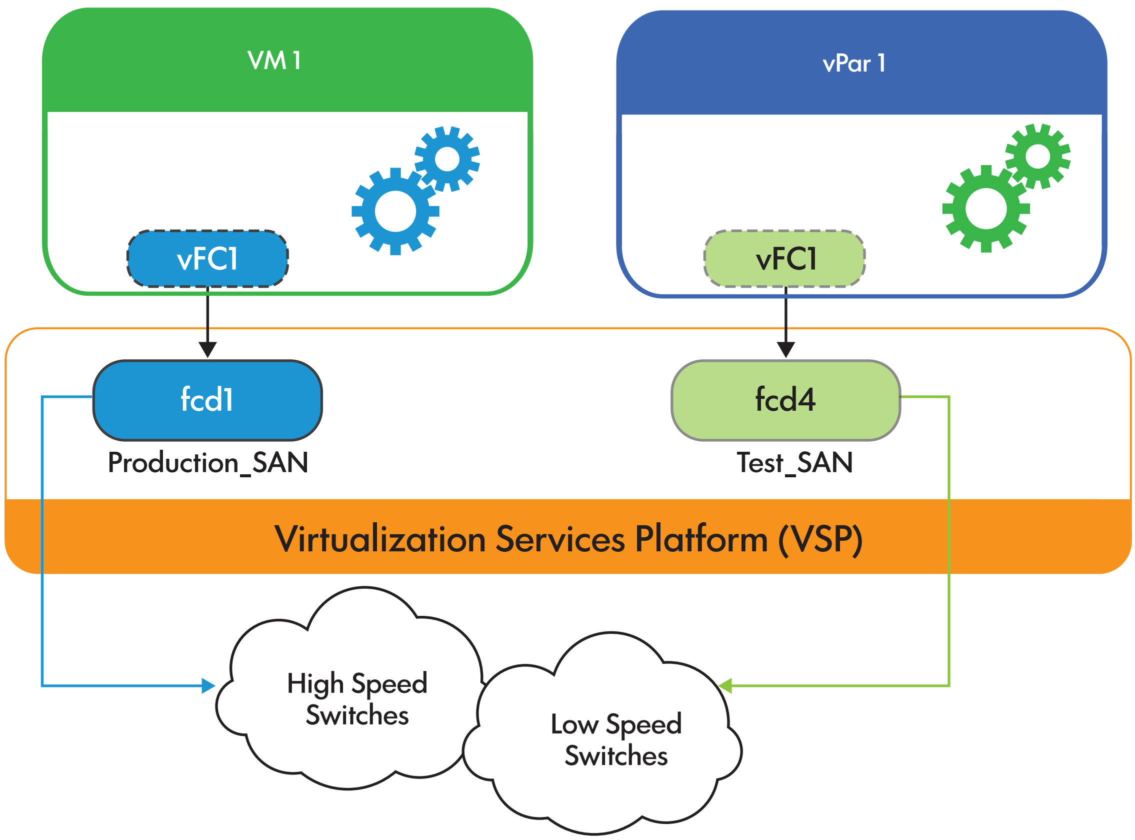
With the increasing consolidation of workloads in virtual environments, it becomes necessary to ensure that your critical workloads get to operate on the best of resources—both on the original VSP host and on any target VSP they may migrate to.

With product versions prior to v6.3.5, it was not possible to categorize NPIV capable physical resources. Starting from v6.3.5 onwards, the product allows NPIV capable resources to be labeled on the source and target VSPs. This labelling can be based on the speed of FC ports or the load on the switch to which the physical FC ports are connected. After this is done, the `hpvmmigrate` command will use the label information as a hint while picking FC ports on the target VSP for placement of NPIV HBAs of the migrating vPar or VM guest.

The NPIV capable physical FC ports with similar characteristics can be grouped together into specific pools based on either zones or workloads running on guests and so on. Classifying an NPIV HBA into a pool helps to maintain the SAN isolation during guest migrations.

Figure 13 (page 107) describes the classification of an NPIV HBA into a pool and about the maintenance of SAN isolation.

Figure 13 NPIV pool—SAN isolation with NPIV HBA



Creating and managing NPIV pools

To label a NPIV capable resource it has to be added to the NPIV pool with the label specified. All the NPIV capable resources that are not labeled will be considered to be part of the default pool – DEFAULT_POOL.

The `hpvmhwmgmt` command is used to create and manage NPIV resource pool.

- To list all NPIV capable FC ports in the NPIV pool

  ```
  #hpvmhwmgmt -p npiv -l
  ```

- To add a NPIV capable FC port to a NPIV pool

  ```
  #hpvmhwmgmt -p npiv -a device dsf -L label
  ```

- To modify the NPIV pool of a NPIV capable FC port

  ```
  #hpvmhwmgmt -p npiv -m device dsf -L label
  ```

- To delete an NPIV capable FC port from a NPIV pool

  ```
  #hpvmhwmgmt -p npiv -d device dsf -L label
  ```

Example 10 Label an NPIV capable FC port

To associate an NPIV capable FC port `/dev/fcd5` to a label NPIV_POOL_PRODUCTION, the following command can be used:

```
#hpvmhwmgmt -p npiv -a /dev/fcd5 -L NPIV_POOL_PRODUCTION
```

Example 11 Modify the label of an NPIV capable FC port

To modify the label associated with the NPIV capable FC port in the NPIV pool `/dev/fcd5` from NPIV_POOL_PRODUCTION to NPIV_POOL_TEST, the following command can be used:

```
#hpvmhwmgmt -p npiv -m /dev/fcd5 -L NPIV_POOL_TEST
```

NOTE:

- Before addition or modification of an NPIV HBA in a NPIV resource pool, ensure that
    - Physical FC ports (pFC) on the system supports NPIV.
    - pFCs are not OFFLINE or DISABLED.
- The label strings “NONE” and “NULL” are not valid.
- The label string “DEFAULT_POOL” is reserved.

Example 12 Delete the label associated with a FC port

To delete the label NPIV_POOL_TEST associated with FC port `/dev/fcd5` from the NPIV resource pool NPIV_POOL_TEST, the following command can be used:

```
#hpvmhwmgmt -p npiv -d /dev/fcd5 -L NPIV_POOL_TEST
```

Example 13 List all the labels associated with the NPIV capable FC ports in the NPIV pool

To list the labels associated with NPIV capable FC ports in the NPIV resource pool, the following command can be used:

```
#hpvmhwmgmt -p npiv -l

Device         Label
– – – – –      – – – – – –
/dev/fcd2      NPIV_POOL_TEST
/dev/fcd5      NPIV_POOL_PRODUCTION
```

Bandwidth management for NPIV HBAs

The HP-UX vPars and Integrity VM product enables the creation of multiple virtual Fibre Channel ports (vFCs) over one physical Fibre Channel port (pFC). An HBA resource is shared across all vPars and VMs that have active vFC instances created on it. It includes the capacity and bandwidth of the physical link, which is shared between the vFCs and pFC instances. With increase in workload consolidation, more diverse loads get consolidated onto a single platform. In such instances, higher priority workload running inside a VM guest or vPar may experience a reduction in bandwidth availability when a low priority VM guest or vPar uses the available bandwidth on shared pFCs. In such situations, the server administrator may want to provision higher bandwidth for some of the Integrity VMs or vPars as compared to the rest.

This can be done by dedicating a 16 Gb FC card against an 8 Gb or 4 Gb FC card for VMs running critical workloads, or by provisioning FC connectivity to the end storage device through a faster switch. However, it does not result in the optimal usage of the available resources on the VSP. Even these workloads may need to be migrated from one VSP to another. In this case, it is desirable to place the NPIV vFCs on the right set of target pFCs to match the original bandwidth criteria that were set up on the source.

Up until v6.4 of the vPars and Integrity VM product, there was no way to associate any bandwidth requirements with a vFC. v6.4 of vPars and Integrity VM product along with the patches (listed in software dependencies) addresses this problem. It introduces the capability to assign specific bandwidth entitlements, and cap the bandwidth usage of vFCs. Bandwidth entitlement measures and controls the usage of a communication link. Bandwidth capping is about limiting an entity's (vFC's) bandwidth usage, when there is contention for bandwidth. It ensures minimum guaranteed bandwidth when there is heavy load on the pFC. The bandwidth entitlement of a vFC can be crossed when there is sufficient bandwidth available on the shared pFC. Also, to retain bandwidth requirements across migrations, it is ensured that the vFCs of vPar or VM guest are placed on the target VSP such that its bandwidth requirements are met on the target VSP. NPIV capable physical resources on the source and target VSPs can be grouped into specific pools or groups. It allows the administrator to consolidate and migrate Integrity VMs and vPars running workloads with varied priority or bandwidth requirement on a set of VSPs sharing FC fabric.

With the bandwidth management feature, the specified share of the physical FC port's bandwidth is guaranteed to be available for an NPIV vFC under all conditions, especially under heavy usage of a pFCs' bandwidth by multiple vFCs.

NOTE: Automatic bandwidth negotiations may not be automatically relayed to HP-UX vPars and Integrity VM.

Table 15 (page 109) lists the mapping of bandwidth granularities to its corresponding percentages for a 16Gb HBA port.

Table 15 Bandwidth to entitlement mapping on a 16Gb HBA port

Bandwidth	Percentage
2 Gb	12.5%
4 Gb	25%
6 Gb	37.5%
8 Gb	50%
10 Gb	62.5%
12 Gb	75%
14 Gb	87.5%

Dependencies and prerequisites

Bandwidth entitlement of NPIV HBAs is supported only on 16 Gb Qlogic FC cards. For more detail on list of supported HBAs, see *HP-UX vPars and Integrity VM v6.4 Release Notes* at **http://www.hpe.com/info/hpux-hpvm-docs/**.

Software dependencies

In addition to HP-UX vPars and Integrity VM v6.4, the following or superseding patches are required to enable this functionality:

On VSP

- PHSS_44424
- PHSS_44425
- PHSS_44426

- PHSS_44428
- PHSS_44429
- PHSS_44430

On Guest

- PHSS_44425
- PHSS_44427
- PHSS_44429
- PHSS_44431

NOTE: For more information on the Fibre Channel(FC) driver and the firmware supported for bandwidth entitlement, see *HP-UX vPars and Integrity VM v6.4 Release Notes* available at **http://www.hpe.com/info/hpux-vpars-docs**.

Supported limits

Table 16 (page 110) lists the supported limits associated with NPIV HBAs with bandwidth entitlement in HP-UX vPars and Integrity VM v6.4 on 11i v3 vPars and VM guests.

Table 16 Supported limits for number of bandwidth entitled HBAs per pFC

Limit description	Supported limit
Number of NPIV HBAs with 12.5% entitlement	7
Number of NPIV HBAs with 25% entitlement	3
Number of NPIV HBAs with 37.5% entitlement	2
Number of NPIV HBAs with 50% entitlement	1
Number of NPIV HBAs with 62.5% entitlement	1
Number of NPIV HBAs with 75% entitlement	1
Number of NPIV HBAs with 87.5% entitlement	1

NOTE: In configurations, where multiple NPIV HBAs with different bandwidth entitlement are created on the same physical HBA, the supported limits count varies based on the available bandwidth on the physical HBA. Use the `/opt/hpvm/bin/hpvmstatus -n -V` command to find out the number of NPIV HBAs that can be created with a particular bandwidth entitlement.

Configuring an NPIV HBA with bandwidth entitlement

The overall configuration process for NPIV HBAs with bandwidth entitlement is same as an NPIV HBA. Starting HP-UX vPars and Integrity VM v6.4 with PK2 or superseding patches supports (listed in “Software dependencies ” (page 109)), a new field `percent` is introduced for configuring NPIV HBAs with bandwidth entitlement.

The following sections describe how to determine whether an existing FC card on the VSP supports bandwidth entitlement or not, and how to specify the NPIV HBA resource with bandwidth entitlement.

Verifying whether VSP can support NPIV HBA

To verify whether the VSP can support NPIV HBA with bandwidth entitlement:

- Check the physical HBAs on the system to verify they support NPIV and bandwidth entitlement before creating an NPIV HBA with bandwidth entitlement.
- Run the `fcmsutil` command on a VSP Fibre Channel HBA:

  `/opt/fcms/bin/fcmsutil /dev/fcXXX npiv_info`

  where /dev/fcXXX is the DSF (device special file) associated with the Fibre Channel port. It can be obtained from the `ioscan -kfnC fc` command:

```
# ioscan -fknC fc
Class     I  H/W Path       Driver S/W State   H/W Type     Description
====================================================================
fc        0  0/0/0/3/0/0/0  fcd   CLAIMED     INTERFACE    HP SN1000Q 16Gb Dual Port
PCIe Fibre Channel Adapter (FC Port 1)
                           /dev/fcd0
fc        1  0/0/0/3/0/0/1  fcd   CLAIMED     INTERFACE    HP SN1000Q 16Gb Dual Port
PCIe Fibre Channel Adapter (FC Port 2)
                           /dev/fcd1
```

The following sample shows whether NPIV is supported on the VSP:

```
# /opt/fcms/bin/fcmsutil /dev/fcd0

  Vendor ID is = 0x1077
  Device ID is = 0x2031
  PCI Sub-system Vendor ID is = 0x103C
  PCI Sub-system ID is = 0x17E8
  PCI Mode = PCI Express x4
  ISP Code version = 8.1.80
  ISP Chip version = 2
  Topology = PTTOPT_FABRIC
  Link Speed = 16Gb
  Local N_Port_id is = 0x690d00
  Previous N_Port_id is = None
  N_Port Node World Wide Name = 0x50014380231c4dc5
  N_Port Port World Wide Name = 0x50014380231c4dc4
  Switch Port World Wide Name = 0x200d0027f84f7fa8
  Switch Node World Wide Name = 0x10000027f84f7fa8
  N_Port Symbolic Port Name = hpsen6_fcd0
  N_Port Symbolic Node Name = hpsen6_HP-UX_B.11.31
  Driver state = ONLINE
  Hardware Path is = 0/0/0/6/0/0/0
  Maximum Frame Size = 2048
  Driver-Firmware Dump Available = NO
  Driver-Firmware Dump Timestamp = N/A
  TYPE = PFC
  NPIV Supported = YES
  Driver Version = @(#) fcd B.11.31.1603 Dec  3 2015
```

If NPIV is supported on running the command again with the `npiv_info` option provides information whether the physical HBA can support bandwidth entitlement or not.

```
  # /opt/fcms/bin/fcmsutil /dev/fcd0 npiv_info
PFC Hardware Path              = 0/0/0/3/0/0/0
PFC DSF                        = /dev/fcd0
PFC Class Instance             = 0
PFC Driver state               = ONLINE
PFC Port WWN                   = 0x50014380231c4e60
PFC Node WWN                   = 0x50014380231c4e61
PFC Switch Port WWN            = 0x200d000533da5c9a
PFC Switch Node WWN            = 0x1000000533da5c9a

FlexFC Virtual Fibre Channel (VFC)
----------------------------------
Maximum Supported FlexFC VFC   = 16
```

```
Number Active FlexFC VFC       = 0

HPVM Virtual Fibre Channel (VFC)
----------------------------------
Maximum Supported HPVM VFC     = 48
Number Active HPVM VFC         = 0

NPIV QOS Enabled               = Yes
Maximum supported HPVM QOS VFC = 0x7
Number Active HPVM QOS VFC     = 0x0
NPIV QOS Bandwidth in use      = 0%
```

Alternatively, executing the `hpvmstatus` command with the `-n` option provides information about all physical HBA Fibre Channel ports which support NPIV.

NOTE: The bandwidth entitlement statistics reported by FC driver tools (fcmsutil or fcdutil), and the `hpvmstatus` command have the following discrepancies:

- The `hpvmstatus` CLI command reports bandwidth entitlement statistics more accurately for the NPIV HBAs, whereas the FC driver tools display only the integral part of the bandwidth entitlement for the NPIV HBAs.
- The bandwidth entitlement reserved for the physical FC (pFC) port is displayed appropriately in the `hpvmstatus` CLI command, whereas the FC driver tools do not report this reservation for pFC in the generated statistics report.

Specifying an NPIV HBA resource with bandwidth entitlement

Specify an NPIV HBA resource with bandwidth entitlement in the following format:

```
devicetype:adaptertype:bus,device,vWWP,vWWN:storage:device:percent
where:
percent     Bandwidth entitlement of an NPIV HBA.
```

Creating and managing NPIV HBA with bandwidth entitlement

Enable or disable the QOS mode on the physical HBA

If the Quality of Service (QOS) mode is enabled on physical HBA, NPIV HBA can be created on it. If the QOS mode is disabled, then NPIV HBA without entitlement can be created with bandwidth entitlement. NPIV HBA with and without bandwidth entitlement cannot coexist on the same physical HBA.

The state of QOS mode on the physical HBA can be changed from on to off, or from off to on, but it succeeds only when there are no active vFCs.

Example 14 Enable QOS mode on physical HBA

```
# fcmsutil /dev/fcd0 set_qos 1
```

In this example, QOS mode is enabled on the physical HBA. On this physical HBA, NPIV HBAs can be created only with bandwidth entitlement.

Example 15 Disable QOS mode on physical HBA

```
# fcmsutil /dev/fcd0 set_qos 0
```

In this example, QOS mode is disabled on the physical HBA. NPIV HBA can be created only without bandwidth entitlement on this physical HBA.

Adding NPIV HBA resource with bandwidth entitlement

You can specify an NPIV HBA resource with bandwidth entitlement when creating the HP-UX vPars and Integrity VM guest, or after creating the HP-UX vPars or Integrity VM guest, or when it is online. For the resource string format, see “Specifying an NPIV HBA resource with bandwidth entitlement” (page 112).

IMPORTANT: Before creating an NPIV HBA, ensure the physical HBAs on the system support NPIV, and the QOS mode is enabled on the physical HBA.

Example 16 Create an NPIV HBA with bandwidth entitlement using the GUID server for WWNs on QOS disabled physical HBA

Create an HP-UX vPar named vpar1 with an NPIV HBA created on /dev/fcd0, with an entitlement of 25% on QOS enabled physical HBA, using the GUID server to assign port and node WWNs. You can also use the `hpvmstatus -n` command to display NPIV capable fibre channel devices which also support bandwidth entitlement.

```
# fcmsutil /dev/fcd0 npiv_info | grep "QOS Enabled"
#
```

QOS mode is disabled on this pFC.

```
# vparcreate -P vpar1 -a hba:avio_stor::npiv:/dev/fcd0:25
vparcreate: ERROR (vpar1): Bandwidth entitlement is not supported on physical device: '/dev/fcd0'.
vparcreate: Unable to create device hba:avio_stor::npiv:/dev/fcd0:25.
vparcreate: Unable to modify vPar or VM 'vpar1'.
vparcreate: Unable to modify the vPar.
#
```

Example 17 Create an NPIV HBA with bandwidth entitlement by manually specifying WWNs on QOS enabled physical HBA

Add an NPIV HBA created on /dev/fcd1 using a virtual port WWN of 0x50014C2000000006 and virtual node WWN of 0x50014C2800000006, bandwidth entitlement of 25%, to the HP-UX vPar named vPar1, whose vPar ID is 1. Obtain the port and node WWNs from your storage administrator or other source.

```
# fcmsutil /dev/fcd0 npiv_info | grep "QOS Enabled"
NPIV QOS Enabled              = Yes
#
# vparmodify -p 1 -a hba:avio_stor:,,0x50014C2000000006,
0x50014C2800000006:npiv:/dev/fcd1:25
```

In the resource string, you can skip the bus and slot numbers for an NPIV HBA. VSP picks the available bus and slot number for the NPIV HBA. If the VSP is not configured to obtain WWNs from a GUID server, you cannot skip the virtual node WWN and port WWN.

Viewing NPIV resources with bandwidth entitlement

The `hpvmstatus` command output includes the NPIV HBA with bandwidth entitlement in the I/O details for vPars and VM guests that have NPIV HBAs configured with bandwidth entitlement.

Example 18 Determine NPIV HBA bandwidth entitlement details for a VM named guest1

```
# hpvmstatus -P guest1 -d
[Virtual Machine Devices]

[Storage Interface Details]
disk:avio_stor:0,0,0:file:/boot_disks/boot_disk1
hba:avio_stor:0,5,0x50014C2000000007,0x50014C2800000007:npiv:/dev/fcd1
:25
hba:avio_stor:0,6,0x50014C2000000009,0x50014C2800000009:npiv:/dev/fcd2
:50

[Network Interface Details]
network:avio_lan:0,1,0x261D1E8F73E3:vswitch:vswitch1:portid:2

[Direct I/O Interface Details]

[Misc Interface Details]
 serial:com1::tty:console
```

The bandwidth entitlements are:

- 25% for NPIV HBA whose port WWN is 0x50014C2000000007
- 50% for NPIV HBA whose port WWN is 0x50014C2000000009

Example 19 Determine NPIV HBA bandwidth details using hpvmstatus –V command

```
# hpvmstatus -P guest1 –V
 ...
Device type               : hba
Adapter type              : avio_stor
Ioscan format             : 0/0/5/0
Bus                       : 0
Device                    : 5
Function                  : 0
NPIV WWNs (port_id,node_id)     :
0x50014C2000000007,0x50014C2800000007
Lun                       : 0
Physical Device           : /dev/fcd1
Bandwidth Entitlement     : 25%

Device type               : hba
Adapter type              : avio_stor
Ioscan format             : 0/0/6/0
Bus                       : 0
Device                    : 6
Function                  : 0
NPIV WWNs (port_id,node_id)     :
0x50014C2000000009,0x50014C2800000009
Lun                       : 0
Physical Device           : /dev/fcd2
Bandwidth Entitlement     : 50%
```

The bandwidth entitlements are:

- 25% for an NPIV HBA whose port WWN is 0x50014C2000000007
- 50% for an NPIV HBA whose port WWN is 0x50014C2000000009

Example 20 Determine NPIV HBA bandwidth details using `hpvmdevinfo` command

```
# hpvmdevinfo -P guest1 –V
...
Virtual Machine Name      : guest1
Virtual Machine Number    : 2
VM Device Type            : hba
VM Adapter Type           : avio_stor
VM bus, device            : [0,5]
Backing Store Type        : npiv
Host Device Name          : /dev/fcd1
VM Device Name            : /dev/gvsd0
NPIV WWNs (port,node)     : '0x50014C2000000007,0x50014C2800000007'
Bandwidth Entitlement     : 25%

Virtual Machine Name      : guest1
Virtual Machine Number    : 2
VM Device Type            : hba
VM Adapter Type           : avio_stor
VM bus, device            : [0,6]
Backing Store Type        : npiv
Host Device Name          : /dev/fcd2
VM Device Name            : /dev/gvsd3
NPIV WWNs (port,node)     : '0x50014C2000000009,0x50014C2800000009'
Bandwidth Entitlement     : 50%
```

The bandwidth entitlements are:

- 25% for an NPIV HBA whose port WWN is 0x50014C2000000007
- 50% for an NPIV HBA whose port WWN is 0x50014C2000000009

Example 21 Determine pFCs that are available for creation of NPIV HBAs with their bandwidth entitlement using `hpvmstatus -n` command

The `hpvmstatus -n` command displays all the pFCs that are configured on the VSP with the bandwidth entitlement capability support and the available bandwidth on each of the pFCs.

```
# hpvmstatus -n
Physical HBA - /dev/fcd0
Bandwidth entitlement support - YES
Bandwidth entitlement for pFC - 12.50%
Bandwidth in use by active NPIVs' - 87.50%
Active NPIV HBAs with entitlement set - 4
Bandwidth available - 0.00%
Label - label1

Physical HBA - /dev/fcd1
Bandwidth entitlement support - NO
Label - DEFAULT_POOL

Physical HBA - /dev/fcd2
Bandwidth entitlement support - YES
Bandwidth entitlement for pFC - 12.50%
Bandwidth in use by active NPIVs' - 0.00%
Active NPIV HBAs with entitlement set - 0
Bandwidth available - 87.50%
Label - DEFAULT_POOL

Physical HBA - /dev/fcd3
Bandwidth entitlement support - YES
Bandwidth entitlement for pFC - 12.50%
Bandwidth in use by active NPIVs' - 0.00%
Active NPIV HBAs with entitlement set - 0
Bandwidth available - 87.50%
Label - DEFAULT_POOL

Physical HBA - /dev/fclp4
Bandwidth entitlement support - NO
Label - label2

Physical HBA - /dev/fclp5
Bandwidth entitlement support - NO
Label - DEFAULT_POOL

Physical HBA - /dev/fcd6
Bandwidth entitlement support - NO
Label - DEFAULT_POOL

Physical HBA - /dev/fcd7
Bandwidth entitlement support - NO
Label - DEFAULT_POOL

#
```

Modifying bandwidth entitlement of an NPIV HBA

The `vparmodify` or `hpvmmodify` command is used to modify bandwidth entitlement of an NPIV HBA on a vPar or VM guest. To modify bandwidth entitlement, you must either stop and restart the vPar or VM guest or do an online deletion followed by addition of the vHBA. The following operations are supported:

- Modify bandwidth entitlement of an NPIV HBA.
- Modify an NPIV HBA with bandwidth entitlement to an NPIV HBA without bandwidth entitlement.
- Modify an NPIV HBA without bandwidth entitlement to an NPIV HBA with bandwidth entitlement.

Example 22 Modify bandwidth entitlement of an NPIV HBA

```
# hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :25

 # hpvmmodify -P guest2 -m
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :50

 # hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :50
```

In this example, initial bandwidth entitlement of an NPIV HBA, whose port WWN is 0x50014C2000000004, is 25%. The bandwidth entitlement is modified to 50% by running the `hpvmmodify –m` command.

Example 23 Modify an NPIV HBA with bandwidth entitlement into an NPIV HBA without bandwidth entitlement

```
# hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :50

 # hpvmmodify -P guest2 -m
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0

 # hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
```

In this example, the initial bandwidth entitlement of an NPIV HBA is 50%. The NPIV HBA is modified into an NPIV HBA without bandwidth entitlement by running the `hpvmmodify –m` command, provided the physical HBA mode is changed to QOS mode disable to support NPIV HBA without bandwidth.

Example 24 Modify an NPIV HBA without bandwidth entitlement into an NPIV HBA with bandwidth entitlement

```
# hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0

 # hpvmmodify -P guest2 -m hba:
avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :25

 # hpvmstatus -P guest2 -d | grep hba
 hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0
 :25
```

In this example, an NPIV HBA, whose port WWN is 0x50014C2000000004, was initially configured without any bandwidth entitlement. The NPIV HBA is modified into an NPIV HBA with bandwidth entitlement of 25% by running the `hpvmmodify –m` command, provided the physical HBA mode is changed to QOS mode disable to support NPIV HBA without bandwidth.

Deleting configured NPIV HBAs with bandwidth entitlement

The `vparmodify` or `hpvmmodify` command is used to delete an NPIV HBA from a vPar or VM guest.

Example 25 Deleting an NPIV HBA resource

```
# hpvmstatus -P guest1 -d | grep hba
 hba:avio_stor:0,5,0x50014C2000000007,0x50014C2800000007:npiv:/dev/fcd1
 :25
 hba:avio_stor:0,6,0x50014C2000000009,0x50014C2800000009:npiv:/dev/fcd2
 :50

 # hpvmmodify -P guest1 -d
 hba:avio_stor:0,5,0x50014C2000000007,0x50014C2800000007:npiv:/dev/fcd1
 :25
 hpvmmodify: A Dynamic IO deletion operation has been initiated for
 this VM or vPar. Please check hpvmstatus output or syslog for
 completion status.

 # hpvmstatus -P guest1 -d | grep hba
 hba:avio_stor:0,6,0x50014C2000000009,0x50014C2800000009:npiv:/dev/fcd2
 :50
```

In this example, a VM guest named guest1 is configured with two NPIV HBAs with bandwidth entitlement. The NPIV HBA, whose port WWN is 0x50014C2000000007, is configured with 25% bandwidth entitlement. This NPIV HBA is deleted from an online guest by running the `hpvmmodify -d` command.

Ignoring bandwidth entitlement during guest start

Starting HP-UX vPars and Integrity VM v6.4 with PK2 or superseding patches supports (listed in “Software dependencies ” (page 109)), a new per guest configuration parameter *ignore_npiv_entitlement* is introduced. When this parameter is enabled, an NPIV HBA, whose bandwidth entitlement cannot be assigned, is not created on the pFC. This guest option is disabled by default. You must enable this parameter before starting the guest. If the parameter is enabled on a live guest, the change gets updated in the guest configuration.

To enable this parameter, run the following command on the desired guest:

```
# hpvmmodify -p <guest_id> -x ignore_npiv_entitlement=enabled
```

To disable this parameter, run the following command on the desired guest:

```
# hpvmmodify -p <guest_id> -x ignore_npiv_entitlement=disabled
```

The following example illustrates the usage the *ignore_npiv_entitlement* parameter.

Example 26 Usage of ignore_npiv_entitlement

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
DEMO-GUEST               1 SH   HPUX    Off            2     6     1    6 GB
guest1                   2 SH   HPUX    On (OS)        1     3     1    2 GB
guest2                   3 VP   HPUX    Off            1     1     0    2 GB
vPar1                    4 VP   HPUX    Off            1     2     0    2 GB

# hpvmstatus -P guest1 -d | grep hba
hba:avio_stor:0,5,0x50014C200000000A,0x50014C280000000A:npiv:/dev/fcd0:75
hba:avio_stor:0,6,0x50014C2000000009,0x50014C2800000009:npiv:/dev/fcd2:50

# hpvmstatus -P guest2 -d | grep hba
hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0:25

# hpvmstatus -n | head -8
Physical HBA - /dev/fcd0
   Bandwidth entitlement support - YES
   Bandwidth entitlement for pFC - 12.50%
   Bandwidth in use by active NPIVs' - 75.00%
   Active NPIV vHBAs with entitlement set - 1
   Active NPIV vHBAs without entitlement - 0
   Bandwidth available – 12.50%
   Label - DEFAULT_POOL

# hpvmstart -P guest2
vPar/VM guest2 configuration problems:
  Error 1 on item /dev/fcd0: The limit for bandwidth associated with
physical device: '/dev/fcd0' will be exceeded.
hpvmstart: Unable to continue.

# hpvmmodify -P guest2 -x ignore_npiv_entitlement=enabled
vPar/VM guest2 configuration problems:
   Warning 1 on item /dev/fcd0: The limit for bandwidth associated with
physical device: '/dev/fcd0' will be exceeded.
These problems may prevent the vPar or VM guest2 from starting.
hpvmmodify: The modification process is continuing.

# hpvmstart -P guest2
(C) Copyright 2000 - 2014 Hewlett-Packard Development Company, L.P.
Mapping vPar/VM memory: 2048MB
...
Starting thread initialization
(UsrVsdAddLun) 23, 2, 28: Ioctl error to VSD driver. caller: UsrVsdInitHost
Warning initializing VSD HBA:
Possible causes:
- Physical device(/dev/fcd0) - Bandwidth unavailable on physical device.
...
hpvmstart: Successful start initiation of vPar or VM 'guest2'

# hpvmstatus -P guest2 -d | grep hba
hba:avio_stor:0,0,0x50014C2000000004,0x50014C2800000004:npiv:/dev/fcd0:25*

# hpvmstatus -P guest2 -V | grep Bandwidth
Bandwidth Entitlement   : 25%*

# hpvmdevinfo -P guest2 -V | grep Bandwidth
Bandwidth Entitlement     : 25%*
```

In this example, a VM guest named guest1 is configured with an NPIV HBA, whose port WWN is 0x50014C200000000A. The bandwidth entitlement of this NPIV HBA is 75%. The `hpvmstatus -n` output displays the bandwidth available on the physical HBA /dev/fcd0 to be 12.50%. The second VM guest named guest2 has an NPIV HBA configured to it whose port WWN is 0x50014C2000000004. It has a bandwidth entitlement of 25%, which is more than the available bandwidth on the physical HBA /dev/fcd0.

When the VM guest named guest2 is started, this operation fails because the bandwidth entitlement of the NPIV HBA configured cannot be honored. Then, the *ignore_npiv_entitlement* parameter is enabled, and the VM is started. After enabling the parameter, the failure for creation of NPIV HBA is ignored and the VM guest starts up.

The `hpvmstatus -p <guest_id> -d`, `hpvmstatus -p <guest_id> -V`, and `hpvmdevinfo -p <guest_id> -V` commands display this change with an '*' against the desired bandwidth entitlement value. The '*' against the bandwidth indicates that the bandwidth capped NPIV HBA is not created.

Migrating VM and vPar guests with NPIV HBAs

vPars and Integrity VM v6.1.5 and later versions support online migration of VM guests and offline migration of vPars and VM guests with NPIV HBAs across an FC fabric. After the migration, the NPIV HBAs of a particular guest might not always be spread across different pFCs ports or adapters on the target host. For more information about the modified behavior, see "Selecting physical HBA ports during migration with NPIV HBAs" (page 218).

Starting from v6.3.5, the placement of NPIV HBAs of a guests across physical FC ports on the target VSP post migration has become more predictable. For more information about migration, see "Migrating VMs and vPars" (page 203).

Troubleshooting NPIV storage problems

For more information about troubleshooting NPIV storage problems, see "NPIV storage devices" (page 293).

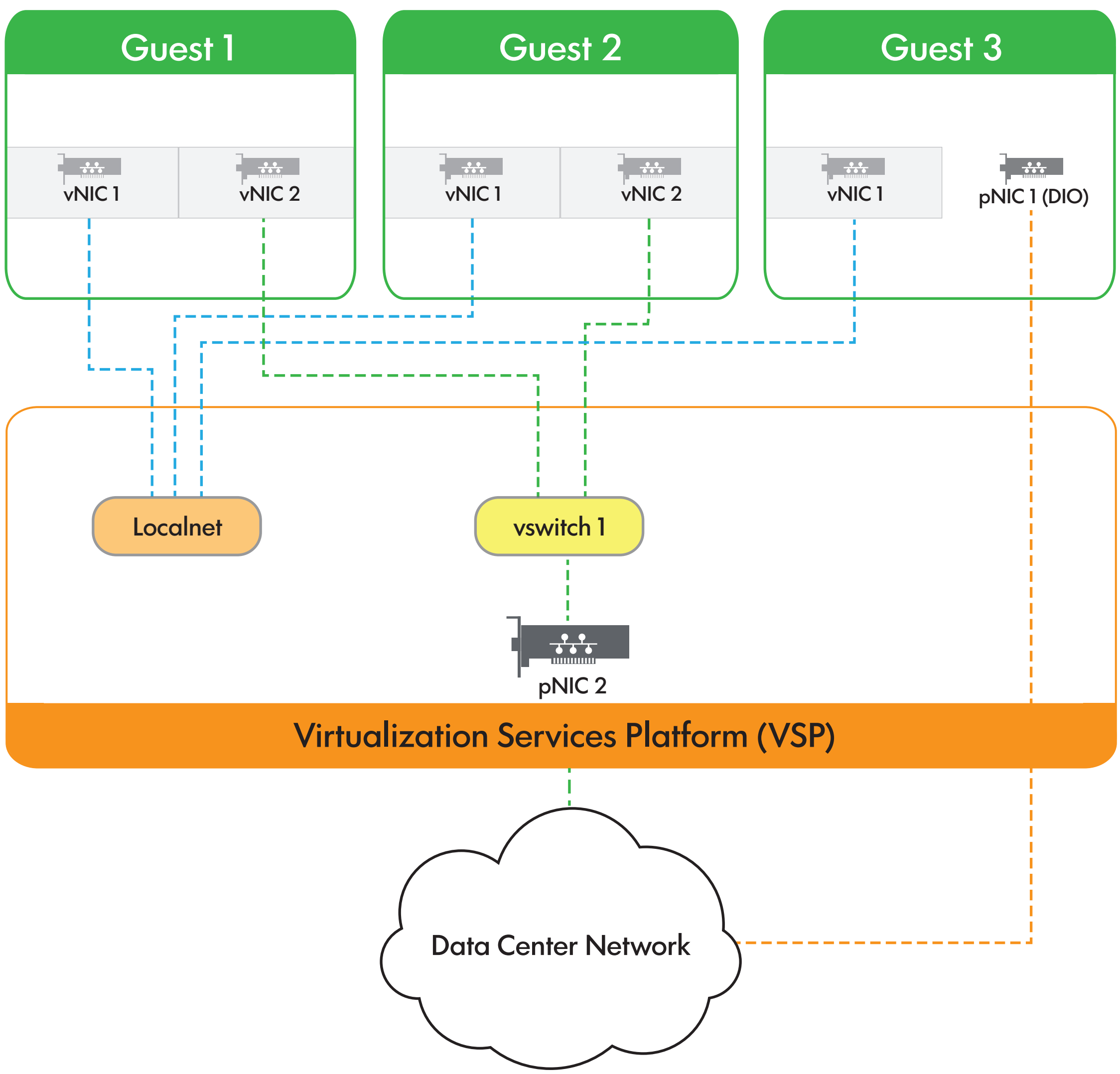
8 Creating virtual and direct I/O networks

The vPars and Integrity VM supports two types of networking I/O: AVIO and DIO. With AVIO networking, the I/O device drivers for the devices in the guest operating system are virtualization aware, eliminating some of the virtualization overhead. However, the guest operating system still does not have direct visibility to the underlying hardware, and the remaining virtualization overhead prevents the guest from achieving near native performance for certain I/O intensive workloads. With DIO networking, which is supported on HPE Integrity Server Blade system BL8x0c i2/i4, HPE Integrity Superdome 2 i2/i4, and rx2800 i2/i4, a vPar and a VM can have direct control of the I/O device. The DIO networking feature minimizes the device emulation overhead and also allows guest operating systems to control devices for which emulation does not exist, thus enabling access to I/O hardware technology without requiring the support of either vPars or Integrity VM.

NOTE: Both AVIO and DIO networking support HPE Virtual Connect.

The basic network configuration with a combination of virtual and direct I/O network interfaces is illustrated in Figure 14 (page 122).

Figure 14 Virtual and DIO network configuration



Introduction to AVIO network configuration

The guest virtual network configuration provides flexibility in network configuration, allowing you to provide high availability, performance, and security to the vPars or VM guests running on the VSP.

The virtual network configuration consists of the following components:

- VSP pNIC – the physical network adapter, which might be configured with APA. (For more information about APA, see *HP Auto Port Aggregation (APA) Support Guide*.)

  NOTE: Trunking software such as APA is supported on the DIO interfaces in the guest. Trunking of AVIO interfaces is not supported on the guest.

  You can configure APA on the VSP to provide a highly available fault-tolerant LAN for the vswitch (APA in active or passive mode) or to increase the bandwidth of the vswitch LAN (APA active or active mode). Before you stop APA, use the `hpvmnet -h` command to halt the vswitch. If you do not halt the vswitch first, the `hpvmnet` command reports an incorrect MAC address for the vswitch.

- Guest vNIC — the virtual network adapter, as recognized by the guest operating system.
- Virtual switch — the virtual network switch that is associated with a pNIC. This is maintained by the VSP, and can be allocated to one or more guests.

CAUTION: You must not connect the vswitches to the network devices that are set to promiscuous mode and do not run applications such as `tcpdump` on the VSP on interfaces that are used for virtual switches.

Using redundant pNICs and APA, you can ensure high availability of the guest networks and provide greater capacity for the VSP system that is running many guests with network intensive applications.

You can configure HP-UX VLANs for the guests. VLANs isolates broadcast and multicast traffic by determining the targets that must receive that traffic, thereby making better use of switch and end-station resources. With VLANs, broadcasts and multicasts go only to the intended nodes in the VLAN.

Creating virtual networks

You can allocate virtual network devices or vNICs to the vPar or VM guest when you create them with the `hpvmcreate` command or when you modify an existing vPar or VM guest using the `hpvmmodify` command, as described in "Administering vPars" (page 164) and "Administering VMs" (page 145). To add a vNIC to a guest, use the following command option:`-a network:adaptertype:bus,device,mac-addr:vswitch:vswitch-name:portid:portnumber`

However, before you allocate the vswitch to the vPar or VM guest, you must create the vswitch using the `hpvmnet` or `vparnet` command.

Creating and managing vswitches

The following sections describe how to create, modify, delete, and manage vswitches.

Creating vswitches

To allow guests to access network devices, you must create vswitches on the VSP.

To create vswitches, use the `hpvmnet` command. The following is the basic format of the `hpvmnet` command to create a vswitch:

hpvmnet -c -S *vswitch-name* -n *nic-id*

where

-c

indicates the creation of a vswitch.

-S

vswitch-name specifies the name of the vswitch.

-n

nic-id specifies the network interface on the VSP that the new vswitch uses. For example, `-n 0` indicates `lan0`. Network interfaces are displayed by the `nwmgr` command. If you do not include the `-n` option, a local vswitch is created, as described in "Local networks" (page 126).

The `hpvmnet` command also allows you to view and manage the vswitches on the VSP. Table 17 (page 124) lists the options that can be used with the `hpvmnet` command.

Table 17 Options to the hpvmnet command

Option	Description
-b	Boots a vswitch. The vswitch must be booted before it can accept network traffic. All vswitches are booted automatically when Integrity VM is started.
-c	Creates a new vswitch.
-h	Halts one or all vswitches. You must confirm this action.
-F	Omits the confirmation dialog before halting, deleting, or rebooting the vswitch. This option is intended for use by scripts and other non-interactive applications (Force mode). **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.
-d	Deletes a virtual switch. You must confirm this action.
-n *nic-id*	Specifies the network interface on the VSP that the new vswitch uses. For example, to associate a vswitch to `lan0`, enter `-n 0`.
-p *n*	Specifies the port number. To view information about all ports, enter `-p all`.
-Q	Specifies the command function that must proceed without confirmation. By default, the command requires confirmation, and does not proceed without it.
-r	Restarts the vswitch.
-s *vswitch_number*	Specifies the vswitch by its number.
-S *vswitch_name*	Specifies the vswitch by name. The vswitch name can be up to 64 characters and must be unique on the VSP.
-u portid:*portnum*:vlanid:[*vlanid* | none]	Configures the port *portnum* on the virtual switch so that it is isolated to the VLAN specified by *vlanid*. For more information about VLAN, see Section (page 131).
-i	Enables the list of VLAN ids on the list of ports. Specifying *all* allows you to enable all VLANs at once.
-A	Displays information about vswitches in verbose mode. If you specify the vswitch using either the `-S` or `-s` options, network counters are included in the display.
-o	Disables the list of VLAN ids on the list of ports. Specifying *all* disables all VLANs at once.
-Z	Used with the `-A` option, clears statistics after retrieving them.

Table 17 Options to the `hpvmnet` command *(continued)*

Option	Description
`-M`	Displays verbose resource information in a machine-readable format.
`-X`	Displays verbose resource information in XML format.
`-V`	Enables verbose mode, displaying detailed information about one or all vswitches.
`-v`	Displays the version number of the `hpvmnet` command in addition to the vswitch information.
`-C`	Changes the specified vswitch. If used with the `-N` option, the changes are made to the cloned vswitch. You must include either the `-S` or `-s` option.
`-N` *`new-vswitch-name`*	Creates a new vswitch based on the existing vswitch. For *`new_vswitch_name`*, specify the unique name of the new virtual switch. The name of the vswitch can be up to 64 characters. You must include either the `-S` or `-s` option.

NOTE: When working with vPars, you can also use the `vparnet` command. For more information about using the `vparnet` command, see the *vparnet*(1M).

The following command creates a virtual switch called `clan1` that is associated with `lan1`. The second `hpvmnet` command displays information about all the vswitches.

```
# hpvmnet -c -S clan1 -n 1
# hpvmnet

Name     Number State   Mode      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared           N/A            N/A
myswitch      2 Up      Shared           N/A            N/A
clan1         5 Down    Shared    lan1
```

The physical point of attachment (PPA) for `clan1` is `1`. Two vswitches (`localnet` and `lan0`) communicate over the localnet.

To boot a vswitch, enter the `hpvmnet` command with the `-b` option. For example, to boot the vswitch named `clan1`, enter the following command:

```
# hpvmnet -S clan1 -b
# hpvmnet -v

Name     Number State   Mode      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared           N/A            N/A
myswitch      2 Up      Shared           N/A            N/A
clan1         5 Up      Shared    lan1   0x00306e3977ab
```

NOTE: The `clan1` vswitch is associated with the network interface on the VSP that has MAC address `0x00306e3977ab` (this is not the MAC address of any VM connected to this vswitch).

For more information about connecting vswitches to guests, see “Administering VMs” (page 145). For more information about modifying virtual networks, see “Adding vNICs” (page 130).

You can create multiple vswitches associated with the same host physical NIC. However, you cannot boot (`hpvmnet —b`) more than one of them at the same time.

NOTE: The Cisco switch for HPE BladeSystem c-Class Server Blades has a protocol error that causes it to respond to every MAC address. Because MAC addresses are unique, Integrity VM verifies that the generated guest virtual MAC address is unique. If one of these bad switches is on your network, the Integrity VM verification fails.

The `hpvmcreate` command might fail with the following messages:

```
hpvmcreate:  WARNING (host): Failed after 3 attempts.
hpvmcreate:  WARNING (host): Unable to create Ethernet MAC Address.
```

Similarly, the `hpvmstart` command might fail with the following messages:

```
# hpvmstart -P vm2
HPVM guest vm2 configuration problems:
Warning 1 on itme nic1: Guest MAC address for switch nic1 is in use.
```

Cisco Systems, Inc. released a fix for the Cisco Catalyst Blade Switch 3020 in December 2006, which is available from the Cisco Systems website:

http://cco.cisco.com

NOTE: This link will take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise does not control and is not responsible for information outside of **http://www.hpe.com**.

The fix is also available from the HPE website:

http://www.hpe.com

From the HPE website, select Software & Driver downloads and search for switch cisco 3020. The minimum required firmware version is 12.2(35) SE.

Local networks

Virtual network communication might be limited to VMs on the VSP system through the use of vswitches that are not connected to a physical NIC. A virtual network such as this is called a local virtual network or a local network (`localnet`). To create a local network, a vswitch must first be created using the `hpvmnet` command without the `-n` option, so that it is not connected to the physical network. For example, to create a local network vswitch named `clan0`, and to start it, enter the following commands:

```
# hpvmnet -c -S clan0
# hpvmnet -b -S clan0
```

All vNICs connected to that vswitch will then be on the same local network. The VSP does not communicate on local networks.

The following command adds a vNIC to the guest `host1`, which can be used to communicate with any VM connected to the `localnet` vswitch.

```
# hpvmmodify -P host1 -a network:avio_lan::vswitch:clan0
```

During startup of the Integrity VM software, a default vswitch, localnet, is created and booted. The localnet vswitch can be added to a guest, which allows communication with any other guest using the localnet vswitch. For example,

```
# hpvmmodify -P compass1 -a network:avio_lan::vswitch:localnet
```

Changing vswitches

You can use the `-C` option to change the pNIC, which the guest uses. For example, enter the `nwmgr` command as follows:

```
# nwmgr
Name/          Interface Station        Sub-     Interface      Related
ClassInstance  State     Address        system   Type           Interface
============== ========= ============== ======== ============== =========
lan0           UP        0x00306E4A93E6 iexgbe   10GBASE-KR
```

```
lan1           UP       0x00306E4A92EF iexgbe   10GBASE-KR

# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared             N/A            N/A
hostnet     296 Up      Shared    lan0     0x00306e4a93e6
```

If `lan0` goes down, enter the following command to swap to use `lan1`:

```
# hpvmnet -C -S hostnet -n 1
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared             N/A            N/A
hostnet     296 Up      Shared    lan1     0x00306e4a92ef
```

Cloning vswitches

Using the `-N` option with the `-C` option creates a new vswitch based on the changed vswitch information. For example, the following command sequence displays the current vswitch (`vmvlan`), modifies the vswitch to specify connection to `lan1`, and creates a new vswitch named `clnvlan`. The final command displays information about the new vswitch.

```
#hpvmnet -S vmvlan
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
vmvlan       13 Up      Shared    lan900   0x00306e4bc7bf

[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
1       Reserved     none     1
2       Reserved     20       1
3       Reserved     none     1

# hpvmnet -C -S vmvlan -n 1 -N clnvlan
# hpvmnet -S clnvlan
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
clnvlan     320 Down    Shared    lan1

[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
2       Available    20       0
```

NOTE: Only the configured VLAN port identification data is copied to the new vswitch. You can use the `hpvmnet` command when you have a vswitch with numerous VLAN ports. This process makes it unnecessary to re-enter all the port data for each new vswitch.

Deleting vswitches

To delete a vswitch, first, stop the vswitch using the `-h` option with the `hpvmnet` command. Delete the vNIC from the guests using the `hpvmmodify` command and then, delete the vswitch using the `-d` option with the `hpvmnet` command. For example, the following command shows the error that prevents you from deleting an active vswitch (`clan1`):

```
# hpvmnet -S clan1 -d

hpvmnet: The vswitch is currently active
hpvmnet: Unable to continue
```

The following example uses the `hpvmnet` command to halt the vswitch and then to delete it. Both the commands require you to confirm the action.

```
# hpvmnet -S clan1 -h

hpvmnet: Halt the vswitch 'clan1'? [n/y]: y

# hpvmnet -S clan1 -d

hpvmnet: Remove the vswitch 'clan1'? [n/y] y
```

The default command function (if you press **Enter**) is to not perform the function of the command. To perform the command function, enter **y**.

In the case of commands where a confirmation is required, such as the `hpvmnet -h` command, you can include the `–Q` option to override the confirmation process. This is useful in scripts and processes that are not interactive. For example, to stop a vswitch (`clan1`) without requiring confirmation from the user, enter the following commands:

```
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared             N/A            N/A
clan1         2 Up      Shared    lan0     0x00306e39f70b
# hpvmnet -S clan1 -h -Q
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared             N/A            N/A
clan1         2 Down    Shared    lan0
```

When an active vswitch backing interface goes offline, the VSP automatically determines that the vswitch backing interface is gone. When the backing interface becomes online the guest network automatically becomes functional.

Recreating vswitches

To change the vswitch to use another pNIC on the VSP (for example, to change from `lan0` to `lan1`),

1. Delete the vswitch associated with `lan0`. For example,

   ```
   # hpvmnet -S myswitch -h -Q
   # hpvmnet -S myswitch -d
   ```

2. Create a new vswitch associated with `lan1`. For example,

   ```
   # hpvmnet -S myswitch -c -n 1
   ```

3. Add a new vNIC to your guest using the new vswitch. For example,

   ```
   # hpvmmodify -P guestname -a network:avio_lan:,,:vswitch:myswitch
   ```

Starting vswitches

Virtual switches (vswitches) start automatically when the VSP system is started. You can start the vswitch manually using the `–b` option with the `hpvmnet` command. For example, the following command boots the vswitch named `clan1`:

```
# hpvmnet -S clan1 -b
```

You must restart a vswitch after the following events:

- The MAC address corresponding to the LAN number being used by the virtual switch is changed on the VSP (either by swapping the network adapter associated with the vswitch or associating the vswitch with a different network adapter).
- The way the network adapter accepts and passes on packets to the next network layer is changed. This can occur as a result of using the `ifconfig` or `lanadmin` command to set the checksum offloading (CKO) to on or off.
- If you use the `hpvmmodify` command to change the adapter type for a virtual NIC (vswitch port).

Halting vswitches

You can use the `hpvmnet -h` command to halt a vswitch. For example,

```
# hpvmnet -S clan1 -h
hpvmnet: Halt the vswitch 'clan1'? [n]: y
```

APA can be configured on the VSP to provide a highly available LAN for the vswitch (APA in active or passive mode) or to increase the bandwidth of the vswitch LAN (APA active or active mode). Before you stop APA, halt the vswitches associated with it. If you do not bring down the vswitch first, the `hpvmnet` command reports an incorrect MAC address for the vswitch.

Restarting vswitches

You must restart a vswitch when you do any one or more of the following:

- Replace the physical network card associated with the vswitch.
- Change a VSP IP address associated with the network interface card of the vswitch.
- Change the network interface characteristics on the VSP. For example, by using the `nwmgr` command to change CKO.

When you restart a vswitch, it is not necessary to restart the guests using the vswitch.

Guest AVIO interface behavior

The following list describes the guest AVIO interface behavior when guest boots while vswitch is down or resetting:

- If you boot a guest when the vswitch is not up, AVIO interfaces associated with the vswitch might not be claimed in the guest. For example, this might occur if the guest is booted prior to booting the vswitch or if the corresponding network interface on the VSP is not cabled. If you encounter this problem, first, fix the vswitch state (that is, ensure that `hpvmnet` displays its state as Up), and then run the `ioscan` command in the guest. These actions claim the AVIO interfaces.
- If the vswitch is in an unstable state while the guest is booting, guest AVIO interfaces might fail initialization and move to the DOWN state (as displayed by the `nwmgr` command). When this occurs, first, ensure that the vswitch enters a stable state, and then reset the guest interface using the `nwmgr` command.

Managing vNICs

After you create the vswitch, you can allocate it to one or more VMs for use by guest operating systems and applications. To create a vNIC for a VM, enter one of the following commands:

- To create a new VM with one vswitch:

  ```
  # hpvmcreate -P vm-name -a network:adapter-type:[hardware-address]:vswitch:vswitch-name
  ```

- To create a new VM based on the configuration of an existing VM:

```
# hpvmclone -P vm-name -N clone-vm-name —a network:adapter-type:[hardware-address]:vswitch:vswitch-name
```

The vNIC specified with this command is added to the new VM.

- To modify an existing VM:

  ```
  # hpvmmodify -P vm-name —a network:adapter-type:[hardware-address]:vswitch:vswitch-name
  ```

  The `—a` option adds the specified vNIC to the VM.

As with virtual storage devices, use the `-a` *`rsrc`* option to associate a guest virtual network device with a vswitch. Before you use this option to associate the virtual network device with a vswitch, create the vswitch using the `hpvmnet` command. The format of the *`rsrc`* parameter for network devices is:

`network:`*`adapter-type`*`:[`*`hardware-address`*`]:vswitch:`*`vswitch-name`*

The guest virtual network device information consists of the following fields, separated by colons:

- `network`
- *adapter-type*, which can be `avio_lan`
- `[`*`hardware-address`*`]` (optional), formatted as *`bus,device,mac-addr`*. If you do not specify the hardware address, or a portion of it, the information is generated. Hewlett Packard Enterprise recommends allowing Integrity VM to generate the hardware address. The hardware address consists of the following information:
  - *bus* (virtual network device PCI bus number)
  - *device* (virtual network device PCI slot number)
  - *mac-addr* (the virtual network device MAC address) in one of the following formats: `0xaabbcc001122` or `aa-bb-cc-00-11-22`. The MAC address that you enter is verified to ensure that it does not conflict with any of the physical network adapter MAC addresses of the VSP.
- *`vswitch`*

  The virtual switch information is formatted as `vswitch:`*`vswitch-name`* (where *`vswitch-name`* is the name assigned to the virtual network switch when you created it using the `hpvmnet` command).

Adding vNICs

You can define a vNIC for a guest using the `hpvmmodify` command. For example, the following command adds a vNIC to the guest named `host1` either dynamically or to a guest in offline mode:

```
# hpvmmodify -P host1 -a network:avio_lan:0,0,0x00306E39F70B:vswitch:clan1
```

The guest configuration file `/var/opt/hpvm/guests/`*`guestname`*`/vmm_config.current` contains an entry for each guest virtual network device. When the guest is booted (through the `hpvmstart` or `hpvmconsole` command), the guest LAN is configured as specified in the LAN entry in the guest configuration file. For example,

```
.
.
.
# Virtual Network Devices
#
lan(0,0).0x00306E39F70B = switch(clan1).4
.
.
.
```

The `localnet` vswitch can be used as a local network, and vNICs can be specified for a guest. For example,

```
# hpvmmodify -P host1 -a network:avio_lan::vswitch:clan0
```

NOTE: Never directly modify the guest configuration files. Always use the Integrity VM commands to modify the virtual devices and VMs. Failure to follow this procedure results in unexpected problems when guests are started.

The virtual network entry in the guest configuration file includes the guest information on the left side of the equal sign (=), and VSP information on the right. The data about the guest LAN example includes the following information:

`lan(0,0)`	Bus 0 and device number 0 indicate the guest LAN hardware path.
`0xEEEE4077E7EB`	Guest virtual MAC address.
`switch(clan1)`	The vswitch name is `clan1`.
`4`	The VLAN port number is 4.

The output of running the `nwmgr` command on the guest `host1`:

```
# nwmgr

Name/          Interface Station        Sub-     Interface      Related
ClassInstance  State     Address        system   Type           Interface
============== ========= ============== ======== ============== =========
lan0           UP        0xEEEE4077E7EB iexgbe   10GBASE-KR
lan1           UP        0x00306E3977AB iexgbe   10GBASE-KR
lan2           UP        0x00306E4CE96E iexgbe   10GBASE-KR
```

NOTE: Do not include the hardware address (for example, bus, device, mac-addr) with the `hpvmmodify` command, because Integrity VM picks an available pcibus, pcislot and generates a random MAC address.

The hardware path from the output of `nwmgr` command on the guest matches the path in the guest configuration file. The `Station Address` in the `nwmgr` output also matches the guest virtual MAC address in the guest configuration file.

Removing vNICs

To remove a vNIC from a configuration of the VM, use the `-d` option with the `hpvmmodify` command. The `-d` option allows you to specify the vswitch and the vNIC information. The following is the syntax of the `hpvmmodify -d` command:

```
# hpvmmodify -P vm-name -d network:adapter-type:[hardware-address]:vswitch:vswitch-name
```

Configuring VLANs

A LAN defines a broadcast domain in which bridges and switches connect all end nodes. Broadcasts are received by every node on the LAN, but not by nodes outside the LAN.

A VLAN defines logical connectivity instead of the physical connectivity defined by a LAN. A VLAN provides a way to partition a LAN logically such that the broadcast domain for a VLAN is limited to the nodes and switches that are members of the VLAN.

VLANs provide the following benefits:

- Enhanced security through traffic isolation within nodes that are VLAN members.
- Bandwidth preservation, limiting the broadcast domain to a VLAN instead of the entire LAN.
- Enhanced manageability for node migrations and network topology changes.

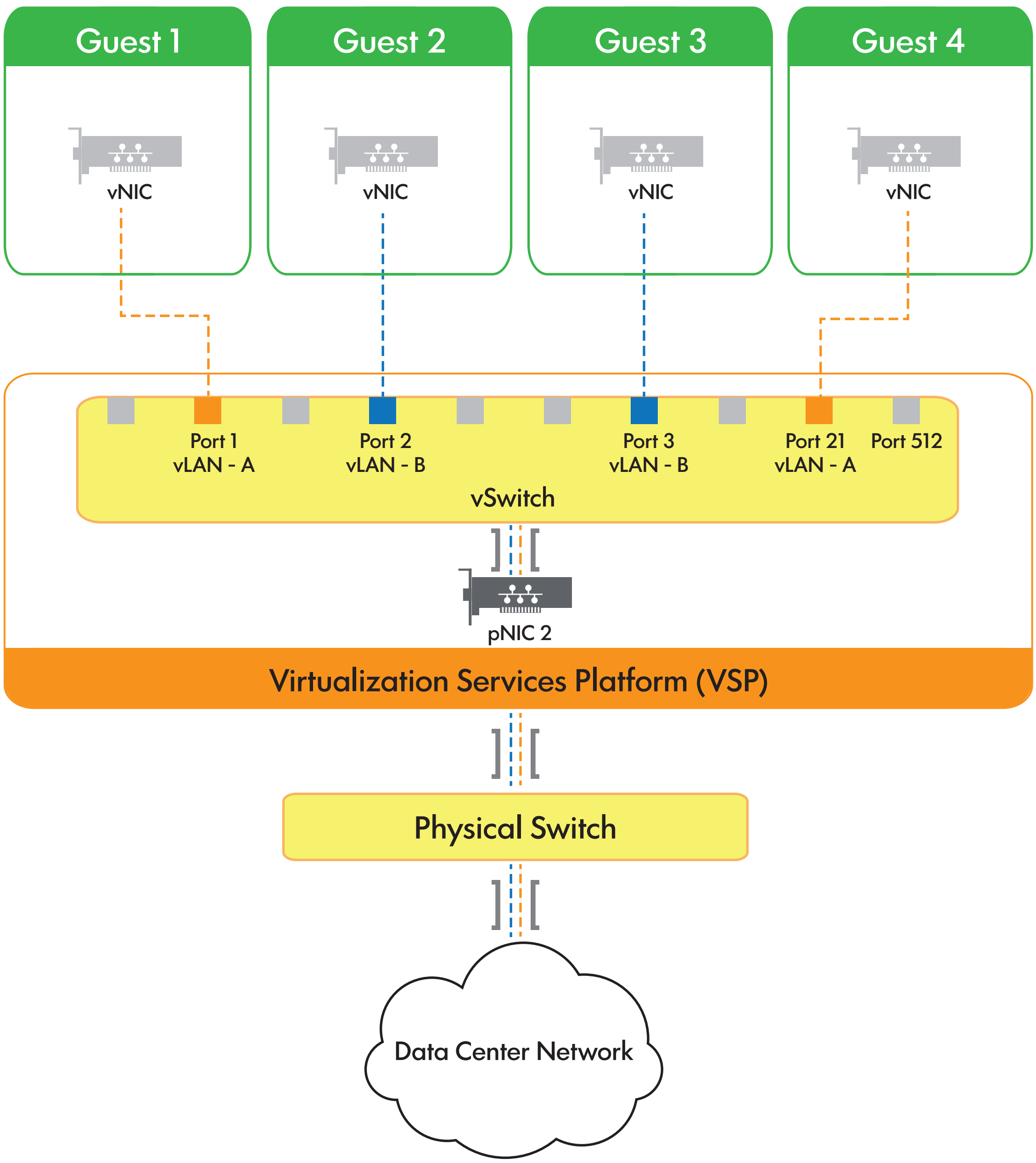
The following sections describe the Port-based VLAN feature, Guest-based VLAN feature, and VLAN-backed vswitch feature.

NOTE: All three features are supported on the AVIO network.

Port-based VLANs

Figure 15 (page 132) shows a basic VM VLAN that allows guests on different VSP systems to communicate.

Figure 15 Integrity VM VLAN configuration example



A vNIC on a guest is associated with a port on the vswitch and all network communication to and from the guest passes through this vswitch port. You can configure VLAN rules on the individual ports of the vswitch, similar to most physical switches. Each VLAN is identified by a VLAN identifier (VLAN ID). The VLAN ID is a number in the range 0 to 4094. A port on the vswitch can be assigned a VLAN ID that identifies the VLAN to which the port (and, therefore, the guest vNIC using that port) belongs.

Ports on a vswitch that are configured for the same VLAN ID can communicate with each other. Ports on a vswitch that are configured for different VLAN IDs are isolated from each other. Ports on a vswitch that do not have any VLAN ID assigned cannot communicate with ports that have a VLAN ID assigned, but can communicate with other ports that do not have VLAN ID assigned. The port IDs for a vswitch can range 0 to 511.

The AVIO network vNIC is presented to guest operating system as PCI-X 1000Base-T with the speed of physical network interface card backing the vswitch. The AVIO emulation can lead to an incorrect calculation of vNIC performance by some network performance application on the guest.

To accurately calculate vNIC performance, consider the speed of the backing device on the Integrity VSP.

If the guest must communicate with the VSP or outside the VSP over a VLAN, additional configuration is necessary. For communication with the VSP, configure a VLAN interface on the VSP interface for that vswitch. This VLAN interface must have the same VLAN ID as the guest port. For more information about configuring VLANs on the VSP, see the using HP-UX VLANs manual. You must not use the `hpvmnet` command to create a virtual switch that is associated with a VLAN port on the VSP (that is, a LAN created with `nwmgr -a -S vlan` or `lanadmin -V`). This “nested VLAN” configuration is not supported.

Frames arriving at the vswitch from a guest can be “tagged” by the vswitch. Tagging consists of inserting the VLAN ID information into the MAC header before forwarding the frame. Tagged frames destined for a guest are always stripped of the tag information in the frame before being forwarded. For Integrity VM, only tag-unaware guests are supported.

To configure a VLAN:

1. Create and start the vswitch. For example, to create and boot vswitch `vmlan4` on `lan1`, enter the following command:

   ```
   # hpvmnet -c -S vmlan4 -n 1
   # hpvmnet -b -S vmlan4
   ```

2. Use the `hpvmnet` command with the `—u` option to create the port, and assign it a VLAN ID. For example, to create ports 1 and 2 for VLAN 100, enter the following command:

   ```
   # hpvmnet -S vmlan4 -u portid:1:vlanid:100
   # hpvmnet -S vmlan4 -u portid:2:vlanid:100
   ```

3. Add the vswitch ports to the guest configuration using the `hpvmmodify` command. For example, to add the new VLAN ports to guests `vm1` and `vm2`, enter the following command:

   ```
   # hpvmmodify -P vm1 -a network:avio_lan::vswitch:vmlan4:portid:1
   # hpvmmodify -P vm2 -a network:avio_lan::vswitch:vmlan4:portid:2
   ```

The output of the following command shows the resulting configuration:

```
# hpvmnet -S vmlan4
  Name     Number State   Mode      PPA    MAC Address    IP Address
  ======== ====== ======= ========= ====== ============== ===============
  vmlan4        2 Up      Shared    lan4   0x00127942fce3 192.1.2.205
  [Port Configuration Details]
  Port    Port         Untagged Number of    Active VM
  Number  state        VLANID   Reserved VMs
  ======= ============ ======== ============ ============
  1       Active       100      2            vm1
  2       Active       100      1            vm2
  3       Active       none     2            vm1
  4       Active       none     1            vm2
```

The two VMs, `vm1` and `vm2`, have access to the virtual switch `vmlan4` and are active on VLAN 100. Specifically, port 1 (guest `vm1`) and port 2 (guest `vm2`) can communicate with each other. Port 1 (guest `vm1`) and port 4 (guest `vm2`) cannot communicate with each other.

The `hpvmnet` command displays the following information about the VLAN ports:

- Port number.
- State of the port. Table 18 (page 134) lists the possible VLAN port states.

Table 18 VLAN port states

State	Description
`Active`	The port is active and is allocated to a running guest. No other guests with the same vNIC with the same vswitch and port can start.
`Down`	The port is inactive and is allocated to a running guest. No other guests with the same vNIC with the same vswitch and port can start.
`Reserved`	At least one guest reserved the port for its vNIC, but no guest that uses the port is running.
`Available`	No guest reserved the port for its vNIC. When a VLAN is configured on the port, that port is displayed as Available. If no VLAN is configured, the port is not displayed.

- The untagged VLAN ID number (if any).
- The number of VMs that have access to the VLAN.
- The names of VMs that are up and that have access to the VLAN.

Cloning guests with VLAN information

If you use the `hpvmclone` command to clone guests, the operation automatically assigns new port numbers for new guests. To assign the same port number to the new guest, use the `–S` option, as follows:

```
# hpvmclone -P vm1 -N vmclone1 -S
```

This command creates a new guest (`vmclone1`) based on the existing guest `vm1`, and preserves the vswitch port number so that the new guest has access to the same VLANs as the existing guest.

Viewing VLAN information

You can view the vswitches and ports on a vswitch used by a guest using the `hpvmstatus` command. For example, to view the network information about the guest named `vm1`, enter the following command:

```
# hpvmstatus -P vm1

.
.
.
[Network Interface Details]
  Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
  ========= ========== ========== ======= === === === ==============
  vswitch   lan        localnet   1         0   1   0 de-19-57-23-74-bd
  vswitch   lan        localnet   2         0   2   0 7a-fb-4e-68-4f-5f
  vswitch   lan        vmlan4     1         0   4   0 6a-e8-c6-fa-b5-bc
  vswitch   lan        vmlan4     2         0   5   0 fa-18-82-9f-1a-95
  vswitch   lan        vmlan900   1         0   6   0 86-81-0b-6d-52-36
  vswitch   lan        vmlan900   2         0   7   0 6a-b9-cf-06-02-94
.
.
.
```

The preceding example shows the Network Interface Details portion of the output of the `hpvmstatus` command. In the list of network interfaces, note that each virtual network connection is associated with either port 1 or port 2 of several vswitches. The vswitch named `vmlan4` is associated with Bus/Dev/Ftn 0/4/0 on port 1 and with 0/5/0 on port 2.

To disconfigure a VLAN, use the following command:

```
# hpvmnet -S vswitch-name -u portid:portnum:vlanid:none
```

To view information about a specific VLAN port, include the `-p` option to the `hpvmnet` command. For example, to view VLAN information for port 2 on the vswitch named `vmlan4`, enter the following command:

```
# hpvmnet -S vmlan4 -p 2
  Vswitch Name            :  vmlan4
  Max Number of Ports     :  512
  Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlanId         :  none
```

To view the all the VLANs defined on the vswitch named `vmlan4`, enter the following command:

```
# hpvmnet -S vmlan4 -p all
  Vswitch Name            :  vmlan4
  Max Number of Ports     :  512
  Configured Ports        :  4
  Port Number             :  1
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  none
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlandID        :  none
Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlanID         :  none
Port Number             :  3
    Port State            :  Active
    Active VM             :  vm2
    Untagged VlanId       :  none
    Reserved VMs          :  vm2
    Adaptor               :  avio_lan
    Tagged VlanId         :  none
Port Number           :  4
    Port State            :  Active
    Active VM             :  vm2
    Untagged VlanId       :  100
    Reserved VMs          :  vm2
    Adaptor               :  avio_lan
    Tagged VlanID         :  none
```

Guest-based VLANs (AVIO)

To use guest-based VLANs, you must first enable the tagged VLAN IDs on the vswitch port. To enable the tagged VLAN IDs, use the `hpvmnet -S <vsw> -i` command. To disable the VLAN IDs, use the `hpvmnet` command with the `-o` option.

On a vswitch port, you cannot use a VLAN ID as both an untagged VLAN ID and a tagged VLAN ID at the same time. That is, a VLAN ID used with the `hpvmnet` command with the `-u` option cannot be used with the `hpvmnet -i` option.

Guest-based VLANs are supported with HP-UX 11i v3 guests only.

The following commands show the process to create guest based vlan.

To create untagged vlan id on port:

```
hpvmnet -S vmlan4 -u portid:8:vlanid:102
```

To create multiple tagged vlan id on port

```
hpvmnet -S vmlan4 -i portid:8:vlanid:103,104
```

```
# hpvmnet -S vmlan4 -p 8
  Vswitch Name            :  vmlan4
Max Number of Ports     :  512
Port Number             :  8
  Port State            :  Reserved
  Active VM             :
  Untagged VlanId       :  102
  Reserved VMs          :  vm4
  Adapter               :  avio_lan
  Tagged VLANs          :  103, 104
```

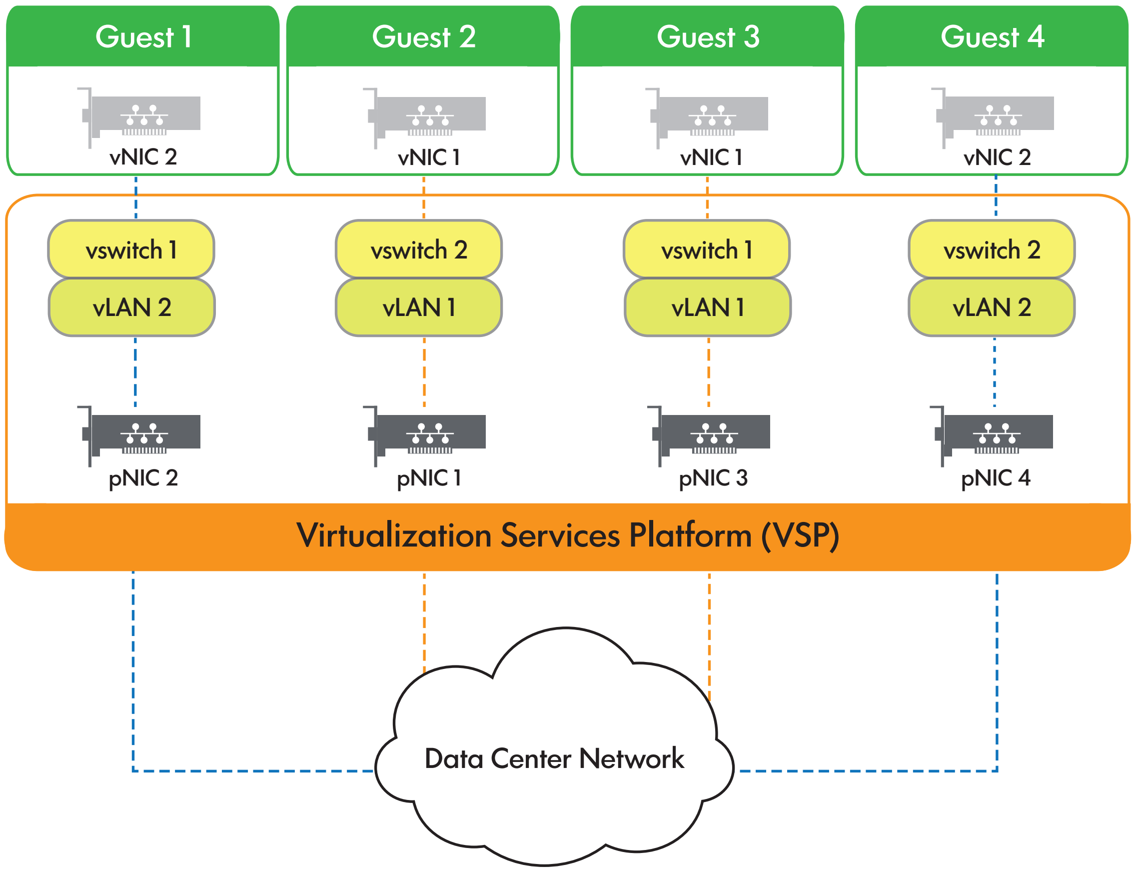
Configuring VLANs on virtual switches

The VLAN-backed vswitch (VBVsw) feature enables a virtual switch to be backed by a physical network device with HP-UX VLAN (IEEE 802.1Q) configured. The feature allows this type of vswitch to function such as a vswitch that is bound to a physical interface or an aggregate. Each VLAN backing the vswitch can be considered as a single network even though it is a discrete logical LAN managed by the VSP.

On the VSP, you can configure multiple VLAN interfaces on a guest LAN backed by VBVsw type vswitch. The network traffic delivered to and from the guest is filtered using the VLAN ID. Guest LANs backed to the vswitch that has VLAN configured share the same VLAN ID. Thus, these guest LANs can communicate with each other as if they were on the same physical network.

For more information about VLANs on HP-UX, see *HP-UX VLAN Administrator's Guide for HP-UX 11i v3* and *Planning and Implementing VLANs with HP-UX* manuals.

Figure 16 Integrity VM vswitch configuration example



Creating and managing a vswitch with a VLAN interface

To illustrate how to create and manage a vswitch with a VLAN interface, assume that your system has physical and aggregate interfaces as shown by the following format:

```
Name/          Interface Station        Sub-     Interface      Related
ClassInstance  State     Address        system   Type           Interface
============== ========= ============== ======== ============== =========
lan0           UP        0x0017A4AB5461 igelan   1000Base-T
lan1           UP        0x0017A4AB5460 igelan   1000Base-T
lan2           UP        0x001A4B06E90A iether   1000Base-T
lan3           UP        0x001A4B06E90B iether   1000Base-T     lan900
lan900         UP        0x001A4B06E90B hp_apa   hp_apa
lan901         DOWN      0x000000000000 hp_apa   hp_apa
lan902         DOWN      0x000000000000 hp_apa   hp_apa
lan903         DOWN      0x000000000000 hp_apa   hp_apa
lan904         DOWN      0x000000000000 hp_apa   hp_apa
```

To configure a PPA of the VLAN interface (VPPA) with a VLAN ID = 20 on the lan900 aggregate, enter the following:

```
# nwmgr -a -S vlan -A vlanid=20, ppa=900
VLAN interface lan5000 successfully configured.
lan5000 current values:
   VPPA = 5000
   Related PPA = 900
   VLAN ID = 20
   VLAN Name = UNNAMED
   Priority = 0
   Priority Override Level = CONF_PRI
   ToS = 0
   ToS Override Level = IP_HEADER

VLAN           Related    VLAN  Pri   Pri         ToS   Tos        Name
```

```
Interface     Interface  ID          Override        Override
Name                                 Level           Level
============= =========  ===== ==== ========== ==== ========== ================
lan5000       lan900     20    0    CONF_PRI   0    IP_HEADER  UNNAMED
```

To create, boot, and view a vswitch bound to VLAN lan5000, enter the following:

```
# hpvmnet -c -S vs5020 -n 5000
# hpvmnet -b -S vs5020
# hpvmnet -S vs5020
Name     Number State   Mode      NamePPA  MAC Address    IPv4 Address
======== ====== ======= ========= ======== ============== ===============
vs5020       18 Up      Shared    lan5000   0x001a4b06e90b

[Port Configuration Details]
Port    Port         Port     Untagged Number of    Active VM    Tagged
Number  State        Adaptor  VLANID   Reserved VMs              VLANIDs
======= ============ ======== ======== ============ ============ =============
1       Reserved     avio_lan none     2                         none
2       Reserved     avio_lan none     1                         none
3       Active       avio_lan none     1            u03          none
```

To enable the VBVsw feature, HP-UX PHNE_40215 or a superseding patch is required on the VSP. This patch is available as an individual patch or as part of "FEATURE11i" bundle. To verify whether the patch is installed, enter the following:

```
# swlist -l product |  grep LAN cumulative patch
PHNE_40215             1.0             LAN cumulative patch
```

The `dlpi_max_ub_promisc` kernel tunable must be set to 16 when using a VBVsw type vswitch. Otherwise, attempting to boot the vswitch fails with the following error message from the `hpvmnet` command:

```
# hpvmnet -b -S vs5000
hpvmnetd: setup_downlink: promisc failed, recv_ack:
promisc_phys: UNIX error - Device busy, errno 5
```

To set the kernel tunable, enter the following:

```
# kctune dlpi_max_ub_promisc=16
```

Configuring VLANs on physical switches

When communicating with a remote VSP or guest over the network, you might need to configure VLANs on the physical switches. The physical switch ports that are used must be configured specifically to allow the relevant VLANs. If the remote host is VLAN aware, you must configure VLAN interfaces on the host for the relevant VLANs. Use the `nwmgr` command to configure VLANs on a remote HP-UX host. For example, to configure a VLAN interface with VLAN ID 100 on `lan4`, enter the following command:

```
# nwmgr –a –S vlan –A vlanid=100,ppa=4
```

NOTE: When OLRAD suspend operation of card is initiated on a physical NIC backed to a vswitch, then this event on a physical NIC initiates link down on all the vNICs associated with that vswitch and resume initates link up on all the vNICs of that vswitch.

However, the deletion of physical NIC backed to a vswitch returns data critical warnings and must be exercised with caution. For more information about OLRAD, see “PCI OLR support on VSPs” (page 173).

Direct I/O networking

The direct I/O networking feature supported in vPars and Integrity VM Version 6 allows administrators to assign network ports directly to a vPar or VM guest, giving the vPar or VM guest direct and exclusive access to the port on the NIC. NIC ports that are configured to be used for direct I/O are not shareable and cannot be used to back a vswitch. Before a NIC port or card can be assigned to a vPar or VM guest, you must add it to a resource pool named DIO pool. DIO pool refers to a pool of direct I/O network capable devices that can be assigned to vPars or VMs.

NICs that support direct I/O networking on HPE Integrity BL8x0c i2/i4, Superdome 2 i2/i4, and rx2800 i2/i4 servers provide either FLA or DLA. The function in FLA refers to a single function on a multi-function NIC. A function can be single port on a multi-port card. Some cards support multiple functions on a single port. The device in DLA refers to the entire multi-port NIC (all functions of the NIC). If a card supports FLA, each function (port) can be individually added or removed from the DIO pool. FLA functions (ports) can be individually assigned to vPars or VM guests. Each FLA function of the same card can be used by different vPars or VM guests at the same time.

If a NIC supports only DLA, the entire card is added or removed from the DIO pool. You cannot assign a single port or function of a DLA card to the DIO pool. After a DLA card is added to the DIO pool, individual functions can be assigned to vPars or VM guests. To assign different functions of a DLA card to multiple vPars or VM guests, the vPar or VM guest cannot be configured to 'reserve' resources (`resources_reserved` setting). However, if multiple vPars or VM guests are assigned functions of the same DLA card (no reserved resources), only one VM can be booted at a time. For example,

- If you assign all four ports or functions of an FLA card to the DIO pool, you can assign port1 to vPar1, port2 to vPar2, and boot both vPar1 and vPar2 at the same time.
- If you assign a DLA NIC with four ports to the DIO pool, you can assign port1 to vm1 and port2 to vm2 only if `resources_reserved` is set to false. You can boot either vm1 or vm2.

The direct I/O networking functionality provides the following:

- 10 GB Ethernet network functions.
- Support for FlexNICs created by HPE Virtual Connect.
- Near-native network performance in vPar environments.
- Improved performance over AVIO networking in VM environments.
- CPU OL* operations with vPars.
- DLKM operations in the vPar or VM guest.
- Interrupt migrations in the vPar or VM guest and on the VSP.
- Running vPars or VM guests with DIO as Serviceguard nodes or Serviceguard packages.
- Support for HP-UX network providers.
- Support for direct I/O networking functionality with the HP APA product.

Using direct I/O networking

The following commands provide direct I/O networking for vPars and VM guests:

- The `hpvmhwmgmt` command allows you to:
  - List direct I/O capable functions on the VSP:

    ```
    # hpvmhwmgmt -p dio -l
    ```

NOTE: This command displays the assignment level.

– function: Function Level Assignment (FLA)

  Each function can be added or deleted individually to or from the DIO pool.
  Each function can be added or deleted individually to vPars or VM guests.
  Each function can be used individually by vPars or VM guests and the VSP.

– device: Device Level Assignment (DLA)

  The entire device is added or deleted to or from the DIO pool when one function of the device is specified.
  Each function can be added or deleted individually to vPars or VM guests.
  Only one vPar or VM at a time can use functions that are part of the same device.

◦ Add a function to the direct I/O pool:

  # **hpvmhwmgmt -p dio -a *hwpath* [-L label]**

NOTE: You cannot add a function if it is in use by the VSP or restricted for VSP use. Labels are optional and are used for offline migration.

◦ Delete a function from the direct I/O pool:

  # **hpvmhwmgmt -p dio -d *hwpath***

◦ Modify a label:

  # **hpvmhwmgmt -p dio -m *hwpath* -L label**

◦ Delete a label:

  # **hpvmhwmgmt -p dio -m *hwpath* -L none**

NOTE: The `hpvmdevmgmt —a, m, d` command blocks any attempt to add, modify, or delete the label attribute.

◦ List the factory MAC address of Direct I/O devices:

  # **hpvmhwmgmt -p dio -l -q**

NOTE: You can use the `hpvmhwmgmt` command with the `hpvmdevinfo` command to get the mapping between factory MAC and HPVM assigned MAC address of a DIO device assigned to a guest.

Example 27

```
# hpvmhwmgmt -p dio -l -q
H/W Path         Class   Owner  H/W Address
=============  ===== ====== ==============
7/0/0/2/0/0/0  lan      host   0X002655A976B6
7/0/0/2/0/0/1  lan      host   0X002655A976BA

# hpvmdevinfo -M | grep 7/0/0/2/0/0/0
myhost:myhost_G02:1:lan:dio:0;4;0xFEB453310183:hwpath:7/0/0/2/0/0/0:0/0/0/4/0 (lan1)
```

- The `hpvmmodify` command allows you to:

  ◦ Add a direct I/O function to a vPar or VM guest:

    # **hpvmmodify -P *vm* -a lan:dio:*[b,d,macaddr]*:hwpath:*hwpath***

NOTE: The function must already be in the direct I/O pool.

- Delete a direct I/O function from a vPar or VM guest:

  `# hpvmmodify -P vm -d lan:dio:[b,d,macaddr]:hwpath:hwpath`

- Replace a direct I/O function in a vPar or VM guest:

  `# hpvmmodify -P vpar -m lan:dio:b,d,macAddr:hwpath:new-hwpath`

- Modify the MAC address:

  `# hpvmmodify -P vpar -m lan:dio:b,d,new-macAddr:hwpath:hwpath`

- The `hpvmstatus` command allows you to:
  - View vPar and VM guest configurations. The direct I/O network functions are included in the #NETs count.

    `# hpvmstatus`

  - View specific vPar or VM I/O details:

    `# hpvmstatus -P vm -d`

    NOTE: There are no new switches specific to direct I/O in the `hpvmstatus` command output.

- The `hpvmstart` command allows you to:
  - Start a vPar or VM guest with direct I/O:

    `# hpvmstart -P vm`

    NOTE: Two vPars or two VM guests cannot start if they are using the same direct I/O function. Also, two vPars or two VM guests cannot start if they are using the same DLA device.

- The `hpvmstop` command allows you to:
  - Stop a vPar or VM guest that is using direct I/O:

    `# hpvmstop -P vpar`

    NOTE: There are no new switches specific to direct I/O for the `hpvmstart` or `hpvmstop` command.

To map direct I/O devices between the VSP and the vPars or VM guests:

- From the VSP:

  `# hpvmdevinfo -P vm`

- From the vPar or VM guest:

  `# hpvmdevinfo`

To restrict DIO-capable devices to the VSP, use the following command:

`hpvmdevmgmt -a rdev:hwpath`

NOTE: If the *hwpath* is for a DLA function, all functions will be added.

The *hwpath* must be assigned to the VSP to restrict for VSP use. If the *hwpath* is already in use by a vPar or VM guest, the `-a` add option fails.

Hewlett Packard Enterprise recommends that administrators manually restrict all functions that are used by the VSP for VSP networking, because that is currently not done automatically. Hewlett Packard Enterprise also recommends that administrators manually restrict all functions that are assigned to vPars and VM guests for use with AVIO, to avoid conflicts at vPar or VM guest boot time, because those functions will not appear to be in use until the vPars and VM guests are booted.

Use `hpvmhwmgmt` or `vparhwmgmt` command to view the DIO supported cards on a VSP and also the assignment level the NICs support (device or function):

```
# vparhwmgmt -l -p dio
                                                                 Assignment
H/W Path        Class    Owner Description                       Level      Label
--------------  -------  ----- -------------------------         ---------- -------------
0/0/0/3/0/0/0   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/1   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/2   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/3   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/4   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/5   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/6   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/3/0/0/7   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/0   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/1   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/2   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/3   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/4   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/5   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/6   lan      host  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/7   lan      host  HP  PCIe 2-p 10GbE Built-         device
```

Use the `hpvmhwmgmt -p dio -a` *`path`* command to assign the card or function to the DIO pool. For DLA cards, you can use the path of any port on the card. All functions of the card are assigned to the DIO pool. After the function or device is added to the DIO pool, the `hpvmhwmgmt` command shows the owner as `hpvm` and not `host`.

NOTE: If you use the `-L label` option when adding a DLA card to the DIO pool, only the function (path) that was specified in the command line will be labeled, other ports of the DLA card must be labeled individually.

```
# hpvmhwmgmt -p dio -a 0/0/0/4/0/0/1 -L DLA1
# hpvmhwmgmt -l -p dio
                                                                 Assignment
H/W Path        Class    Owner Description                       Level      Label
--------------  -------  ----- -------------------------         ---------- -------------
0/0/0/3/0/0/0   lan      host  HP  PCIe 2-p 10GbE Built-         device
…
0/0/0/4/0/0/0   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/1   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device     DLA1
0/0/0/4/0/0/2   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/3   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/4   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/5   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/6   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device
0/0/0/4/0/0/7   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device


# hpvmhwmgmt -p dio -m 0/0/0/4/0/0/7 -L DLA1.1
# hpvmhwmgmt -p dio -l | grep DLA1
0/0/0/4/0/0/1   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device     DLA1
0/0/0/4/0/0/7   lan      hpvm  HP  PCIe 2-p 10GbE Built-         device     DLA1.1
```

When a DIO device is added to the DIO pool, `ioscan` shows the device is claimed by the `hpvmdio` device:

```
# ioscan -funC hpvmdio

Class     I  H/W Path       Driver    S/W State H/W Type   Description
======= === ============= ========= ========= =========  ================
```

```
hpvmdio  0  0/0/0/4/0/0/0  hpvmdio  CLAIMED   INTERFACE  HP PCIe 2-p 10GbE Built-in FLEX-10
                           /dev/hpvmdio0
….
```

You cannot add a function to the pool if it is in use by the VSP:

```
# hpvmnet
Name                  Number State   Mode      NamePPA MAC Address    IPv4 Address
===================== ====== ======= ========= ======= ============== ===============
localnet                   1 Up      Shared            N/A            N/A
hpnet                      2 Up      Shared    lan0    0x1cc1de40d040 10.43.212.199
priv_net                   3 Up      Shared    lan1    0x1cc1de40d044

# hpvmhwmgmt -l -p dio | grep 0/0/0/3/0/0/7
0/0/0/3/0/0/7  lan     host  HP  PCIe 2-p 10GbE Built- device

# hpvmhwmgmt -p dio -a 0/0/0/3/0/0/7
hpvmhwmgmt: Sibling path '0/0/0/3/0/0/0' (lan0) is being used as vswitch 'hpnet'.
hpvmhwmgmt: Sibling path '0/0/0/3/0/0/1' (lan1) is being used as vswitch 'priv_net'.
hpvmhwmgmt: Lan devices used as vswitches cannot be added to the DIO pool.
hpvmhwmgmt: Unable to manage dio pool resource.
```

Use the `vparstatus -A` command to view the functions available in the DIO pool:

```
# vparstatus -A | grep dio
        lan:dio::hwpath:0/0/0/4/0/0/0
        lan:dio::hwpath:0/0/0/4/0/0/1
        lan:dio::hwpath:0/0/0/4/0/0/2
        lan:dio::hwpath:0/0/0/4/0/0/3
        lan:dio::hwpath:0/0/0/4/0/0/4
        lan:dio::hwpath:0/0/0/4/0/0/5
        lan:dio::hwpath:0/0/0/4/0/0/6
        lan:dio::hwpath:0/0/0/4/0/0/7
```

Use the `hpvmmodify` command or `vparmodify` command to add the DIO device to an existing guest:

```
# vparmodify -p vpar1 -a lan:dio::hwpath:0/0/0/4/0/0/0
```

If you attempt to add a function of a DLA device when another vPar or VM guest is assigned a function on that same DLA device **and** has `resources_reserved` set to true, the add fails:

```
# vparmodify -p vpar2 -a lan:dio::hwpath:0/0/0/4/0/0/1
vPar/VM vpar2 configuration problems:
    Error 1: The sibling DLA function: '0/0/0/4/0/0/0' of function: '0/0/0/4/0/0/1'
             is in use by another guest. vparmodify: Unable to modify the vPar.
```

Setting the `resources_reserved` flag on the vPar or VM guest to false allows you to add the function to the vPar or VM guest:

```
# vparmodify -p vpar1 -x resources_reserved=false
# vparmodify -p vpar2 -a lan:dio::hwpath:0/0/0/4/0/0/1

# vparstatus -v -p vpar1 | grep dio
        lan:dio:0,6,0x7e06f5393261:hwpath:0/0/0/4/0/0/0
# vparstatus -v -p vpar2 | grep dio
        lan:dio:0,4,0xca7e0c0d0e96:hwpath:0/0/0/4/0/0/1
```

However, only one of these vPars boots at one time:

```
# vparboot -p vpar1
(C) Copyright 2000 - 2012 Hewlett-Packard Development Company, L.P.
UsrDirectAdd: hw_path="0/0/0/4/0/0/0"  MAC=0x7e06f5393261.
…

# vparstatus
[Virtual Partition]
Num Name     RunState      State
=== ======= ============  =========
  2 vpar1    EFI           Active
  1 vpar2    DOWN          Inactive

…
```

```
# vparboot -p vpar2
vPar/VM vpar2 configuration problems:
Error 1: The sibling DLA function: '0/0/0/4/0/0/0' of function: '0/0/0/4/0/0/1'
is in use by another guest. vparboot: Unable to continue.
```

NOTE: Trunking software such as APA is supported on DIO interfaces in the guest. For more information about APA, see *Auto Port Aggregation (APA) Support Guide*.

For the syntax and complete list of options for these commands, see the appropriate manpages.

Troubleshooting AVIO and DIO network problems

For more information about troubleshooting AVIO and DIO network problems, see "Networking" (page 295).

9 Administering VMs

After installing the vPars and Integrity VM product, you can create VMs and virtual resources for the VMs to use.

NOTE: The Integrity VM commands can be used to configure and manage both vPars and VM. They support overall product features. Hewlett Packard Enterprise recommends using Integrity VM commands over vPar commands for managing vPars or VM.

Taking backups of guest configurations

Some commands or GUI actions modify the configuration of guests; the following is a partial list of such commands:

- `hpvmclone(1M)`
- `hpvmcreate(1M)`
- `hpvmdevmgmt(1M)`
- `hpvmhostgdev(1M)`
- `hpvmhostrdev(1M)`
- `hpvmmodify(1M)`
- `hpvmmove_suspend(1M)`
- `hpvmnet(1M)`
- `hpvmnvram(1M)`
- `hpvmremove(1M)`
- `hpvmhwmgmt(1M)`
- `vparcreate3(1M)`
- `vparhwmgmt3(1M)`
- `vparmodify3(1M)`
- `vparnet3(1M)`
- `vparremove3(1M)`

In addition, operations within the guest, such as modification of boot-paths (from EFI shell or HP-UX operating system) or modification of EFI variables (from EFI shell) will modify the file on VSP used to emulate nvram for the vPar or Integrity VM guest. It is advisable to backup the content of `/var/opt/hpvm/` on the VSP, before and after significant configuration changes.

Specifying VM attributes

When you create a new VM, you specify its attributes. Later, you can change the VM attributes.

You can set the attributes of a VM using the following commands:

- `hpvmcreate`, which creates a new VM.
- `hpvmclone`, which creates a new VM based on an existing VM.
- `hpvmmigrate`, which moves a VM from one system to another.
- `hpvmmodify`, which modifies an existing VM.

All these commands accept the same options for specifying VM attributes. Table 19 (page 146) lists each attribute and command option.

Table 19 Attributes of a VM

VM attributes	Description	Command option	Default value
VM name	You must specify a name when you create or modify a VM. You cannot modify this attribute.	`-P` *vm-name*	The VM name can have up to 255 alphanumeric characters, including A-Z, a-z, 0–9, the dash (—), the underscore (_), and period (.). The VM name must not start with a dash.
Operating system type	Specify the guest operating system type. For more information about guest operating system type, see “Specifying guest operating system type” (page 246).	`-O` *os_type [:version]*	If you do not specify the operating system type, it is set to UNKNOWN.
Virtual CPUs (vCPUs)	You can specify the number of CPUs that a VM can use.	`-c` *number_vcpus*	If you do not specify this attribute when you create the VM, the default is one vCPU.
VM type	Specify the type of guest. For more information about VM type, see “Specifying VM type” (page 242).	`-x` vm_type=*type*	If not specified, by default a shared VM is created.
CPU entitlement	The minimum amount of processing power guaranteed to the VM.	`-e` *percent[:max_percent]*`-E` *cycles[:max_cycles]*	If you do not specify this attribute when you create the VM, the default is 10%.
Memory	Total amount of memory allocated to the VM.	`-r` *amount*	If you do not specify this attribute when you create the VM, the default is 2 GB.
Virtual devices	You can allocate virtual network switches and virtual storage devices to the VM. The VSP presents devices to the VM as virtual devices. The VM network consists of vNICs and vswitches. For VMs to communicate either with other VMs or outside the VSP system, each virtual network of the VM must be associated with a vswitch. If you start a VM without a vswitch, the VM does not have a network communication channel. VM also supports DIO networking where physical devices are directly presented. DIO devices do not require a vswitch. Virtual storage devices are backed by physical devices on the VSP system. You can specify one of the following devices – disk, dvd, tape, changer, burner, or hba. For more information about virtual devices, see “Virtual devices” (page 148).	`-a` *rsrc*	If you do not specify this attribute when you create the VM, it will not have access to network and storage devices.

Table 19 Attributes of a VM *(continued)*

VM attributes	Description	Command option	Default value
VM label	A short description of VM. For more information about VM label, see "Creating VM labels" (page 246).	`-l` *vm_label*	If you do not specify this attribute, the VM will not have a label.
Startup behavior	Sets the start attribute of the VM. For more information about VM boot attributes, see "Specifying the VM boot attribute" (page 246)	`-B` *start_attribute*	If you do not specify this attribute, it is set to `auto`, and the VM starts when Integrity VM is started.
Dynamic memory	Specify whether the VM uses dynamic memory and its associated values. For more information about dynamic memory attributes, see "Specifying dynamic memory parameters" (page 149)	`-x` *keyword=parameter*	If you do not specify this attribute, dynamic memory is not enabled for the guest.
Group with administrator or operator privileges	Specify group accounts that will have administrator or operator privileges to the VM. For more information guest administrator and operator privileges, see "Creating guest administrators and operators" (page 247).	`-g [+]`*group*`[:`*admin*`|`*oper*`]`	If you do not specify this attribute, group accounts cannot have `admin` or `oper` privileges.
Resource reservations	Enable or disable resource reservation. For more information about resource reservation, see "Reserved resources and resource over-commitment" (page 54).	`-x` resources_reserved= [true | false]	If not specified, resources will not be reserved when the VM is off.
User with administrator or operator privileges	Specify user accounts that will have administrator or operator privileges to the VM. For more information about administrator and operator, see "Creating guest administrators and operators" (page 247).	`-u [+]`*user*`[:`*admin*`|`*oper*`]`	If you do not specify this attribute, user accounts cannot have `admin` or `oper` privileges.

VM name

Use the `-P` *`vm-name`* option to specify the name of the new VM. This option is required for the `hpvmcreate` command. In the following example, the new VM is named `host1`. On the VSP, enter the following command:

```
# hpvmcreate -P host1
```

The VM name can include up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and period (.). The VM name must not start with a dash.

Reserved resources

Use the `-x resources_reserved={true, false}` option to specify whether CPU, memory, and device resources must be reserved while the VM is in the off state.

```
# hpvmcreate -P host1 -x resources_reserved=true
```

Resource reservations attempt to guarantee that resources will be available so that the VM can be started at any time. For more information about reserved resources, see Section (page 54).

Virtual CPUs

The following command specifies the number of virtual CPUs to allocate:

```
# hpvmcreate -c number_vcpus[:minimum[:maximum]]
```

If you do not specify the number of vCPUs, the default is 1. For example, to set the new VM `host1` to have two vCPUs, enter the following:

```
# hpvmcreate -P host1 -c 2
```

The default minimum and maximum boundary values are a minimum of one (1) virtual CPU and a maximum of 32 virtual CPUs.

To set the new VM to have minimum of 1 vCPU and maximum of 4 vCPU boundary values, and two (2) virtual CPUs, run the following command:

```
# hpvmcreate -P host1 -c 2:1:4
```

CPU entitlement

Use the `-e` or `-E` option to specify the CPU entitlement of the VM.

```
# hpvmcreate -P <vm-name> -e percent[:max_percent]
# hpvmcreate -P <vm-name> -E cycles[:max_cycles]
```

When you create a VM, you can use the `-e` option to specify the entitlement as a percentage, from 5% to 100%. If you do not specify the entitlement, the VM receives 10% entitlement by default. The maximum entitlement is 100% by default.

For example, to specify an entitlement of 20% for the new VM `host1`, enter the following command:

```
# hpvmcreate -P host1 -e 20
```

Alternatively, you can use the `-E` option to specify the entitlement as the number of CPU clock cycles per second to be guaranteed to each vCPU on the VM.

For more information about VM entitlement, see Section (page 49).

Guest memory allocation

Use the `-r` *`amount`* option to specify the amount of virtual memory to be allocated to the guest. If you do not specify the memory allocation, the default is 2 GB. For example, to allocate 3 GB to the VM `host1`, enter the following command:

```
# hpvmcreate -P host1 -r 3G
```

Virtual devices

Use the `-a` option to allocate virtual network interfaces and virtual storage devices to the VM. The VSP presents devices to the VM as “virtual devices.” Attached I/O devices, such as tape, DVD burner, and autochanger are not presented as virtual devices. They are presented as physical I/O devices. You specify both, the physical device to allocate to the VM and the virtual device name that the VM uses to access the device. The following examples provide brief instructions for creating virtual network devices and virtual storage devices.

Example 28 Create a VM with virtual network interface backed by virtual switch

Create a VM named Oslo in the local system specifying 2 GB of memory, 2 CPUs, and virtual network interface backed by virtual switch “sitelan”.

```
# hpvmcreate -P Oslo -r 2048 -c 2 -a network:avio_lan::vswitch:sitelan
```

For more information about creating and managing virtual switches, see “Creating virtual and direct I/O networks” (page 122).

Example 29 Create a VM with virtual disk backed by a whole disk

Create a VM named Oslo in the local system specifying 2 GB of memory, 2 CPUs, and virtual disk backed by a whole disk “/dev/rdisk/disk70”.

```
# hpvmcreate -P Oslo -r 2048 -c 2 -a
disk:avio_stor::disk:/dev/rdisk/disk70
```

For more information about different backing store devices, see “Storage devices” (page 60).

Example 30 Create a VM with vHBA

Create a VM named vm001 with a virtualized HBA using NPIV port assuming a GUID manager is available to assign World Wide Port Name and World Wide Node Name.

```
# hpvmcreate -P vm001 -a hba:avio_stor::npiv:/dev/fcd0
```

For more information about configuring NPIV, see the *hpvmresources*(5) and “NPIV with vPars and Integrity VM” (page 99).

Example 31 Create a VM with network interface backed by a DIO function

Add the DIO function “0/0/0/4/0/0/0” to the direct I/O pool using the `hpvmhwmgmt` command:

```
# hpvmhwmgmt -p dio -a 0/0/0/4/0/0/0
```

Create a VM named Oslo in the local system specifying memory of 2 GB, 2 CPUs, and virtual network interface backed by a DIO function “0/0/0/4/0/0/0”

```
# hpvmcreate -P Oslo -r 2048 -c 2 -a lan:dio::hwpath:0/0/0/4/0/0/0
```

For more information about configuring VM guests with DIO functions, see “Direct I/O networking” (page 138).

Specifying dynamic memory parameters

Specifies whether the new VM (shared VM type only) uses dynamic memory and the values associated with it by including the following keywords:

- `dynamic_memory_control={0|1}`
- `ram_dyn_type={none|any|driver}`
- `ram_dyn_min=`*amount*
- `ram_dyn_max=`*amount*
- `ram_dyn_target_start=`*amount*
- `ram_dyn_entitlement=`*amount*
- `amr_enable={0|1}`
- `amr_chunk_size=`*amount*

For more information about using dynamic memory for guests, see “Managing dynamic memory from the guest” (page 257).

Configuration limits

Table 20 (page 150) lists the configuration limits for Integrity VM v6.4. For NPIV supported limits, see Table 14 (page 100).

Table 20 Integrity VM v6.4 configuration limits

Configuration item	Support limit
# vCPUs/VM — Maximum (HP-UX 11i v2)	16
# vCPUs/VM — Maximum (HP-UX 11i v3)	32
# vCPUs/pCPU — Maximum	20
# VMs per VSP — Maximum	254
# pCPUs in VSP	HP-UX limit
Memory per VM — Minimum (HP-UX 11i v2)	1 GB
Memory per VM — Minimum (HP-UX 11i v3)	2 GB or the minimum required for HP-UX 11i v3 to boot
Memory per VM — Maximum (HP-UX 11i v2)	128 GB
Memory per VM — Maximum (HP-UX 11i v3)	256 GB
# virtual AVIO storage devices / VM or vPar— Maximum	256 AVIO
# virtual NICs / VM or vPar— Maximum	62
# virtual switches — Maximum	50
# virtual NICs / vswitch	511
# file backing store devices / VM or vPar — Maximum	30
Maximum size of backing store for AVIO (disk, lvol, file)	HP-UX limit
Maximum # PCI functions per vPar/VM for DIO	16

Sizing guidelines

The sizing guidelines for Integrity VMs Version 4.0 and later are different from that of earlier releases due to several factors, including the change of VSP operating system to HP-UX 11i v3. The formulas used to calculate VM capacity are outlined in the white paper hardware consolidation with integrity virtual machines. The sizing information and related calculations are updated in revisions to this white paper dated September 2008 or later. The latest version of this white paper is available at: **http://hpe.com/info/virtualization-manuals**.

Default guest settings for HP-UX

Table 21 (page 150) lists the default guest settings for HP-UX and `Unknown` guests. An `Unknown` guest is a VM that has not booted with any operating system. When an `Unknown` guest type boots, the appropriate operating system type is applied to the guest configuration.

The following guest OS specific settings are applied if you specify the `-o` option for the operating system type in the `hpvmcreate` command.

Table 21 Guest default settings

Attribute	HP-UX guest default setting	Unknown guest operating system default setting
Maximum CPUs	32	32
Default CPUs	1	1
Default memory	2 GB	2 GB
Minimum memory	512 MB[1]	32 MB
Maximum memory	256 GB	256 GB

Table 21 Guest default settings *(continued)*

Attribute	HP-UX guest default setting	Unknown guest operating system default setting
Default reserved memory	64 MB	64 MB
Minimum reserved memory	32 MB	32 MB
Maximum reserved memory	256 GB	256 GB

[1] The minimum memory requirement for HP-UX 11i v2 is 512 MB. The minimum memory requirement for HP-UX 11i v3 is 1 GB (see "System Requirements" section in the *HP-UX 11i v3 Installation and Update Guide*); however, the HP-UX 11i v3 installation and update guide warns that cold installations with 1 GB or less memory might fail or take a long time to complete. Therefore, Hewlett Packard Enterprise recommends 2 GB for cold installations of HP-UX 11i v3.

NOTE: The amount of memory you must allocate to the guest must be sufficient to allow the guest operating system to boot. This amount might differ from the defaults documented here. For specific memory requirements, see the documentation for the operating system and applications on the guest.

Using the `hpvmcreate` command

To create a VM, run the `hpvmcreate` command. Enter the `-P` option to specify the VM name (up to 255 alphanumeric characters). All other options are optional and might be added to the VM configuration later using the `hpvmmodify` command.

Table 22 (page 151) lists the options that can be used with the `hpvmcreate` command.

Table 22 Options to the `hpvmcreate` command

Option	Description
`-P` *`vm-name`*	VM name. You must specify a name when you create or modify the VM. You cannot modify this characteristic.
`-O` *`os_type[:version]`*	Specifies the type and version of the operating system. If you do not specify the operating system type, it is set to UNKNOWN. The version is specific to the operating system type and can consist of up to 255 alphanumeric characters, including A-Z, a-z, 0–9, the dash (—), the underscore (_), and the period (.).
`-c` *`number_vcpus[:min[:max]]`*	Virtual CPUs (vCPUs) allocated. If you do not specify this attribute when you create the VM, the default is one vCPU.
`-e` *`percent[:max_percent]`*`-E` *`cycles[:max_cycles]`*	CPU entitlement allocated. If you do not specify this attribute when you create the VM, the default entitlement is 10% and the max_percent is 100%.
`-r` *`amount`*	Memory allocated. If you do not specify this attribute when you create the VM, the default is 2 GB.
`-a` *`rsrc`*	Virtual devices created. If you do not specify this attribute when you create the VM, the VM will not have access to network and storage devices.
`-l` *`vm_label`*	The label (an optional text string associated with the VM) for the VM.
`-B` *`start_attribute`*	The startup behavior of the VM (`auto` or `manual`).
`-x` *`keyword=parameter`*	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max`

Table 22 Options to the `hpvmcreate` command *(continued)*

Option	Description
	• `ram_dyn_target_start` • `ram_dyn_entitlement=`*`amount`* • `amr_enable={0|1}` • `amr_chunk_size=`*`amount`* • `sched_preference` • `graceful_stop_timeout` For more information about dynamic memory, see “Managing dynamic memory from the guest” (page 257). Also specifies values for OVMM: • migrate_copy_phase_timeout={number of seconds} • migrate_frozen_phase_timeout={number of seconds} • migrate_init_phase_timeout={number of seconds} • migrate_io_quiesce_phase_timeout={number of seconds} • online_migration={enabled	disabled} • tunables={name=value[,name=value,...]} For information about OVMM, see “Migrating VMs and vPars” (page 203). Specifies arbitrary VM or vPar attributes that control their behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}`
`-F`	Suppresses all resource conflict checks and associated warning messages (force mode). This option is primarily intended for use by scripts and other non-interactive applications. Note that you will not receive notification about any potential resource problems for a VM created with the `-F` option. **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.
`-s`	Verifies the VM configuration and returns warnings or errors, but does not create the VM. This option is used to initiate resource verification of the `hpvmcreate` command for a VM configuration without actually creating the VM. If the `-s` option is not specified, the VM is created even if resource warnings occur.
`-g` *`group`*`[:`*`admin`* `|` *`oper`*`]`	Group with administrator or operator privileges over the VM. Enter the group name for *`group`*, and enter either `admin` or `oper`.
`-u` *`user`*`[:`*`admin`* `|` *`oper`*`]`	User with administrator or operator privileges over the VM. Enter the user name for *`user`*, and enter either `admin` or `oper`.
`-i`*`package-name`*	Specifies whether the VM is managed by Serviceguard or gWLM (or both). For the argument, specify one or more of the following parameters: • `SG` indicates that the VSP is a Serviceguard cluster node. • `SG-`*`pkgname`* indicates that the VSP is a Serviceguard package. • `GWLM` indicates that the VSP is managed by gWLM. • `NONE` indicates there are no external managers. For a node that is managed by both Serviceguard and gWLM, parameters are separated with a comma. For example: `SG_host1,gWLM`. **CAUTION:** Use this option only if instructed by Hewlett Packard Enterprise.

Table 22 Options to the hpvmcreate command *(continued)*

Option	Description
-j {0 | 1}	Specifies whether the VM is a distributed guest (that is, managed by Serviceguard and can be failed over to another cluster member).
-K *console_IP_Addr*	Specifies the IP address used to connect to the virtual iLO Remote Console of the guest. The address must be specified in the IPv4 dot notation. The –L option must also be specified.
-L *console_IP_Addr_Netmask*	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.

Example of VM creation

To create a VM named guest1, enter the following command:

```
# hpvmcreate -P guest1 -c 4 -r 10G
```

This command creates a VM named guest1 that does not have network access and allocated storage devices. To view the characteristics of the VM, enter the hpvmstatus command. For example,

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vPar0002                 2 VP   HPUX    Off            3     0     0 2048 MB
guest1                   3 SH   UNKNOWN Off            4     0     0   10 GB
ux1                      1 SH   HPUX    Off            4     2     0    3 GB
```

The guest1 VM is assigned VM number 3, is created with an UNKNOWN operating system type, four vCPUs, zero storage devices, zero network devices, and 10 GB of memory. For more information about running VMs under Serviceguard, see *Serviceguard Toolkit for Integrity Virtual Servers User Guide* at **http://www.hpe.com/info/hpux-serviceguard-docs**.

Starting VMs

To start the VM, run the hpvmstart command. You can specify either the VM name or the VM number (listed in the hpvmstatus display under VM #).

The hpvmstart command syntax is:

```
# hpvmstart {-P vm-name | -p vm_number} [-F | -s | -Q]
```

Table 23 (page 153) lists the options that can be used with the hpvmstart command.

Table 23 Options to the hpvmstart command

Option	Description
-P *vm-name*	Specifies the name of the VM. Specify either the -P option or the -p option.
-p *vm_number*	Specifies the number of the VM. To determine the VM number, enter the hpvmstatus command.
-F	Suppresses all resource conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The -F option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support OR explicitly stated in the Administrator Guide.

Table 23 Options to the `hpvmstart` **command** *(continued)*

Option	Description
`-s`	Sanity-checks the VM configuration and returns warnings or errors, but does not start the VM.
`-Q`	Quietly executes the command. The default is to prompt for confirmation of the command before performing it.

For example, to start the new VM `host1`, enter the following command:

```
# hpvmstart -P host1
(C) Copyright 2000 - 2015 Hewlett-Packard Development Company, L.P.
Mapping vPar/VM memory: 2048MB
   mapping low RAM (0-80000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/9e69613e-dba8-11e1-b802-
00237d4506f4/vmm_config.next):
 Allocated 2147483648 bytes at 0x6000000100000000
    locking memory: 0-80000000
  allocating overhead RAM (6000000180000000-600000018c000000, 192MB)
    locking memory: 6000000180000000-600000018c000000
  allocating datalogger memory: FF800000-FF900000 (1024KB)
  allocating firmware RAM (fff00000-100000000, 1024KB)
    locked SAL RAM: 00000000fff00000 (8KB)
    locked ESI RAM: 00000000fff02000 (8KB)
    locked PAL RAM: 00000000fff04000 (8KB)
    locked Min Save State: 00000000fff0a000 (4KB)
    locked datalogger: 00000000ff800000 (1024KB)
Creation of VM minor device 1
Device file = /var/opt/hpvm/uuids/9e69613e-dba8-11e1-b802-00237d4506f4/vm_dev
Loading boot image
Image initial IP=102000 GP=69E000
Starting event polling thread
guestStatsStartThread: Started guestStatsCollectLoop - thread = 6
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of vPar or VM 'host1'
```

The `hpvmstatus` command displays the allocation of memory and devices. After you start the VM, the `hpvmstatus` command displays the VM status as `On (EFI)`, because the VM is powered on but the guest operating system is not running. Because the operating system has not been installed, the guest OS type is listed as `UNKNOWN`.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type    OS Type   State  #VCPUs #Devs #Nets  Memory
==================== ===== ======= ========= ====== ====== ===== =====  =======
config1                  1 SH      HPUX      Off       1     5     1     512 MB
config2                  2 SH      HPUX      Off       1     7     1       1 GB
guest1                   5 SH      HPUX      On (OS)   1     5     1       1 GB
host1                   13 SH      UNKNOWN   On (EFI)  1     0     0       2 GB
```

For more information about using the `hpvmstatus` command, see "Managing vPars and VMs using CLI" (page 239).

NOTE: When configuring or starting Integrity VM guests, the following warning message might be displayed if storage associated with the guest appears to be performing very poorly.

```
hpvmcreate: WARNING (host): Device /dev/rdisk/disk17 took 32 seconds to open.
```

Changing VM configurations

You can create a VM with characteristics that the VSP cannot supply at the time of creation. This allows you to create VMs to run, after system configuration changes. For example, the following command creates the VM `host1` with 3 vCPUs and 4 GB of allocated memory:

```
# hpvmcreate -P host1 -c 3 -r 4G
HPVM guest host1 configuration problems:
    Warning 1: Guest's vcpus exceeds server's physical cpus.
```

```
    Warning 2: Insufficient cpu resource for guest.
These problems may prevent HPVM guest host1 from starting.
hpvmcreate: The creation process is continuing.
```

Because the VSP is currently not configured to support the new VM, warning messages indicate the specific characteristics that are inadequate.

When you start a VM, the VSP determines whether the current system configuration can support the characteristics of the VM. The ability of the system to run the VM can be affected by the other VMs that are currently running, because the VMs share the physical processors and memory. Any allocated vswitches must be started, and storage devices must be made available to the VM. If the VM cannot be started, the following type of message is generated:

```
# hpvmstart -P host1
HPVM guest host1 configuration problems:
Warning 1: Insufficient free memory for guest.
Warning 2: Insufficient cpu resource for guest.
   These problems may prevent HPVM guest host1 from booting.
hpvmstart: Unable to continue.
```

You can either change the system configuration, or modify the VM. To modify the characteristics of a VM, use the `hpvmmodify` command. When you use the `hpvmmodify` command to modify a guest, the entire guest configuration is re-evaluated. Any problems that might prevent the guest from starting are reported. For example, if a guest has a reference to a host device that no longer exists, and you enter an `hpvmmodify` command that modifies the guest but does not fix the bad reference, a warning message is displayed. Table 24 (page 155) lists the options that can be used with the `hpvmmodify` command.

For example, to modify the characteristics of the problematic VM `host1` to remove vCPUs and memory, enter the following command:

```
# hpvmmodify -P host1 -c 1 -r 2G
```

This command changes the following characteristics of the VM named `host1`:

- The `-c 1` option specifies one vCPU.
- The `-r 2G` option specifies two GB of memory.

Table 24 Options to the `hpvmmodify` command

Option	Description
`-P` *`vm-name`*	Specifies the name of the VM. You must specify either the `-P` option or the `-p` option.
`-p` *`vm_number`*	Specifies the number of the VM. To determine the VM number, enter the `hpvmstatus` command.
`-F`	Suppresses all resource conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.
`-s`	Sanity-checks the VM configuration and returns warnings or errors, but does not start the VM.
`-N` *`new-vm-name`*	Specifies a new name for the VM. The name can consists of up to 255 alphanumeric characters including A-Z, a-z, 0-9, the dash (-), the underscore character (_), and the period (.). The VM name cannot start with a dash (—).
`-l` *`vm_label`*	Modifies the descriptive label for this VM. The label can contain up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). To include spaces, the label must be quoted (" ").

Table 24 Options to the `hpvmmodify` command *(continued)*

Option	Description
`-B` *`start_attr`*	Modifies the startup behavior of the VM. For *`start_attr`*, enter one of the following: `auto`: Automatically starts the VM when Integrity VM is initialized on the VSP. `manual`: The VM is not started automatically. Use the `hpvmstart` command to start the VM manually.
`-O` *`os_type`*`[:`*`version`*`]`	Modifies the type and version of the operating system running on the VM. For the *`os_type`*, specify the following (case-insensitive) value: `hpux`
`-c` *`number_vcpus[:min[:max]]`*	Modifies the number of virtual CPUs this VM detects at boot time. If unspecified, the number defaults to one. The maximum number of vCPUs that you can allocate to a VM is the number of physical processors on the VSP system.

`-e` *`percent[:max_percent]`* `|`
`-E` *`cycles[:max_cycles]`*
	Modifies the CPU entitlement of the VM in CPU cycles. To specify the percentage of CPU power, enter the following option:
	`-e` *`percent[:max_percent]`*
	To specify the clock cycles, enter one of the following options:
	`-E` *`cycles[:max_cycles]`*`M (for megahertz)`
	`-E` *`cycles[:max_cycles]`*`G (for gigahertz)`

`-g` *`group`*`[:`*`admin`* `|` *`oper`*`]`	Specifies a group authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-K` *`console_IP_Addr`*	Specifies the IP address used to connect to the virtual iLO Remote Console of the guest. The address must be specified in IPv4 dot notation or 0. If 0 is entered, then the guest will no longer have virtual iLO Remote Console access using IP.
`-L` *`console_IP_Addr_Netmask`*	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.
`-u` *`user`*`[:`*`admin`* `|` *`oper`*`]`	Specifies a user authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user.
`-a` *`rsrc`*	Adds a virtual storage or network device to the VM. For more information, see *hpvmresources*(5).
`-m` *`rsrc`*	Modifies an existing I/O resource for a VM. The resource is specified as described. You must specify the hardware address of the device to modify. The physical device portion of the `rsrc` specifies a new physical device that replaces the one in use.
`-d` *`rsrc`*	Deletes a virtual resource.
`-r` *`amount`*	Modifies the amount of memory available to this VM. Specify the amount as either *`amount`*`M` (for megabtyes) or *`amount`*`G` (for gigabytes).
`-i` *`package-name`*	Specifies whether the VM is managed by Serviceguard or gWLM (or both). For the argument, specify one or more of the following parameters: • `SG` indicates that the VM is a Serviceguard cluster node. • `SG-`*`pkgname`* indicates that the VM is a Serviceguard package. • `GWLM` indicates that the VM is managed by gWLM. • `NONE` indicates there are no external managers. For a VM that is managed by both Serviceguard and gWLM, parameters are separated with a comma. For example: `SG_host1,gWLM`. Do not specify this option. This option is used internally by Integrity VM.

Table 24 Options to the `hpvmmodify` command *(continued)*

Option	Description
`-j [0|1]`	Specifies whether the VM is a distributed guest (that is, managed by Serviceguard) and can be failed over to another cluster member running Integrity VM. Do not specify this option. This option is used internally by Integrity VM.
`-x` *`keyword=parameter`*	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max` • `ram_dyn_target_start` • `ram_dyn_entitlement=`*`amount`* • `amr_enable={0|1}` • `amr_chunk_size=`*`amount`* • `runnable_status` • `not_runnable_reason` • `graceful_stop_timeout` • `sched_preference` • `suspend={enable | disable}` • `suspend_file=delete` Specifies settings for OVMM: • `online_migration` • `migrate_init_phase_timeout` • `migrate_copy_phase_timeout` • `migrate_io_quiesce_phase_timeout` • `migrate_frozen_phase_timeout` For more information about dynamic memory, see “Managing dynamic memory from the guest” (page 257). Specifies VM or vPar attributes that control their behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}` Modifies the modify_status, visible_status, register_status, and runnable_status. For more information about the `hpvmmodify` command, see Table 24 (page 155).

If the `hpvmmodify` command does not display any warnings, the VSP system will be ready to start the VM.

After you make the necessary modifications, use the `hpvmstart` command to start the VM. For example:

```
# hpvmstart -P host1
(C) Copyright 2000 - 2016 Hewlett-Packard Development Company, L.P.
Initializing System Event Log
Initializing Forward Progress Log
Opening minor device and creating guest machine container
Creation of VM, minor device 2
Allocating guest memory: 2048MB
  allocating low RAM (0-40000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
```

```
/vmm_config.next): Allocated 1073741824 bytes at 0x6000000100000000
    locking memory: 0-40000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.next): Allocated 860 bytes at 0x6000000140000000
    locked SAL RAM: 00000000ffaa0000 (4KB)
    locked ESI RAM: 00000000ffaa1000 (4KB)
    locked PAL RAM: 00000000ffaa4000 (4KB)
    locked Min Save State: 00000000ffaa5000 (1KB)
RAM alignment: 40000000
Memory base low : 6000000100000000
Memory base FW  : 6000000140000000
Loading boot image
Image initial IP=102000 GP=62C000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'host1'
```

The VM `host1` is started. Now, the guest operating system must be installed.

NOTE: You might receive the following note-level message in the `/var/opt/hpvm/common/command.log` file under certain circumstances:

```
mm/dd/yy hh:mm:ss|NOTE|host|root|Unable to open file '/dev/rdisk/diskxxx' - Device busy.
```

This note might be logged if:

- A guest is configured with an attached `avio_stor` burner:

  `resource: -a burner:avio_stor::[b,d,t]:attach_path:`*`lunpath_hardware_path`*

- The guest is then booted to EFI.
- Then the `hpvmmodify` command is run to add a device or remove a device other than the burner.

You may safely ignore this message.

For information about creating HP-UX guests, see “Installing HP-UX vPars and Integrity VM” (page 22).

Cloning VMs

After you have created a guest, you can quickly and easily create additional guests by using the `hpvmclone` command. Such as the `hpvmcreate`, `hpvmmigrate`, and `hpvmmodify` commands, the `hpvmclone` command accepts the command options listed in Table 19 (page 146) for specifying virtual devices, network interfaces, and other VM characteristics. This allows you to create new guests with similar characteristics, but different virtual resources.

Table 25 (page 158) lists the options that can be used with the `hpvmclone` command.

Table 25 Options to the `hpvmclone` command

Option	Description
`-P` *`vm-name`*	Specifies the name of the existing VM to be cloned. You must specify either the `-P` option or the `-p` option.
`-p` *`vm-number`*	Specifies the number of the existing VM to be cloned. You must specify either the `-P` option or the `-p` option.
`-K` *`console_IP_Addr`*	Specifies the IP address used to connect to the virtual iLO Remote Console of the guest. The address must be specified in IPv4 dot notation or 0. If 0 is entered, then the guest will no longer have virtual iLO Remote Console access using IP.

Table 25 Options to the `hpvmclone` **command** *(continued)*

Option	Description
`-L` *`console_IP_Addr_Netmask`*	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.
`-N` *`clone-vm-name`*	Specifies the name of the new VM (the clone). The *`clone-vm-name`* can be up to 255 alphanumeric characters. The same VM name cannot already exist on the same VSP system.

`-e` *`percent[:max_percent]`*`|-E` *`cycles[:max_cycles]`*	Specifies the CPU entitlement of the VM in CPU cycles. To specify the percentage of CPU power, enter the following option:
	`-e` *`percent[:max_percent]`*
	To specify the clock cycles, enter one of the following options:
	`-E` *`cycles[:max_cycles]`*`M (for megahertz)`
	`-E` *`cycles[:max_cycles]`*`G (for gigahertz)`

`-l` *`vm_label`*	Specifies a descriptive label for this VM. The label can contain up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). To include spaces, the label must be quoted (" ").
`-B` *`start_attr`*	Specifies the startup behavior of the VM. For *`start_attr`*, enter one of the following keywords: `auto`: Automatically starts the VM when the VSP is started (autoboot). `manual`: The VM is not started automatically. Use the `hpvmstart` command to start the VM manually.
`-O` *`os_type`*`[:`*`version`*`]`	Specifies the type and version of the operating system running on the VM. For the *`os_type`* parameter, you can specify one of the following (case-insensitive) values: `hpux`
`-a` *`rsrc`*	Creates a virtual device for the new VM (clone). Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see “Storage devices” (page 60). For information about forming a virtual network device specification, see “Creating virtual and direct I/O networks” (page 122).
`-d` *`rsrc`*	Deletes a virtual device that is defined on the existing VM in the clone VM configuration. Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see “Storage devices” (page 60). For information about forming a virtual network device specification, see “Creating virtual and direct I/O networks” (page 122).
`-m` *`rsrc`*	Modifies a virtual device that is defined on the existing VM in the clone VM configuration. Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see “Storage devices” (page 60). For information about forming a virtual network device specification, see “Creating virtual and direct I/O networks” (page 122).
`-F`	Suppresses all resource-conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only at the direction of HPE Support.

Table 25 Options to the `hpvmclone` command *(continued)*

Option	Description
`-c` *`number_vcpus`*	Specifies the number of vCPUs the VM detects at boot time. If unspecified, the number defaults to one. The maximum number of vCPUs that you can allocate to a VM is the number of physical processors on the VSP system.
`-r` *`amount`*	Specifies the amount of memory available to the VM. Specify the amount as either *`amount`*`M` (for megabtyes) or *`amount`*`G` (for gigabytes).
`-S` *`amount`*	Specifies that the cloned guest must share the same virtual LAN (VLAN) ports as the source guest. By default, the `hpvmclone` command allocates VLAN ports that are different from those allocated to the guest that is the source of the clone operation. For more information about using VLANS on VMs, see “Configuring VLANs” (page 131).
`-g` *`group`*`[:{admin|oper}]`	Specifies a group authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-u` *`user`*`[:{admin|oper}]`	Specifies a user authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-x` *`keyword=parameter`*	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max` • `ram_dyn_target_start` • `ram_dyn_entitlement=`*`amount`* • `amr_enable={0|1}` • `amr_chunk_size=`*`amount`* • `graceful_stop_timeout` • `mac_address` • `sched_preference` • `serial_number` • `tunables` • `suspend={enable | disable}` • `suspend_file=delete` For OVMM, the parameters values are: • `online_migration` • `migrate_frozen_phase_timeout` • `migrate_copy_phase_timeout` • `migrate_io_quiesce_timeout` • `migrate_init_phase_timeout` For more information about dynamic memory, see “Managing dynamic memory from the guest” (page 257). Specifies the following VM or vPar attributes that control VM or vPar behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}` To specify the serial number of the new VM, enter `serial_number={new | same}`

For example, to clone the VM named `host3`, to create a new VM named `clone1`, enter the following commands. First, view the current guest status on the VSP:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type   OS Type  State  #VCPUs #Devs #Nets   Memory
==================== ===== ====== ======== ====== ===== ===== ======= ========
host1                  2   SH     HPUX     On (OS)   1     1     1       2 GB
host2                  3   SH     UNKNOWN  Off       1     1     1       1 GB
host3                  4   SH     HPUX     Off       1     1     1       2 GB
```

You can create a clone of `host3` by entering the following command. The new VM is named `clone1`:

```
# hpvmclone -P host3 -N clone1
```

To see the results of the command, enter the `hpvmstatus` command again:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type   OS Type  State   #VCPUs #Devs #Nets   Memory
==================== ===== ====== ======== ======  ===== ===== =======  ========
host1                  2   SH     HPUX     On (OS)    1     1     1      2 GB
host2                  3   SH     UNKNOWN  Off        1     1     1      1 GB
host3                  4   SH     HPUX     Off        1     1     1      2 GB
clone1                 5   SH     HPUX     Off        1     1     1      2 GB
```

The `hpvmclone` command creates a copy of an existing VM and its configuration information. This command copies the configuration files of the existing guest. It does not copy the actual data and software associated with the guest. Use the `-b` option to specify a storage device to be physically duplicated in the cloning process. The `clone_vm_name` must not already exist on the same VSP.

The new configuration information of the VM can be modified from the original configuration file by using command options. If you do not specify any options, all the original parameters are retained. This causes resource conflicts if both the original and clone VMs are booted together.

Resources are checked to determine whether the VM could boot by itself on the server. Problems are reported as WARNINGS. These warnings do not prevent the new VM from being created. These conditions will, however, prevent the guest from starting.

Backing storage devices (for example, directories and files) cannot be shared, and therefore they cannot be used by two running guests at the same time. In this case, you must either enter a different backing store, or run only one of the guests at a time. For more information about storage devices, see “Storage devices” (page 60).

Use the `-b` option to specify a storage device to be physically duplicated in the cloning process. This feature allows you to specify any number of storage devices, and supports all the possible physical device types (`disk`, `lv`, and `file`), with the exception of NPIV HBAs.

Because there is no guarantee that other VMs would be running at the same time the new VM, use the following command to check the device for dependents:

```
# hpvmdevmgmt -l entry_name
```

For more information about the `hpvmdevmgmt` command and the guest device management database, see “Storage devices” (page 60).

Stopping VMs

NOTE: To stop a guest, Hewlett Packard Enterprise recommends that you perform an operating system shutdown from a privileged account on the guest using native operating system commands. If the guest does not respond, use the `hpvmstop -g` command on the VSP. Do not stop a guest by killing the `hpvmapp` process.

To stop a running VM, use the `hpvmstop` command. You must confirm this command. Table 26 (page 162) lists the options that can be used with the `hpvmstop` command:

Table 26 Options to the `hpvmstop` command

Option	Description
`-P` *vm-name*	Specifies the name of the VM.
`-p` *vm_number*	Specifies the number of the VM. To display the VM number, enter the `hpvmstatus` command.
`-a`	Specifies all the VMs that are running. You must also specify the `-F` option.
`-h`	Performs a hard stop on the VM, similar to a power failure. This is the default.
`-g`	Performs a graceful shutdown on the VM.
`-F`	Forces the command to act without requiring confirmation. **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.
`-Q`	Performs the operation without requiring you to confirm the command.
`-q`	Makes certain scripted operations less verbose (quiet mode).

For example, the following command stops the VM named `host1`.

```
# hpvmstop -P host1
hpvmstop: Stop the virtual machine 'host1'? [n/y]: y
```

The default action of this command (if you press **Enter**) is to not perform the command operation. To continue the operation, you must enter **`y`**.

The `hpvmstatus` command shows that the VM is `Off`.

```
# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  Type    OS Type   State   #VCPUs  #Devs  #Nets    Memory
==================== ===== ======= ======== ======  =====   =====  =======  ======
config1                 1   SH      HPUX      Off        1     5        1    512 MB
config2                 2   SH      HPUX      Off        1     7        1      1 GB
guest1                  5   SH      HPUX      On (OS)    1     5        1      1 GB
host1                  12   SH      UNKNOWN   Off        1     0        0      2 GB
```

To enter the command without requiring a confirmation (for example, in a script), enter the following command:

```
# hpvmstop -P host1 -Q
#
```

To quickly shut down all three VMs that are running on the VSP, enter the following command:

```
# hpvmstop -a -F
Stopping virtual machine host1
Stopping virtual machine host2
Stopping virtual machine host3
```

NOTE: When stopping a guest that is running a heavy I/O load, the `hpvmstop` command can exhaust the timeout allotted for stop and exit. When this happens, the `SIGKILL` has been sent to the running `hpvmapp` process and will be received by that process when pending I/Os complete. The `SIGKILL` then terminates the guest.

This is expected behavior for an I/O intensive process. This behavior is not specific to Integrity VM, but is how the signal-delivery mechanism works in the HP-UX operating system.

You can also use the `hpvmconsole` command to force the VM to shut down. However, after you install the guest operating system, you must use the standard operating system commands and procedures on the guest to shut it down.

Removing VMs

To remove a VM from the VSP, use the `hpvmremove` command. By default, you are required to confirm this action. Table 27 (page 163) lists the options that can be used with the `hpvmremove` command.

Table 27 Options to the `hpvmremove` command

Option	Description
`-P` *vm-name*	Specifies the name of the VM. You must include either the `-P` or `-p` option.
`-p` *vm_number*	Specifies the number of the VM. To view the VM number, run the `hpvmstatus` command.
`-F`	Forces the command to act regardless of errors. **NOTE:** The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.
`-Q`	Performs the command without requiring user input to confirm.

For example, the following command removes the VM named `host1`. The subsequent `hpvmstatus` command shows that `host1` is removed:

```
# hpvmremove -P host1
hpvmremove: Remove the virtual machine 'host1'? [n/y]: y
```

The default action of this command (if you press **Enter**) is to not perform the command action. To perform the action, you must enter **y**.

This command removes `host1` and all its configuration files, and restores resources allocated to that guest to the pool of available resources of the VSP. (Any guest operating system and application data on the VSP storage devices are not affected.)

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type    OS Type  State  #VCPUs #Devs  #Nets    Memory
==================== ===== ======= ======== ====== ===== =====  =======  ======
config1                  1 SH      HPUX     Off       1      5      1       512 MB
config2                  2 SH      HPUX     Off       1      7      1         1 GB
guest1                   5 SH      HPUX     On (OS)   1      5      1         1 GB
```

To remove the guest without requiring user confirmation (for example, in a script), enter the following command:

```
# hpvmremove -P host1 -Q
```

Troubleshooting VM creation problems

For more information about troubleshooting VM creation problems, see “Creating VMs” (page 292).

10 Administering vPars

To create vPars, you must run appropriate commands from the VSP or use the HP-UX Integrity Virtual Server Manager, the GUI application, which you can access from the **Tools** page in HP SMH installed on the VSP.

This chapter discusses the various tasks that you can perform from the VSP using the commands. For more information about the tasks that you can perform using the GUI, see HP-UX integrity virtual server manager help that comes with the GUI application.

NOTE: The Integrity VM commands can be used to configure and manage both vPars and VM. They support overall product features. Hewlett Packard Enterprise recommends using Integrity VM commands over vPar commands for managing vPars or VM.

Taking backups of guest configurations

For guidelines and recommendations on taking backups of guest configurations, see "Taking backups of guest configurations" (page 145).

Creating a vPar

When you create a vPar, you must specify its attributes. Later, you can change these attributes.

You can set the attributes of a vPar using the following commands:

- `vparcreate`, which creates a new vPar.
- `vparmodify`, which modifies an existing vPar.

Both these commands accept the same options for specifying the attributes of a vPar. Table 28 (page 164) lists the attributes and the command options.

NOTE: When you use the `vparcreate` command to create a vPar, by default it reserves any resources assigned to that vPar, even when the vPar is off. For more information about reserved resources, see "Reserved resources and resource over-commitment" (page 54). Additionally, the vPar is set to AutoBoot when the VSP is restarted. You can use the `hpvmmodify -B` command to adjust the AutoBoot setting.

Table 28 Attributes of a vPar

vPar attributes	Description	Command option	Default value
vPar ID (name or number)	You can specify a number or a name.	`-p vpar_id`	If you do not specify either a number or a name, a vPar name in the format vParXXXX (where XXXX represents the vPar Id number), with leading zeros is automatically assigned to the newly created vPar.

CPU	You can specify the number of CPUs that a vPar can use. A running vPar cannot use more CPUs than the number of physical CPUs on the VSP system.
	
	You can set min and max values. The min and max values are boundary values that are enforced if the number of CPUs in this vPar changes in the future.
		`-a cpu::num`
		
		`-a cpu::[num]:[min][:[max]]`
		
		OR
		
		`-a core::num`
		
		`-a core::[num]:[min][:[max]]`
			If you do not specify this attribute when you create a vPar, the default is 1 CPU core. If you set any of num, min, or max to 0, the default value is assigned. In vPars v6, the defaults are, num=1, min=1, and max=512.

Table 28 Attributes of a vPar *(continued)*

vPar attributes	Description	Command option	Default value
Memory	The memory is specified in megabytes. The minimum amount of memory you allocate to a vPar must be the total of the following: • The amount of memory required by the operating environment in the vPar. • The amount of memory required by the applications running on the vPar.	`-a mem::mem_size[:{b|f}]`	If you do not specify this attribute when you create a vPar, the default memory allocated is 2 GB. For more information, see Table 38 (page 262).
I/O (virtual devices)	You can allocate virtual network switches and virtual storage devices to the vPar. The VSP presents devices as virtual devices to the vPar. The vPar network consists of vNICs and vswitches. For vPars to communicate either with other vPars or outside the VSP system, each virtual network of the vPar must be associated with a vswitch. If you start a vPar without a vswitch, the vPar has no network communication channel. vPar also supports DIO networking where physical devices are directly presented. DIO devices do not require a vswitch. Virtual storage devices are backed by physical devices on the VSP system. You can specify one of the following devices – disk, dvd, tape, changer, burner, or hba. For more information about virtual devices, see "Virtual devices" (page 148).	`-a rsrc`	If you do not specify this attribute when you create a vPar, the vPar will not have access to network and storage devices.
Virtual iLO Remote Console	You can access the Virtual iLO Remote Console of the vPar using `telnet` or `ssh`. This attribute is the IP address that is used to connect to the Virtual iLO Remote Console of the vPar. You must specify the address in IPv4 dot-decimal notation. If the `-K` option is specified, then the `-L` option must be specified. By default, the root user may access the console of the vPar using the `vparconsole` command or through the Virtual iLO Remote Console, if	`-K console_ip`	If you do not specify this attribute when you create a vPar, the remote console is not started, that is, the virtual console can be accessed only using the `vparconsole` command.

Table 28 Attributes of a vPar *(continued)*

vPar attributes	Description	Command option	Default value
	configured. There is no need to configure a console account if the root user for this purpose does not violate security policy. However, access to the console, through the `vparconsole` command or the remote console, can be granted to groups or individual users, with either administrative or operator virtual iLO permissions.		
IPv4 subnet mask for accessing the Virtual iLO Remote Console	To access the Virtual iLO Remote Console of the vPar if you have specified its IP address using the `-K` option, then you must specify the IPv4 subnet mask too.	`-L netmask`	Not applicable.
Group with administrator or operator privileges	You can specify admin or operator privileges for a group of users.	`-g group:{admin|oper}`	If you do not specify the group authorization, then only the root user has access to the virtual console.
User with administrator or operator privileges	You can specify admin or operator privileges for a user.	`-u user:{admin|oper}`	If you do not specify the user authorization, then only the root user has access to the virtual console.

Example 32 Create a default vPar

Run the `vparcreate` command to create a basic vPar with the default values of 1 CPU, 2 GB memory, and no I/O.

```
# vparcreate
[Creating vPar0001.]
```

Later, use the `vparmodify` command to add I/O and modify other attributes.

```
# vparmodify -p vPar0001 -a network:avio_lan::vswitch:sitelan \
-a hba:avio_stor::npiv:/dev/fcd0
```

Specifying CPU or core min and max limits

The syntax to specify min and max CPUs assigned to a vPar is:

`-[a|m] cpu::[num]:[min][:[max]]`

where:

`-a`
: add (used with `vparcreate` or `vparmodify`).

`-m`
: modify (used with `vparmodify`).

`min`
: the minimum number of CPUs that must remain assigned to the partition.

`max`
: the maximum number of CPUs that can be assigned to the vPar.

NOTE: The vPar can be either UP or DOWN when setting the min or max value. Hence, a reboot is not necessary when you modify the min and max value. When the partition is UP, the CPU count can only be adjusted if the HP-UX OS on the vPar is running. CPU counts cannot be adjusted while the vPar is in EFI state.

Example 33 Setting the minimum number of CPUs to 2

```
hostmachine# vparmodify -p machinename -m cpu:::2
```

Example 34 Setting the minimum number of CPUs to 2 and the maximum to 4

```
hostmachine# vparmodify -p machinename -m cpu:::2:4
```

Adding and deleting CPUs or cores by total

The basic syntax for adding and deleting CPUs is:

`-[a|d|m] cpu::`*`num`*

where:

`-a|d|m`

specifies adding, deleting, or modifying the *`total`* count of CPUs.

`num`

specifies the number of CPUs.

NOTE: The vPar can be either UP or DOWN when using the `cpu::num` syntax.

When the vPar is active, CPUs that were added using the `cpu::num` syntax can be deleted only by using `cpu::`*`num`* syntax.

The total increases or decreases by *`num`* when the `-a` or `-d` option is used, and is set to *`num`* when the `-m` option is used.

vPar does not support assignment of resources based on hardware path or socket locality.

Example 35 Add two CPUs or cores to a vPar

```
hostmachine# vparmodify -p machinename -a cpu::2
```

Example 36 Delete two CPUs or cores from a vPar

```
hostmachine# vparmodify -p machinename -d cpu::2
```

Example 37 Modify the total count of CPUs or cores of a vPar

```
hostmachine# vparmodify -p machinename -m cpu::4
```

Specifying base and floating memory

Starting v6.2, vPar memory is of two types:

- **Base memory** – This can be used by vPar kernel for critical data structures. You can add, but cannot delete base memory from a live vPar.
- **Floating memory** – This is typically used for user applications. You can either add or delete floating memory from a live vPar.

Syntax for specifying memory with `vparcreate` command is:

`-a mem::mem_size[:{b|f}]`

Example 38 Create a vPar with 4 GB base memory

```
hostmachine# vparcreate -p machinename -a mem::4G:b
```

Example 39 Create a vPar with 2 GB base memory and 2 GB floating memory

```
hostmachine# vparcreate -p machinename -a mem::2G:b –a mem::2G:f
```

For more information about base and floating memory, see “Guidelines for base and floating memory configuration ” (page 58).

Specifying I/O devices

Example 40 Create a vPar with virtual network interface backed by virtual switch

Create a vPar named Oslo in the local system, specifying 2 GB of memory, 2 CPUs, and virtual network interface backed by virtual switch “sitelan”.

```
# vparcreate -p Oslo -a mem::2048 -a cpu::2 -a
network:avio_lan::vswitch:sitelan
```

For more information about creating and managing virtual switches, see “Creating virtual and direct I/O networks” (page 122).

Example 41 Create a vPar with virtual disk backed by a whole disk

Create a vPar named Oslo in the local system, specifying memory of 2 GB, 2 CPUs, and a virtual disk backed by a whole disk “/dev/rdisk/disk70”.

```
# vparcreate -p Oslo -a mem::2048 -a cpu::2 -a
disk:avio_star::disk:/dev/rdisk/disk70
```

Example 42 Create a vPar with NPIV HBA

Create a vPar named vpar001 with a virtualized HBA using NPIV port assuming a GUID manager is available to assign World Wide Port Name and World Wide Node Name.

```
# vparcreate -p vpar001 -a hba:avio_stor::npiv:/dev/fcd0
```

For additional information about configuring NPIV, see the *vparresources3(5)* and “NPIV with vPars and Integrity VM” (page 99).

Example 43 Create a vPar with network interface backed by a DIO function

Add the DIO function “0/0/0/4/0/0/0” to the direct I/O pool using the `hpvmhwmgmt` command:

```
# hpvmhwmgmt -p dio -a 0/0/0/4/0/0/0
```

Create a vPar named Oslo in the local system, specifying memory of 2 GB, 2 CPUs, and virtual network interface backed by a DIO function “0/0/0/4/0/0/0”.

```
# vparcreate -p Oslo -a mem::2048 -a cpu::2 -a lan:dio::hwpath:
0/0/0/4/0/0/0
```

For more information about configuration of guests with DIO functions, see “Direct I/O networking” (page 138).

Booting a vPar

You can boot and manage vPars using the same storage media and procedures that you would use if the vPar operating system were running on its own dedicated physical hardware platform. You can allocate administration privileges to specific vPar administrators.

To boot a vPar, the vPar must be in the DOWN run state. To boot a vPar, you must run the `vparboot` command or provide the `-c` *"pc -on"* parameters to the `vparconsole` command.

Each vPar has a console, and you can access the console from the VSP using the `vparconsole` command. Start the console before you run the `vparboot` command if there is a need to interact with EFI. You can also provide the `-f -i -c` *"pc -on"* parameters to the `vparconsole` command to start, and enter the console in interactive mode right after the start.

Example 44 Boot the vPar called Oslo

```
# vparboot -p Oslo
```

OR

```
# vparconsole -P Oslo -f -i -c 'pc -on'
```

For more information about installation of guest OS on vPar, see "Installing HP-UX vPars and Integrity VM" (page 22).

Modifying a vPar

You can modify all the attributes that you specify while creating a vPar. You can rename the vPar, modify the resources, and change group and user level authorization. Some attributes can be modified dynamically, that is, a reboot is not required, while others require a reboot.

The `vparmodify` command must be run from the VSP just as the `vparcreate` command.

The same options used for creating a vPar are applicable for modifying the vPar.

Modifying CPU and Memory resources dynamically

Since vPars and Integrity VM v6.2, you can modify CPU cores and memory dynamically. You can change the CPU core count of a vPar while it is running. Do not reboot the vPar after you modify the CPU core count. You can add base and floating memory to vPar while it is running. You cannot change CPU and MEM online at the same time. For more information about base and floating memory, see "Guidelines for base and floating memory configuration " (page 58).

NOTE: Base memory can be removed only when the vPar is DOWN.

Modifying I/O resources statically

Starting with vPars and Integrity VM v6.3, IO devices can be added to a vPar dynamically.

For more information about IO devices, see "Storage devices" (page 60).

Modifying vPar name and number

The vPar must be in the DOWN run state to modify the name. You can modify the name of a vPar using the `vparmodify -P` command to add a name that does not exist in the current vPar database. The vPar number cannot be modified. The only way you can get a different number is to delete the current vPar and create a new one. When you create a new vPar, you can specify the vPar number with the `-p` option.

Viewing information specific to a vPar

You can view information about a vPar by specifying either the name or the number of the vPar. You must use the `vparstatus` command from the VSP to view vPar information.

By default, the `vparstatus` command displays summary information. To view detailed information you must use the `-v` option. You can also view the vPar information in machine-readable format using the `-M` option.

Alternatively, the `hpvmstatus` command can also be used to view the detail status of vPar. However, when the `hpvmstatus` command is executed without any option, it displays the summary information about both vPar and VM on the VSP.

The information that the `vparstatus` command (and the `hpvmstatus` command) displays includes the following:

- Number and name of the vPar.
- State of the vPar – active or inactive.
- Run-state of the active or inactive vPar.
- Summary of CPU, I/O, and memory resource allocations.
  In both summary and detailed machine-readable format, the following information for the specified vPar is displayed:
  - Total memory size in MB.
  - The number of CPUs assigned to the vPar.
  - The virtual I/O devices assigned to the vPar in the resource statement format.

To view summary information about all the vPars, run the following command:

```
# vparstatus
```

To view the detailed attributes and resources of a specific vPar, for example vpar1, run the following command:

```
# vparstatus -p vpar1 -v
```

To view the detailed attributes and resources of a specific vPar named vpar1 in machine-readable format, run the following command:

```
# vparstatus -p vpar1 -M
```

To view the revisions of partition management tools, run the following command:

```
# vparstatus -r
```

Stopping and resetting a vPar

Shutdown

When a vPar must be completely shut down and not be rebooted, the `-g` option can be used. You can issue a graceful shut down to the OS by using the `vparreset` command.

To shut down a vPar named Oslo, run the following command:

```
# vparreset -f -p Oslo -g
```

NOTE: This command functions only when the guest OS is running, and only if the guest OS is capable of responding to the graceful shutdown request. This command only initiates the graceful shutdown operation, it does not consequently report failure if the OS fails to gracefully shutdown. The preferred method for stopping a vPar is to log in to it, stop all the applications, and then run the `/etc/shutdown -h` command.

If the OS of a vPar becomes unresponsive, there is no prompt neither from a network connection nor through the virtual console (vparconsole). In such a situation, you must manually reset the partition. When a vPar is unresponsive, instead of shutting down the vPar, you can reset or restart the vPar. To recover a vPar that is unresponsive, you can use the `vparreset` command.

CAUTION: When the `vparreset` command is used accidentally, serious consequences can occur. Hence, the `-f` (force) option is required with this command.

You can perform any of the following reset operations.

Hard reset

The hard reset is equivalent to specifying `RS` command in the management processor. You must only do a hard reset if you cannot get the OS to issue its own reboot or shutdown process. The vPar restarts after the hard reset.

To hard reset a vPar named Oslo, run the following command:

```
# vparreset -f -p Oslo -h
```

Power off

The power off option `-d` is useful to break out of a reboot loop, that is, when you do not want the vPar to be rebooted. In such a case, you must manually restart the vPar using the `vparboot` command.

To power off a vPar named Oslo, run the following command:

```
# vparreset -f -p Oslo -d
```

IMPORTANT: In the case of both hard reset and power-off, the operating system of the vPar is abruptly shut down and the crash dump of the OS is not saved. Hence, Hewlett Packard Enterprise recommends shutting down the vPar from the vPar using the `shutdown` command.

Soft reset (transfer of control - TOC)

When you do not specify any option with the `vparreset` command, a soft reset is performed by default. In a soft reset, the crash dump of the OS running on the vPar is saved. This enables the HPE engineers to debug the problem that caused the unresponsiveness. The `-t` option is used for a soft reset. The vPar is restarted after the soft reset is issued.

To soft reset a vPar named Oslo, run the following command:

```
# vparreset -f -p Oslo
```

OR

```
# vparreset -f -p Oslo -t
```

Removing a vPar

When you want to permanently delete a vPar, you can use the `vparremove` command. The vPar must be in the DOWN run-state before you delete the vPar. To bring a vPar to the DOWN run-state, you can either power down (`vparreset` command with `-d` option) the vPar or shutdown the vPar (`vparreset` command with `-g` option).

CAUTION: When the `vparremove` command is used accidently, serious consequences can occur.

Hence, the `-f` (force) option is required with the command.

To remove a vPar named Oslo, run the following command:

```
# vparremove -p Oslo -f
```

Deactivating a vPar configuration

You can deactivate a vPar to remove or deallocate resources from it, while maintaining its configuration settings. This is a way of managing shadow configurations, and allows the shadow configuration on a per vPar basis. The `-x active_config=false` option must be used with either the `vparcreate` or the `vparmodify` command.

You can deactivate a vPar configuration only if the vPar is in the inactive state, that is, the run-state must be DOWN.

To deactivate a single vPar configuration, the `vparmodify` command must be used with the `-x active_config=false` option. After this is done, the vPar instance no longer consumes or reserves the resources allocated to it, and those resources may be distributed to other partitions or the VSP, or those resources may be used to a create different vPar instance.

To reactivate the vPar configuration use `vparmodify` command with the `-x active_config=true`.

NOTE: A vPar configuration cannot be reactivated unless the resources it requires are available and not reserved by other vPar instances. A vPar can still be managed while its configuration is deactivated. However, it cannot be booted.

Example 45 Deactivating a vPar named Gold

```
# vparmodify -p Gold -x active_config=false
```

11 PCI OLR support on VSPs

Online Addition, Replacement and Removal of PCI I/O devices (PCI OLARD) is an important value proposition of HPE Integrity Superdome 2 (SD2) platforms. The OLR functionality provides assurance of continued system availability even when potential problems are identified with active I/O resources.

On SD2 platforms configured as VSP with versions earlier to HP-UX vPars and Integrity VM v6.3, PCI OLR operations are possible only on the host for host devices which are not used by any active guests (both vPars and Integrity VMs).

With the release of HP-UX 11i v3 AR1403 and HP-UX vPars and Integrity VM v6.3, the PCI OLR infrastructure on host and guests have been integrated to ensure that, if required, host devices acting as backing stores or backing interfaces for active guests can be replaced without downtime for any guest or host. The `olrad(1M)` command performs and collates CRA (Critical Resource Analysis) within each guest active on the PCI I/O device that must be replaced. After the PCI I/O device is replaced, all I/O activity within host and all affected guests resume.

Online Addition and Deletion of PCI I/O devices

On platforms that support it, PCI OLA operations are used to add a new PCI I/O card to running operating system instances. With v6.3, for both vPar and Integrity VM guests equivalent functionality is provided by the enhanced `hpvmmodify(1M)` command. This feature is referred to as Dynamic I/O Addition (not to be confused with Direct I/O).

Use cases and benefits of PCI OLR on a VSP

PCI OLR functionality helps to ensure high availability of supported platforms by making it possible to replace I/O cards that shows failure indications without any system downtime.

Dependencies and prerequisites

Software dependencies

The changes are made to both the PCI OLR infrastructure on host and guests. In addition to HP-UX vPars and Integrity VM v6.3 or later, the following host patches are required on SD2 VSP to enable this functionality.

- PHKL_42548
- PHCO_42592
- PHCO_42623
- PHCO_43715

You must ensure that the guest kit on all guests running on the SD2 VSP is upgraded to the v6.3 or later release.

Hardware dependencies

PCI OLR functionality for VSPs is supported only on the SD2 platforms.

Performing PCI OLR on a VSP

Online replacement of I/O cards on a SD2 server configured as a VSP is done exactly the way it is done on a native SD2 server. This section describes additional aspects that apply only to a SD2 server configured as a VSP, including additional aspects considered during a CRA for various types of I/O devices, additional CRA logs and errors, and the differences with respect to time taken for an I/O card replacement in a vPars and Integrity VM environment vice versa an SD server configured as native HP-UX.

CRA on a VSP

On a standalone SD2 server, before an online replacement of an I/O card, a CRA of all the system resources that are impacted by the unavailability of the card in question is performed. Only when this analysis indicates that there is no impact to any of the critical system resources and the operation is safe and will not cause disruption in the functioning of the system, the `olrad` command proceeds to prepare the I/O card for replacement.

Starting with vPars and Integrity VM v6.3, when the SD2 server is configured as a VSP, then, in addition to performing a CRA on the VSP, the `olrad` command triggers a similar CRA on all active vPars or VM guests that have IO resources backed by the card being replaced. Only when the CRA succeeds across all the active guests and the VSP, does the `olrad` command proceed to prepare the I/O card for replacement. Hewlett Packard Enterprise recommends that the VSP administrator runs the `olrad` command with the `-C` option to check the criticality of the I/O card across the VSP and all active guests before attempting to perform an online replacement operation.

This command lists one of the following severity levels:

CRA_SUCCESS	No affected resources in use either on the VSP or on any of the active vPars or VM guests.
CRA_WARNING	Resources are in use on affected devices either on the VSP or on any of the active guests, but none are deemed critical.
CRA_DATA_CRITICAL	Resources are in use on affected devices either on the VSP or on any of the active guests, and there is a probable data loss. The operation must only proceed with the permission of the user.
CRA_SYS_CRITICAL	Resources are in use on affected devices either on the VSP or on any of the active guests, and the operation is likely to bring down the VSP or one or more of the active vPars or VM guests.
CRA_FAILURE	Indicates that an internal CRA error was encountered and the CRA across the VSP or one or more of the active vPars or VM guests cannot be completed.

For more information about CRA framework, results and reports generated by CRA for various configurations, and scenarios on a system installed with the HP-UX 11i v3 operating system, see the white paper **Critical Resource Analysis**.

NPIV devices

Starting with vPars and Integrity VM v6.3, an FC card in an OLR capable PCI slot can be replaced without having to bring down active vPars or VM guests that are configured with NPIV HBAs backed by the FC card. This can be done as long as the NPIV devices impacted are not critical for the operation of any of the vPar or VM guest. In addition, there must be no impacted devices on the VSP being critical to the operation of the VSP.

When a CRA check is done on an FC card seen by the VSP, the `olrad` command triggers a parallel CRA check on each of the vPar or VM guests with NPIV HBAs backed by the FC card. These CRA checks in the vPar or VM guests take into account all aspects that are considered by the Mass Storage CRA checks on a native SD2 server. This includes scenarios like boot and alternate boot path configurations, swap and dump device configurations, Serviceguard lock disk configurations, File system and Volume manager configurations, I/O in progress configuration, and various SAN storage configurations. Only when all these aspects are analyzed per vPar or VM guest and the VSP, and found to have no system critical impact, does the `olrad` command proceed with the next steps.

For more information about how a resource analysis is performed on mass storage components of a system, see the white paper **Critical Resource Analysis**. All the scenarios described in the white paper are applicable to NPIV devices seen within a vPar or VM guest.

If none of the active vPars, VM guest, or the VSP have a system critical impact due to the removal of the storage I/O card, then, the `olrad` command prepares the card for replacement. After this is done, the vPars or VM guests having NPIV HBAs backed by the FC ports on the card being replaced still shows the vHBAs as CLAIMED, but all impacted targets paths and LUN paths goes into the NO_HW state. The target and LUN paths get back to the `CLAIMED` state after the OLRAD operation completes and the new card is put into the I/O slot and powered ON.

NOTE: When a 16Gb Qlogic card with the latest firmware version (>= 8.1.80) and having bandwidth entitlement enabled is OLR'ed and replaced with a card having an older firmware version, the NPIV HBAs may remain in an OFFLINE state forever. For more information on getting the NPIV HBA back online, see "NPIV devices with bandwidth entitlement" (page 295).

DIO devices

Starting with vPars and Integrity VM v6.3, NIC residing in an OL* capable slot and configured to the DIO pool can be replaced without having to bring down active vPars or VM guests to which the functions (ports) of the NIC are assigned. CRA step is performed as a part of the card replacement operations to ensure that the operation is allowed to proceed only if CRA determines that the associated ports of the NIC are not critical for the operation of the vPar or VM. In addition, you must not have impacted devices on the VSP being critical to the operation of the VSP.

To determine whether a particular port of an NIC configured to the DIO pool is critical for the operation of a vPar or VM, LAN CRA module in the vPar or VM is consulted which performs usage analysis and reports any potential impacts from LAN subsystem. Some of the usage scenarios determined by LAN CRA includes NIC port configured with VLAN and IP address, and connected to network, APA (Auto-port Aggregation) group with the link aggregate containing LAN ports from different NIC ports, and so on. For more information about how a resource analysis is performed on LAN components of a system, see the white paper **Critical Resource Analysis**. The usage analysis result from each of the vPar or VM is consolidated and a cumulative criticality is passed back to the `olrad(1M)` command.

After the CRA phase ends, OLRAD performs the step of Pre Replace operation (`olrad -r`) which involves suspending the NIC ports. In the case of NIC configured to the DIO pool and assigned to an active vPar or VM guest, Suspend operation is performed on the NIC port in the vPar or VM and subsequently the Suspend operation is performed on the DIO pool resources on the VSP host claimed by the `hpvmdio` driver. At the end of Pre Replace operation, the `ioscan(1M)` command shows the state of the NIC ports in the vPar or VM as SUSPENDED state. The state of the DIO pool resources on the VSP host will also be shown as SUSPENDED in the `ioscan(1M)` command output.

Upon successful completion of Pre Replace operation, slot will be physically powered off state and at this point you can safely replace the existing card in the slot with another identical NIC (Like to Like replacement). Subsequently, you can enter the `Post Replace` option of the `olrad(1m)` command (`olrad -R`) which results in all the DIO pool resources in the VSP host coming back to the CLAIMED state. Each of the NIC port in the vPar or VM guest is also brought back to CLAIMED state. After this step, you can continue using the NIC ports.

AVIO Networking devices

Starting with vPars and Integrity VM v6.3, a physical NIC plugged onto an OLR capable PCI slot connected to a vswitch, now does additional guest CRA when OLRAD CRA related commands are issued during suspend and replace of the card.

Prior to vPars and Integrity VM v6.3, when an NIC in an OLR capable PCI slot on VSP is replaced (`olrad -r`), it caused the change in vswitch state and state gets transitioned from `UP` to

LinkDown state. The state change event is transmitted to guest using the vswitch and the vNICs state moved to halt. The olrad(1m) command suspends the card without considering the state change implication on the guests using the impacted vNICs.

Starting with vPars and Integrity VM v6.3, the olrad CRA request issued for the physical NIC on VSP initiates parallel CRA check in each of vPar and VM guests for the guest associated with vswitch. LAN CRA module in the vPar or VM performs usage analysis and reports any potential impacts from LAN subsystem perspective. Some of the usage scenarios determined by LAN CRA includes NIC port configured with VLAN and IP address, connected to network, and so on. For more information about how a resource analysis is performed on LAN components of a system, see the white paper **Critical Resource Analysis**. The usage analysis result from each of the vPar or VM is consolidated and a cumulative criticality is passed back to the olrad(1M) command.

After the CRA is complete, the olrad(1M) command suspends the card based on the criticality returned from the CRA. The slot will be powered off and the link state of vswitch goes to LinkDown and link state of NIC goes to DOWN state. The old NIC can be safely replaced with a newer NIC card. After the card is replaced using the olrad -R command, the link state of the vswitch and the vNICs associated with it, gets transitioned to UP state.

NOTE: When Serviceguard is configured as a package, guest LAN interfaces used for exchanging Serviceguard heartbeat packets are backed by a physical NIC on the VSP. A CRA operation on such NICs may report that the card is DATA CRITICAL. This is consistent with the operation of CRA host systems using Serviceguard in a non-virtualized environment.

AVIO Storage devices

For non-NPIV based AVIO storage devices (referred to as legacy AVIO storage), unlike the NPIV devices, the multi-pathing capabilities reside on the VSP and each device has only a single path within a vPar or VM guest. For example, for a disk backing store, the multiple paths seen on the VSP map to a single path seen within the vPar or VM guest.

Prior to vPars and Integrity VM v6.3, a storage I/O card in an OLR capable PCI slot on the VSP can be replaced without bringing down active vPars or VM guests configured with backing stores having paths configured through the card being replaced. This can be done as long as there is at least one unaffected path in the VSP to the vPar or VM guest backing stores that are in use. In addition, you must not have impacted devices on the VSP being critical to the operation of the VSP itself. The olrad command when run on the VSP cannot distinguish between a system critical usage and a data critical usage of the vPar or VM backing stores.

Starting with vPars and Integrity VM v6.3, the olrad command when run on the VSP, can distinguish between a guest boot device and a data device. Starting with v6.3, a storage I/O card on the VSP cannot be replaced without bringing down active vPars or VM guests booted with an AVIO backing store if the I/O card impacts the only available path to the boot device of the vPar or VM guests.

Unlike in the case of NPIV devices, the CRA in the case of legacy AVIO devices only consider the backing store on which the vPar or VM guest is currently booted, as a system critical resource. In some cases, the current boot device may be different from the primary or alternate boot device configured for the vPar or VM guest.

NOTE: With legacy AVIO devices, the CRA will not consider the impact to the primary boot (if different from the current boot device) and an alternate boot disk configurations, swap and dump device configurations, and Serviceguard lock disk configurations as system critical. All of these will be reported as DATA CRITICAL impact if the only available path will go down when the card is replaced. Hence, the usage of the `olrad` command with the `-f` option to override the DATA CRITICAL errors must be exercised with utmost caution.

When the CRA for legacy AVIO storage devices reports the severity for a particular vPar or VM guest as DATA CRITICAL, the VSP administrator must work with the specific guest administrators to manually check and ensure that no primary or alternate boot devices, swap or dump devices, or Serviceguard lock disks are impacted by the unavailability of the card being replaced.

CRA logs

The CRA infrastructure collates the detailed analysis logs from all the subsystem CRA modules and returns the combined logs at the location `/var/adm/cra.log` on the VSP. When the `olrad` command is invoked on a VSP, the CRA log on the VSP will have relevant entries under the following scope:

HPVM NPIV	Guest wise analysis for each vPar or VM guest that has an NPIV resource impacted by the OLRAD operation.
HPVM Direct I/O	Guest wise analysis for each vPar or VM guest that has a DIO resource impacted by the OLRAD operation.
HPVM AVIO Networking	Guest wise analysis for each vPar or VM guest that has a AVIO LAN resource impacted by the OLRAD operation.
HPVM legacy AVIO Storage	Guest wise analysis for each vPar or VM guest that is booted off a legacy AVIO (non-NPIV) backing store, that is impacted by the OLRAD operation.

In cases where a successful CRA can be conducted on the active guests, the guest wise analysis displays the guest instance number along with the severity associated with each guest.

In cases where any of the HPVM components failed to perform a CRA, the guest wise analysis displays the guest instance number along with the reason for failure in analysis, for each guest.

The CRA log on the VSP only provides the overall result of analysis per guest. The VSP administrator has to work with the guest administrators for each of the impacted guests to get the detailed CRA per guest. The detailed CRA for each guest is available in the guest, at the location `/var/adm/cra.log`.

PCI OLR failures

This section describes in detail about the possible errors that can be encountered during a CRA or OLRAD operation on an SD2 configured as a VSP and the workaround to proceed with the CRA or OLRAD operation.

A change in guest configuration is in progress, retry the operation

A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars having IO devices backed by the card being removed in the middle of the change in guest configuration such as addition or removal of CPU, memory, or device using `hpvmmodify` or `vparmodify`command. This operation must be retried after the modification completes.

The guest is at EFI, retry after the guest boot completes

A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars or VM guests with impacted IO devices is at EFI. The operation must be retried after the vPars and VM guests have completed the boot process.

The guest is being shut down or is being rebooted, retry the operation

A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars or VM guests with impacted IO devices are in the middle of being shut down or rebooted. The operation must be retried after the vPars and VM guests have either completed the shutdown or the boot process.

The guest is running a pre-6.3 VirtualBase bundle upgrade to the latest VirtualBase bundle and retry the operation

A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars or VM guests with impacted IO has a pre v6.3 VirtualBase bundle installed. The vPars or VMs have to be brought down to proceed with the PCI OLR operation or the CRA.

All vPars and VM guests running on a v6.3 or later VSP must have the corresponding VirtualBase bundle installed to take advantage of the PCI OLR capability on the VSP.

The guest has more than 32 NPIV HBAs that are backed by the I/O card that are considered for replacement. PCI OLR is not supported on guests having more than 32 devices backed by the I/O card that are considered for replacement.

OR

Guest Instance (Guest ID) vswitch LanX: The Guest has more than 32 vNICs that are backed by the IO card being considered for replacement. PCI OLR is not supported on guests having more than 32 devices backed by I/O card that are considered for replacement.

A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars or VM guests have more than 32 impacted IO devices. The operation can only be performed after bringing down the vPar or VM guest.

You must ensure that no guest has more than 32 NPIV HBAs or more than 32 AVIO LAN interfaces backed by the same I/O card on the VSP to take advantage of the PCI OLR capability on the VSP.

The guest user space daemon is not running; see the HPVM documentation for possible workarounds

Starting with vPars and Integrity VM v6.3, a new HPVM guest user space daemon (`/opt/hpvm/bin/hpvmgud)` is delivered as part of the VirtualBase bundle. The daemon gets launched as part of the guest boot, and is a prerequisite for performing a CRA on the guest.

A PCI OLR operation or a CRA on the VSP is not supported if the HPVM guest user space daemon is not running or is unresponsive in any of the active vPars or VM guests with impacted IO.

The following checks and actions can be performed to ensure that the HPVM guest user space daemon is operational:

- Check all the process on the guest OS and see that `hpvmgud` is running (`ps -aef | egrep -i "hpvmgud"`
- View syslog for guest user daemon log messages.
- Remove the `hpvm` entry from `/etc/inittab` and run `/sbin/init q` and then start the guest user daemon manually by running `/opt/hpvm/bin/hpvmgud` without any options.

The guest CRA timed out, see HPVM documentation for possible workarounds.

For CRA timed out error and possible workaround, see Section (page 200).

The guest CRA failed due to an HPVM internal error, check HPVM documentation for possible workarounds.

This error is issued when the guest is in a state where PCI OLR components on the VSP are unable to communicate with the necessary components within the guest. This includes all guest states when boot or shutdown is in progress. In these states, several guest daemons and kernel operations will not be able to operate reliably.

The solution is to retry the PCI OLR operation on VSP when all guests are in a stable state - either not running at all or has completed start up operations.

Could not suspend the driver in one of the guests.

Error: prep_replace:/usr/sbin/olrad.d/hpvmdio driver script Failed !

A PCI OLR Suspend operation failed on the VSP in at least one of the guests. Try the PCI OLR Suspend operation once again on the same device on the VSP.

Could not resume the driver in one of the guests.

Error: post_replace:/usr/sbin/olrad.d/hpvmdio driver script Failed !

A PCI OLR Resume operation failed on the VSP in at least one of the guests. To recover the state of the card or device on the VSP and the guests, Hewlett Packard Enterprise recommends to perform a PCI OLR suspend followed by a PCI OLR Resume operation on the same device on the VSP.

Examples of PCI OLR operations

This section shows the results of PCI OLR operations on the VSP when there are active guests using some of the VSP I/O cards.

Example 46 Configuration

The VSP has two active VM guests configured with NPIV HBAs.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine   VM #  Type  OS Type State #VCPUs #Devs #Nets Memory
Name
================= ==== ===== ======= ===== ====== ===== ===== =======
guest1   1     SH   HPUX    On(OS) 32     2     1     256 GB
guest2   2     SH   HPUX    On(OS)  2     1     1       8 GB
```

The VSP has two dual port FC cards in PCI OLR capable slots.

```
# ioscan -kfNC fc
Class I H/W Path       Driver  S/W State  H/W Type   Description
======= ============== ======  =========  =========  ===========
fc 1    41/0/2/0/0/0/0  fcd     CLAIMED    INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
Port 1)
fc 2    41/0/2/0/0/0/1  fcd     CLAIMED    INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
Port 2)
fc 4    42/0/1/0/0/0/0  fcd     CLAIMED    INTERFACE  HP SN1000Q 16Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 1)
fc 5    42/0/1/0/0/0/1  fcd     CLAIMED    INTERFACE  HP SN1000Q 16Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 2)
```

FC cards are in the following OLR capable slots:

```
# olrad -q
Slot          Path            Link  Max   Max    Link  Pwr   Occu  Susp OLAR OLD  Mode
                              Spd   Link  Link   Width
                                    Spd   Width
==========   =============   ====  ===== =====  ====  ====  ====  ==== ==== ==== ====
9-0-2-2-0-1  41/0/2/0/0/0    5.0   5.0   x8     x4    On    Yes   No   Yes  Yes  PCIe
10-0-1-1-0-5 42/0/1/0/0/0    5.0   5.0   x8     x8    On    Yes   No   Yes  Yes  PCIe
```

The first guest has its NPIV HBAs backed by the FCD port `/dev/fcd1` and `/dev/fcd4` and the second guest has its NPIV HBA backed by the FC ports `/dev/fcd4`.

```
# hpvmstatus -P guest1 | grep hba
hba   avio_stor    0   0   npiv /dev/fcd4 -0x50014C2000000000,0x50014C2800000000
hba   avio_stor    0   2   npiv /dev/fcd1 -0x50014C2000000002,0x50014C2800000002

# hpvmstatus -P guest2 | grep hba
hba   avio_stor    0   0   npiv /dev/fcd4 -0x50014C2000000001,0x50014C2800000001
```

The VM guest guest1 has a boot disk with two paths passing through `/dev/fcd1` and `/dev/fcd4` on the VSP.

```
guest1# ioscan -kfNd gvsd
Class I  H/W Path    Driver S/W State  H/W Type   Description
=======  ==========  ====== =========  =========  ======================
ext_bus  0  0/0/0/0  gvsd   CLAIMED    INTERFACE  HPVM NPIV Stor Adapter
ext_bus  2  0/0/2/0  gvsd   CLAIMED    INTERFACE  HPVM NPIV Stor Adapter

guest1# hpvmdevinfo
Device  Bus,Device,Target  Backing Store Host Device Virtual Machine
Type                       Type          Name        Device Name
======  =================  ============= =========== ===============
hba     [0,0]              npiv          /dev/fcd4    /dev/gvsd0
hba     [0,2]              npiv          /dev/fcd1    /dev/gvsd2

guest1# setboot

Primary bootpath : 0/0/2/0.0x22540002ac000d2c.0x4001000000000000 (/dev/rdisk/disk0)→  Boot disk
HA Alternate bootpath :
Alternate bootpath :

guest1# ioscan -m lun /dev/rdisk/disk0
Class  Lun  H/W Path          Driver  S/W State  H/W Type  Health Description
I
====== ==== ==============    ======  =========  ========  ====== ===========
disk   0    64000/0xfa00/0x1  esdisk  CLAIMED    DEVICE    online 3PARdataVV
            0/0/2/0.0x22540002ac000d2c.0x4001000000000000
             0/0/0/0.0x21530002ac000d2c.0x4001000000000000 ----> Boot disk has two paths
```

```
            /dev/disk/disk0      /dev/disk/disk0_p2   /dev/rdisk/disk0     /dev/rdisk/disk0_p2
            /dev/disk/disk0_p1   /dev/disk/disk0_p3   /dev/rdisk/disk0_p1  /dev/rdisk/disk0_p3
```

The VM guest guest2 has a boot disk with a single path passing through `/dev/fcd4` on the VSP.

```
guest2# ioscan -kfNd gvsd
Class I  H/W Path    Driver S/W State  H/W Type   Description
=======  ==========  ====== =========  =========  ======================
ext_bus  0  0/0/0/0  gvsd   CLAIMED    INTERFACE  HPVM NPIV Stor Adapter
```

```
guest2# hpvmdevinfo
Device  Bus,Device,Target  Backing Store Host Device Virtual Machine
Type                       Type          Name        Device Name
======  =================  ============= =========== ===============
hba     [0,0]              npiv/dev/fcd4 /dev/gvsd0

guest2# setboot
Primary bootpath : 0/0/0/0.0x21530002ac000d2c.0x4001000000000000(/dev/rdisk/disk0)-------------> Boot disk
HA Alternate bootpath :
Alternate bootpath :

guest2# ioscan -m lun /dev/rdisk/disk0
Class  Lun  H/W Path          Driver  S/W State  H/W Type  Health Description
I
====== ==== ==============    ======  =========  ========  =================
disk   0    64000/0xfa00/0x1  esdisk  CLAIMED    DEVICE    online 3PARdataVV
       0/0/0/0.0x21530002ac000d2c.0x4001000000000000 -------------> Boot disk has a single path
            /dev/disk/disk0      /dev/disk/disk0_p2   /dev/rdisk/disk0     /dev/rdisk/disk0_p2
            /dev/disk/disk0_p1   /dev/disk/disk0_p3   /dev/rdisk/disk0_p1  /dev/rdisk/disk0_p3
```

OLR Operation:

To do an online replacement of the card in slot 10-0-1-1-0-5 (/dev/fcd4 and /dev/fcd5), first run the `olrad` command to determine the criticality of the resource.

```
# olrad -C 10-0-1-1-0-5
Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]
CRA REPORT SUMMARY: CRA detected SYSTEM CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.
```

The CRA reports SYSTEM CRITICAL severity, for more information about SYSTEM CRITICAL severity, see the CRA log. The following is the snippet from the CRA log on the VSP containing the NPIV analysis:

```
ANALYSIS SCOPE: HPVM NPIV
This report provides details of any critical NPIV hardware path usages in the system.

RESULT: SYSTEM CRITICAL NPIV resources will be affected
DETAILED HPVM NPIV CRA REPORT

SYSTEM CRITICAL RESULTS
 Affected vPars or VM guest are:
  2 -----------> VM guest guest2
DATA CRITICAL RESULTS
  Affected vPars or VM guest are:  NONE
WARNINGS
  Affected vPars or VM guest are:
  1 -----------> VM guest guest1
FAILURE
  vPars or VM guests that failed to perform CRA analysis are: NONE
```

The CRA logs inside VM guests `guest1` and `guest2` have the additional information about why the CRA analysis within the VM guests reported WARNING and SYSTEM CRITICAL respectively.

Example 47 Configuration

The configuration is same as for Example 1, with the exception that the VM guest `guest2` is shut down.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State  #VCPUs #Devs #Nets Memory
==================== ====  ==== ======= =====  ====== ===== ===== =======
guest1            1     SH   HPUX    On(OS)  32     2     1    256 GB
guest2            2     SH   HPUX    Off      2     1     1      8 GB
```

As in the previous example, the NPIV boot disk of the `guest1` has two paths, one through `/dev/fcd1` and the other through `/dev/fcd4`.

OLR Operation:

Run the `olrad -C` command to determine if the card in slot 10-0-1-1-0-5 can be replaced:

```
# olrad -C 10-0-1-1-0-5
Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA returned WARNING.
Detailed CRA report is available in /var/adm/cra.log file.
```

The following is the snippet from the CRA log on the VSP:

```
ANALYSIS SCOPE: HPVM NPIV
This report provides details of any critical NPIV hardware path usages in the system.

RESULT: WARNING Some hardware paths to resources are affected
DETAILED HPVM NPIV CRA REPORT

SYSTEM CRITICAL RESULTS
  Affected vPars or VM guest are: NONE
DATA CRITICAL RESULTS
  Affected vPars or VM guest are: NONE

WARNINGS
   Affected vPars or VM guest are: 1 -------> VM guest guest1

FAILURE
 vPars or VM guests that failed to perform CRA analysis are: NONE
```

Now, you can go ahead and replace the card in slot 10-0-1-1-0-5. As the first step, run the `olrad -r` command to prepare the IO card for removal.

```
# olrad -r 10-0-1-1-0-5
Activity: Start of Prepare Replace
Target slot: 10-0-1-1-0-5

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA returned WARNING.
Detailed CRA report is available in /var/adm/cra.log file.

CRA output: resources in use on affected device(s)
Target slot: 10-0-1-1-0-5

Activity: End of Prepare Replace
Target slot: 10-0-1-1-0-5

Activity: Target slot powered off, drivers suspended, OK to replace the card
Target slot: 10-0-1-1-0-5
```

Now, the IO card on the VSP is suspended.

```
# ioscan -kfNC fc
Class I H/W Path            Driver  S/W State  H/W Type   Description
======= ===============     ======  =========  =========
==============================================================
fc 1    41/0/2/0/0/0/0      fcd     CLAIMED    INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter
(FC Port 1)
fc 9    41/0/2/0/0/0/0.0x11  fcd    CLAIMED    INTERFACE  HPVM Virtual FC (VFC) Controller
fc 2    41/0/2/0/0/0/1      fcd     CLAIMED    INTERFACE  HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter
(FC Port 2)
fc 4    42/0/1/0/0/0/0      fcd     SUSPENDED  INTERFACE  HP SN1000Q 16Gb Dual Port PCIe Fibre Channel Adapter
 (FC Port 1)
fc 8    42/0/1/0/0/0/0.0x11  fcd    SUSPENDED  INTERFACE  HPVM Virtual FC (VFC) Controller
fc 5    42/0/1/0/0/0/1      fcd     SUSPENDED  INTERFACE  HP SN1000Q 16Gb Dual Port PCIe Fibre Channel Adapter
 (FC Port 2)
```

And, the targets and LUNs impacted in `guest1` are gone to NO_HW state.

```
guest1# ioscan -kfNH 0/0/0/0
Class I   H/W Path                                       Driver  S/W State  H/W Type    Description
========  ===============================================  ======  =========  =========
================================================
ext_bus   0  0/0/0/0                                   gvsd    CLAIMED    INTERFACE   HPVM NPIV Stor Adapter
tgtpath   0  0/0/0/0.0x21530002ac000d2c                  estp    NO_HW      TGT_PATH    Virtual Storage HBA
 target served by gvsd driver,target port id 0x10200
lunpath   0  0/0/0/0.0x21530002ac000d2c.0x0              eslpt   NO_HW      LUN_PATH    LUN path for ctl0
lunpath   1  0/0/0/0.0x21530002ac000d2c.0x4001000000000000  eslpt   NO_HW      LUN_PATH    LUN path for disk0
```

After the IO card is replaced with a new one, the `olrad -R` command can be used to resume usage of the card.

```
# olrad -R 10-0-1-1-0-5
Activity: Start of Post Replace
Target slot: 10-0-1-1-0-5

Activity: End of Post Replace
Target slot: 10-0-1-1-0-5

Activity: Target slot powered on, drivers resumed, OK to start using the card
Target slot: 10-0-1-1-0-5
```

At this point, the NPIV HBA and the targets and LUNs behind it are ready for use and come back to the CLAIMED state.

```
guest1# ioscan -kfNH 0/0/0/0
Class I   H/W Path                                       Driver  S/W State  H/W Type    Description
========  ===============================================  ======  =========  =========
================================================
ext_bus   0  0/0/0/0                                   gvsd    CLAIMED    INTERFACE   HPVM NPIV Stor Adapter
tgtpath   0  0/0/0/0.0x21530002ac000d2c                  estp    CLAIMED    TGT_PATH    Virtual Storage HBA
 target served by gvsd driver,target port id 0x10200
lunpath   0  0/0/0/0.0x21530002ac000d2c.0x0              eslpt   CLAIMED    LUN_PATH    LUN path for ctl0
lunpath   1  0/0/0/0.0x21530002ac000d2c.0x4001000000000000  eslpt   CLAIMED    LUN_PATH    LUN path for disk0
```

Examples

This section shows the results of PCI OLR operations on the VSP when there are active guests using some of the VSP I/O cards.

Example 48 Configuration

An SD2 system configured as VSP running vPars and Integrity VM v6.3 or later with two active guests and the system has a dual ported NIC supporting DLA.

```
# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State   #VCPUs #Devs #Nets Memory
==================== ====  ==== ======= =====   ====== ===== ===== =======
guest1               1     SH   HPUX    On (OS)  1      1     3     2 GB
guest3               3     SH   HPUX    On (OS)  2      1     1     2 GB
```

Dual ported NIC supporting DLA is residing in an OL* capable slot and configured to DIO pool.

```
# hpvmhwmgmt -p dio -l

                                                Assignment
H/W Path         Class Owner Description        Level       Label
==============   ===== ===== ===============    =========== ======
42/0/0/2/0/0/0   lan   hpvm  HP AM225-60001     PCIe 2-p 1  device
42/0/0/2/0/0/1   lan   hpvm  HP AM225-60001     PCIe 2-p 1  device
```

The slot details of the DLA NIC is as follows:

```
# olrad -q

Slot          Path          Link Max      Max     Link     Pwr   Occu  Susp OLAR OLD  Mode
                            Spd  Link     Link    Width
                                 Spd      Width
10-0-1-0-2-3 42/0/0/2/0/0 5.0   5.0      x8      x8        On    Yes   No  Yes  Yes  PCIe

# ioscan -kfH 42/0/0/2/0/0

Class    I   H/W Path        Driver   S/W State  H/W Type   Description
=======  === ==============  ======== =========  =========  ============================
slot     15  42/0/0/2/0/0    pci_slot CLAIMED    SLOT       PCI Slot
hpvmdio   0  42/0/0/2/0/0/0  hpvmdio  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
hpvmdio   1  42/0/0/2/0/0/1  hpvmdio  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
```

Both ports of the DLA NIC are assigned to an active VM guest with guest ID **1**.

```
# hpvmdevinfo -M | grep dio

pqsbuc03:guest1:1:lan:dio:1;5;0x2E1B47D73CA1:hwpath:42/0/0/2/0/0/0:0/1/5/0 (lan8)
pqsbuc03:guest1:1:lan:dio:2;2;0xAE8CD62ED123:hwpath:42/0/0/2/0/0/1:0/2/2/0 (lan17)
```

In the VM guest **1**, the DIO functions (ports) are seen as lan8 (path 0/1/5/0) and lan17 (path 0/2/2/0) and IP addresses are configured for both the ports.

```
# ioscan -kfnC lan

Class I   H/W Path  Driver  S/W State H/W Type   Description
===== ==  ========  ======= ========= =========  ==========================================
lan    8  0/1/5/0   iexgbe  CLAIMED   INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan   17  0/2/2/0   iexgbe  CLAIMED   INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter

# netstat -in

Name   Mtu   Network   Address         Ipkts  Ierrs Opkts Oerrs Coll
lan17  1500  15.0.0.0  15.213.156.43   4      0     4     0     0
lan8   1500  15.0.0.0  15.213.156.42   6      0     6     0
```

If you want to replace the NIC with another card of same model, you must initially run the `olrad(1M)` command with the `-C` option which reports the resource usage and its criticality. In this example, DLA NIC is residing in slot “10-0-1-0-2-3” and running `olrad -C` on this slot yields the following output:

```
# olrad -C 10-0-1-0-2-3

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA detected DATA CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.
```

The criticality reported by CRA in this case is CRA_DATA_CRITICAL and for more information about the CRA, see the CRA log file `/var/adm/cra.log` on the VSP. The following is the snippet from the CRA log on the VSP containing the HPVM Direct I/O Analysis details.

```
ANALYSIS SCOPE:HPVM- DIRECT I/O
This report provides details of any critical Direct I/O
hardware path usages in the system.

RESULT: DATA CRITICAL Direct I/O resources will be affected.

SYSTEM CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: NONE

DATA CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: 1

WARNING
 Affected vPars or VM guest instances numbers are: NONE

FAILURE
 vPars or VM guests that failed to perform CRA analysis are: NONE
```

The CRA that logs inside the guest have additional information about the DATA critical usage inside the guest. The following is a snippet from the CRA log in the guest for this specific scenario.

```
ANALYSIS SCOPE: NETWORKING
This report provides details of any networking related usages for
a set of h/w paths in the system.
RESULT: DATA-CRITICAL resource usage detected.
DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
0/1/5/0 (lan8)
0/2/2/0 (lan17)

DATA CRITICAL RESULTS:
Interface lan8:  IPv4 address 15.213.156.42
Interface lan17: IPv4 address 15.213.156.43

USEFUL NETWORKING COMMANDS:
lanadmin  lanscan  nwmgr  netstat  ifconfig linkloop
```

In this scenario, where the CRA has returned DATA CRITICAL, if you choose to run the `Pre Replace` option of the `olrad (1M)` command (`olrad -r` option), the operation fails with CRA_DATA_CRITICAL error, reason being that doing this operation renders the VM guest where the DLA NIC ports are assigned, inaccessible to network.

However, if you want to override the CRA criticality report, you can do so on your own risk by using the force (`-f`) option of the `olrad` command. If any of these operations must be performed for system administration purposes, then you must re-assign all the IP addresses configured on each of the ports of the NIC to be replaced to alternative NIC cards.

The following is the sample output of the `olrad(1M)` command when run with the force (`-f`) option:

```
# olrad -f -r 10-0-1-0-2-3

Activity: Start of Prepare Replace
Target slot: 10-0-1-0-2-3
Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA detected DATA CRITICAL usages.
Detailed CRA report is available in /var/adm/cra.log file.
CRA Error     : resources associated with possible data loss
Target slot   : 10-0-1-0-2-3
Activity      : CRA being forced using -f option
Target slot   : 10-0-1-0-2-3
Activity      : End of Prepare Replace
Target slot   : 10-0-1-0-2-3
Activity      : Target slot powered off, drivers suspended, OK to replace the card
Target slot   : 10-0-1-0-2-3
```

To verify that the DLA NIC is successfully suspended, you can use the following options of the `olrad(1M)` command and `ioscan(1M)` command.

```
# olrad -q

Slot         Path         Link  Max   Max    Link   Pwr Occu Susp OLAR  OLD  Mode
                          Spd   Link  Link   Width
                                Spd   Width
10-0-1-0-2-3 42/0/0/2/0/0 5.0   5.0   x8     x8     On  Yes  Yes  Yes   Yes  PCIe

# ioscan -kfnC hpvmdio

Class   I  H/W Path        Driver  S/W State  H/W Type   Description
======= == ==============  ======= ========== =========  ========================
hpvmdio 0  42/0/0/2/0/0  hpvmdio SUSPENDED  INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
             /dev/hpvmdio0
hpvmdio 1  42/0/0/2/0/1  hpvmdio SUSPENDED  INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
            /dev/hpvmdio1
```

Further, ioscan output inside the guest will also show the DLA NIC port in SUSPENDED state.

```
# ioscan -kfnC lan

Class I   H/W Path  Driver  S/W State  H/W Type  Description
===== === ========  ======= =========  ========= ==========================================
lan   8   0/1/5/0   iexgbe  SUSPENDED  INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan   17  0/2/2/0   iexgbe  SUSPENDED  INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
```

After the DLA NIC is replaced with a new one, you can use the `Post Replace` option of the `olrad(1M)` command (`olrad -R`) to resume usage of the card. The following is the sample output of the `olrad -R` command.

```
# olrad -R 10-0-1-0-2-3

Activity: Start of Post Replace
Target slot: 10-0-1-0-2-3

Activity: End of Post Replace
Target slot: 10-0-1-0-2-3

Activity: Target slot powered on, drivers resumed, OK to start using the card
Target slot: 10-0-1-0-2-3
```

To verify that the DLA NIC is successfully resumed, you can use the following options of the `olrad(1M)` command and `ioscan(1M)` command.

```
# olrad -q

Slot         Path         Link Max  Max   Link  Pwr  Occu  Susp OLAR  OLD  Mode
                          Spd  Link Link  Width
                               Spd  Width
10-0-1-0-2-3 42/0/0/2/0/0 5.0  5.0  x8    x8    On   Yes   No   Yes   Yes  PCIe

# ioscan -kfnC hpvmdio
```

```
Class    I  H/W Path        Driver   S/W State  H/W Type  Description
======= == ============== =======  =========  ========= ========================
hpvmdio 0  42/0/0/2/0/0/0  hpvmdio  CLAIMED    INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
              /dev/hpvmdio0
hpvmdio 1  42/0/0/2/0/0/1  hpvmdio  CLAIMED    INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
              /dev/hpvmdio1
```

Further, the ioscan output inside the guest will also show the DLA NIC port in CLAIMED state indicating that the NIC port is successfully resumed.

```
# ioscan -kfnC lan

Class I   H/W Path  Driver  S/W State  H/W Type   Description
===== === ========= ======= =========  =========  =========================
lan    8  0/1/5/0   iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan   17  0/2/2/0   iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
```

Example 49 Configuration

An SD2 system configured as VSP running vPars and Integrity VM v6.3 or later, with two active guests and the system has a dual ported NIC supporting FLA, each port of the FLA NIC assigned to two different active guest.

```
# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State  #VCPUs #Devs #Nets Memory
==================== ====  ==== ======= =====  ====== ===== ===== =======
guest1               1     SH   HPUX    On(OS)  1     1     3     2 GB
guest3               3     SH   HPUX    On(OS)  2     1     1     2 GB
```

Dual ported NIC supporting DLA is residing in an OL* capable slot and configured to DIO pool.

```
# hpvmhwmgmt -p dio -l
                                               Assignment
H/W Path        Class Owner Description        Level       Label
=============== ===== ===== ================== =================
40/0/1/0/0/0/0  lan   hpvm  HP AT118A 2p 10GbE PCIe A function
40/0/1/0/0/0/1  lan   hpvm  HP AT118A 2p 10GbE PCIe A function
```

The slot details of the DLA NIC is as follows:

```
# olrad -q
Slot          Path          Link  Max       Max      Link      Pwr   Occu  Susp OLAR OLD  Mode
                            Spd   Link      Link     Width
                                  Spd       Width
9-0-1-1-0-5 40/0/1/0/0/0  5.0   5.0       x8       x8         On    Yes   No  Yes  Yes  PCIe

# ioscan -kfnC hpvmdio
Class    I   H/W Path        Driver   S/W State  H/W Type   Description
=======  === =============   ======== ========= =========   ============================
hpvmdio   5  40/0/1/0/0/0/0  hpvmdio  CLAIMED   INTERFACE   HP AT118A 2p 10GbE PCIe Adapter
               /dev/hpvmdio5
hpvmdio   1  40/0/1/0/0/0/1  hpvmdio  CLAIMED   INTERFACE   HP AT118A 2p 10GbE PCIe Adapter
               /dev/hpvmdio1
```

Each port of the FLA NIC is assigned to two different active guest (Guest id **1** and Guest id **3**) with each configured with IP address and connected to network.

```
# hpvmdevinfo -M | grep dio
palace1:guest1:1:lan:dio:0;5;0xF688359C7B15:hwpath:40/0/1/0/0/0/0:0/0/5/0 (lan1)
palace1:guest3:3:lan:dio:0;5;0x96A442D65C83:hwpath:40/0/1/0/0/0/1:0/0/5/0 (lan1)
```

In the VM guest 1 and VM guest 3, the DIO functions (port) is seen as lan1 (path 0/0/5/0) and IP addresses are configured for both the ports.

On VM guest 1

```
# ioscan -kfnC lan
Class I  H/W Path  Driver   S/W State  H/W Type  Description
===== == ========= =======  =========  ========= ================================
lan   1  0/0/5/0   iocxgbe  CLAIMED    INTERFACE HP AT118A 2p 10GbE PCIe Adapter

# netstat -in
Name  Mtu   Network  Address         Ipkts  Ierrs Opkts Oerrs Coll
an1   1500  15.0.0.0 15.213.153.214  8      0     8     0     0
```

On VM guest 3

```
# ioscan -kfnC lan
Class I  H/W Path  Driver  S/W State  H/W Type  Description
===== == ========  ======= =========  ========= ================================
lan   1  0/0/5/0   iocxgbe CLAIMED    INTERFACE HP AT118A 2p 10GbE PCIe Adapter

# netstat -in
```

```
Name  Mtu   Network   Address         Ipkts  Ierrs Opkts Oerrs Coll
lan1  1500  15.0.0.0  15.213.153.220  24     0     24    0     0
```

If you want to replace the NIC in slot 10-0-1-0-2-3 with another card of same model, you must initially run the `olrad(1M)` command with `-C` option which reports the resource usage and its criticality.

In this example, as the FLA NIC is having IP configured, the criticality reported by CRA in this case will be CRA_DATA_CRITICAL and for more information about the CRA details, see the CRA log file `/var/adm/cra.log` on the VSP. The following is the snippet from the CRA log on the VSP containing the HPVM Direct I/O Analysis details.

```
ANALYSIS SCOPE: HPVM- DIRECT I/O
This report provides details of any critical Direct I/O hardware path usages in the system.

RESULT: DATA CRITICAL Direct I/O resources will be affected.

SYSTEM CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: NONE

DATA CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: 1 3

WARNING
 Affected vPars or VM guest instances numbers are: NONE

FAILURE
Vpars or VM guests that failed to perform CRA analysis are: NONE
```

The CRA that logs inside the guest have additional information about the DATA critical usage inside the guests.

In this scenario, where the CRA has returned DATA CRITICAL, if you choose to run the `Pre Replace` option of the `olrad (1M)` command (`olrad -r` option), as in example 1, the operation fails with CRA_DATA_CRITICAL error, reason being that doing this operation renders each of the VM guests (Guest 1 and Guest 3) VM where the FLA NIC ports are assigned, inaccessible to network.

As in example 1, if you want to override the CRA criticality report, you can do so on your own risk by using the force (`-f`) option of the `olrad` command. The `olrad(1M)` and `ioscan(1M)` commands can be used to verify that the FLA NIC is successfully suspended.

Further, ioscan output inside each of the guest (Guest 1 and Guest 3) will also show the corresponding FLA NIC port in SUSPENDED state.

On VM guest 1

```
# ioscan -kfnC lan
Class I   H/W Path  Driver  S/W State  H/W Type  Description
===== === ========  ======= =========  ========= ===============================
lan   1   0/0/5/0   iocxgbe SUSPENDED  INTERFACE HP AT118A 2p 10GbE PCIe Adapter
```

On VM guest 3

```
# ioscan -kfnC lan
Class I   H/W Path  Driver  S/W State  H/W Type  Description
===== === ========  ======= =========  ========= ===============================
lan   1   0/0/5/0   iocxgbe SUSPENDED  INTERFACE HP AT118A 2p 10GbE PCIe Adapter
```

After the FLA NIC is replaced with a new one, the `Post Replace` option of the `olrad(1M)` command (`olrad -R`) can be used to resume usage of the card. The `olrad(1M)` and `ioscan` commands can be used to verify that the FLA NIC is successfully resumed.

Example 50 Configuration

An SD2 system configured as VSP running vPars and Integrity VM v6.3 or later with two active guests, and the system has two dual ported NIC, one supporting DLA and the other supporting FLA. DLA and FLA ports are further configured in APA mode for redundancy.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State   #VCPUs #Devs #Nets Memory
==================== ====  ==== ======= =====   ====== ===== ===== =======
pvham1               2     VP   HPUX    On(OS)  3      2     23    22528 MB
pvoly2d              5     SH   HPUX    On(OS)  3      6     18        5 GB
```

The system has two Dual ported NIC, one supporting DLA and the other supporting FLA and is residing in an OL* capable slots and configured to DIO pool.

```
# hpvmhwmgmt -p dio -l
                                                   Assignment
H/W Path        Class Owner Description            Level       Label
==============  ===== ===== ==================     =================
42/0/0/2/0/0/0  lan   hpvm  HP AM225-60001         PCIe 2-p 1 device
42/0/0/2/0/0/1  lan   hpvm  HP AM225-60001         PCIe 2-p 1 device
43/0/1/0/0/0/0  lan   hpvm  HP AT111-60001 10Gb PCIe        function
43/0/1/0/0/0/1  lan   hpvm  HP AT111-60001 10Gb PCIe        function
```

The slot details of the DLA and FLA NIC are as follows:

```
# olrad -q
Slot          Path          Link  Max   Max    Link   Pwr  Occu  Susp OLAR OLD  Mode
                            Spd   Link  Link   Width
                                  Spd   Width
10-0-1-0-2-3  42/0/0/2/0/0  5.0   5.0   x8     x8     On   Yes   No   Yes  Yes  PCIe
10-0-2-1-0-5  43/0/1/0/0/0  5.0   5.0   x8     x8     On   Yes   No   Yes  Yes  PCIe

# ioscan -kfH 42/0/0/2/0/0
Class    I   H/W Path        Driver   S/W State  H/W Type   Description
=======  === ==============  ======== ========== =========  ============================
slot     15  42/0/0/2/0/0    pci_slot CLAIMED    SLOT       PCI Slot
hpvmdio   0  42/0/0/2/0/0/0  hpvmdio  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
hpvmdio   1  42/0/0/2/0/0/1  hpvmdio  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter

# ioscan -kfH 43/0/1/0/0/0
Class    I   H/W Path        Driver   S/W State  H/W Type   Description
=======  === ==============  ======== ========== =========  ============================
slot     22  43/0/1/0/0/0    pci_slot CLAIMED    SLOT       PCI Slot
hpvmdio   3  43/0/1/0/0/0/0  hpvmdio  CLAIMED  INTERFACE    HP AT111-60001 10Gb PCIe 2-port
CNA (NIC) Adapter
hpvmdio   11  43/0/1/0/0/0/1  hpvmdio    CLAIMED     INTERFACE    HP AT111-60001 10Gb PCIe 2-port
CNA (NIC) Adapter
```

Both ports of DLA NIC and one port of FLA NIC are assigned to an active VM guest with guest id 5.

```
# hpvmdevinfo -M | grep dio
pqsbuc03:pvoly2d:5:lan:dio:1;5;0x2E1B47D73CA1:hwpath:42/0/0/2/0/0/0:0/1/5/0 (lan8)
pqsbuc03:pvoly2d:5:lan:dio:2;2;0xAE8CD62ED123:hwpath:42/0/0/2/0/0/1:0/2/2/0 (lan17)
pqsbuc03:pvoly2d:5:lan:dio:2;5;0xBA3462833C28:hwpath:43/0/1/0/0/0/1:0/2/5/0 (lan20)
```

In the VM guest 5, the DIO functions (ports) are seen as lan8 (path 0/1/5/0), lan17 (path 0/2/2/0) and lan20 (path 0/2/5/0).

```
# ioscan -kfnC lan
Class I  H/W Path  Driver   S/W State  H/W Type  Description
===== == ========= =======  =========  ========= ==============================
lan    8  0/1/5/0   iexgbe   CLAIMED    INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan   17  0/2/2/0   iexgbe   CLAIMED    INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan   20  0/2/5/0   iocxgbe  CLAIMED    INTERFACE HP AT111-60001 10Gb PCIe 2-port CNA (NIC)Adapter
```

In the VM guest 5, lan8 and lan20 are further configured as APA and assigned to lan900.

```
# nwmgr
Name          Interface  Station        Sub-     Interface    Related
```

```
ClassInstance State      AddressI       System  Type         Interface
============= =========  ============== ======= ==========   =================
lan8          UP         0x2E1B47D73CA1 iexgbe  10GBASE-SR   lan900
lan900        UP         0x2E1B47D73CA1 hp_apa   hp_apa
lan20         UP         0xBA3462833C28 iocxgbe  10GBASE-SFP lan900

# nwmgr -S apa -I 900 -v
lan900 current values:
 Mode = LAN_MONITOR
 Parent PPA =  -
 APA State = Up
 Membership = 8,20
 Active Port(s) = 8
 Ready Port(s) = 20
 Not Ready Port(s) = —
 Connected Port(s) = 20
 Polling Interval = 10000000
```

If you want to replace the NIC with another card of same model, you must initially run the `olrad(1M)` command with `-C` option which reports the resource usage and its criticality.

The criticality reported by CRA in this case is WARNINGS and for more information about the CRA details, see the CRA log file `/var/adm/cra.log` on the VSP. The following is the snippet from the CRA log on the VSP containing the HPVM Direct I/O Analysis details.

```
ANALYSIS SCOPE: HPVM- DIRECT I/O
This report provides details of any critical Direct I/O hardware path usages in the system.

RESULT: WARNING Some hardware paths to resources are affected.

SYSTEM CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: NONE

DATA CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: NONE

WARNING
 Affected vPars or VM guest instances numbers are: 5

FAILURE
 vPars or VM guests that failed to perform CRA analysis are: NONE
```

The CRA that logs inside the guest have additional information about the WARNINGS usage inside the guest. The following is a snippet from the CRA log in the guest for this specific scenario.

```
ANALYSIS SCOPE: NETWORKING
This report provides details of any networking related usages for
a set of h/w paths in the system.

RESULT: WARNING resources usage detected.
DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
0/1/5/0 (lan8)
0/2/2/0 (lan17)

WARNINGS:
 Auto Port Aggregation(APA) Usage:
  Aggregate lan900 will get impacted by the intended operation on
  the member links listed below: lan8
 NOTE: Intended operation might compromise high availability provided by APA.

USEFUL NETWORKING COMMANDS:
lanadmin  lanscan  nwmgr  netstat  ifconfig linkloop
```

In this scenario, where the CRA has returned WARNINGS, if you choose to run the `Pre Replace` option of the `olrad (1M)` command (`olrad -r` option), on DLA or FLA NIC, the operation succeeds as it does not cause loss of services.

```
# olrad -r 10-0-1-0-2-3
Activity      : Start of Prepare Replace
Target slot   : 10-0-1-0-2-3

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA returned WARNING.
Detailed CRA report is available in /var/adm/cra.log file.

CRA output    : resources in use on affected device(s)
Target slot   : 10-0-1-0-2-3

Activity      : End of Prepare Replace
Target slot   : 10-0-1-0-2-3

Activity      : Target slot powered off, drivers suspended, OK to replace the card
Target slot   : 10-0-1-0-2-3
```

To verify that the DLA NIC is successfully suspended on host and guest, options of the `olrad(1M)` and `ioscan(1M)` commands can be used as in previous examples.

Further, nwmgr output inside the guest will also show that the active port of APA has changed.

In this example, nwmgr output shows lan8 has gone down and lan20 is now the active port in APA.

```
# nwmgr -S apa -I 900 -v
lan900 current values:
 Mode = LAN_MONITOR
 Parent PPA =  -
 APA State = Up
 Membership = 20,8*
 Active Port(s) = 20
 Ready Port(s) =  -
 Not Ready Port(s) = 8
 Connected Port(s) =  -
 Polling Interval = 10000000
 Dead Count = 3
```

After the DLA NIC is replaced with a new one, the `Post Replace` option of the `olrad(1M)` command (`olrad -R`) can be used to resume usage of the card. To verify that the DLA NIC is successfully suspended on the host and guest, the options of the `olrad(1M)` and `ioscan` commands can be used.

Example 51 Configuration

An SD2 system configured as VSP running vPars and Integrity VM v6.3 or later with two active guests, and the system has a Combo card supporting NIC (FLA) and FC functions.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State   #VCPUs #Devs #Nets Memory
==================== ====  ==== ======= =====   ====== ===== ===== =======
pvham1               2     VP   HPUX    On(OS)  3      2     23    22528 MB
pvoly2d              5     SH   HPUX    On(OS)  3      6     18        5 GB
```

The system has a Combo card supporting NIC (FLA) and FC, it is residing in an OL* capable slots and its ports are configured to DIO pool.

```
# hpvmhwmgmt -p dio -l
                                                           Assignment
H/W Path               Class Owner Description         Level    Label
====================== ===== ===== =================== ==============
43/0/2/2/0/0/0/8/0/0/0 lan   hpvm  HP AT094A 10GbE-SFP PCIe  function
43/0/2/2/0/0/0/8/0/0/1 lan   hpvm  HP AT094A 10GbE-SFP PCIe  function
```

The slot details of the FLA NIC are as follows:

```
# olrad -q
Slot          Path          Link  Max   Max    Link   Pwr  Occu  Susp OLAR OLD  Mode
                            Spd   Link  Link   Width
                                  Spd   Width
10-0-2-2-2-2  43/0/2/2/0/0  5.0   5.0   x8     x8     On   Yes   No   Yes  Yes  PCIe

# ioscan -kfNH 43/0/2/2/0/0/0
Class    I   H/W Path        Driver   S/W State  H/W Type   Description
=======  === ==============  ======== ========== =========  ============================
ba       44  43/0/2/2/0/0/0  PCItoPCI CLAIMED    BUS_NEXUS  PCItoPCI Bridge
ba       45  43/0/2/2/0/0/0/4/0 PCItoPCI   CLAIMED     BUS_NEXUS    PCItoPCI Bridge
ba       46  43/0/2/2/0/0/0/5/0 PCItoPCI   CLAIMED     BUS_NEXUS    PCItoPCI Bridge
fc        6  43/0/2/2/0/0/0/5/0/0/0 fclp       CLAIMED     INTERFACE    HP AT094-
60001 PCIe Fibre Channel 2-port 8Gb FC/2-port 10GBE Combo Adapter
fc       11  43/0/2/2/0/0/0/5/0/0/0.0x11 fclp       CLAIMED     INTERFACE    HPVM
Virtual FC (VFC) Controller
tgtpath   4  43/0/2/2/0/0/0/5/0/0/0.0x50001fe15000cd2a estp       CLAIMED     TGT_PATH
fibre_channel target served by fclp driver, target port id 0x31700
lunpath  24  43/0/2/2/0/0/0/5/0/0/0.0x50001fe15000cd2a.0x0 eslpt      CLAIMED     LUN_PATH
LUN path for ctl6
fc        7  43/0/2/2/0/0/0/5/0/0/1                        fclp       CLAIMED     INTERFACE    HP AT094-
60001 PCIe Fibre Channel 2-port 8Gb FC/2-port 10GBE Combo Adapter
ba       47  43/0/2/2/0/0/0/8/0                            PCItoPCI   CLAIMED     BUS_NEXUS    PCItoPCI
Bridge
hpvmdio   2  43/0/2/2/0/0/0/8/0/0/0                        hpvmdio    CLAIMED     INTERFACE    HP
AT094A 10GbE-SFP PCIe 2p 8Gb FC and 2p 1/10Gbe Adapter
hpvmdio   4  43/0/2/2/0/0/0/8/0/0/1                        hpvmdio    CLAIMED     INTERFACE    HP
AT094A 10GbE-SFP PCIe 2p 8Gb FC and 2p 1/10Gbe Adapter
```

Both ports of the FLA NIC are assigned to two different active VM guest `pvoly2d` and `phvam1`.

```
# hpvmdevinfo -M | grep dio
pqsbuc03:pvham1:2:lan:dio:2;2;0x5EFA2C6F1FB5:hwpath:43/0/2/2/0/0/0/8/0/0/0:0/0/2/2/0 (lan11)
pqsbuc03:pvoly2d:5:lan:dio:2;4;0x26D0A2CFE07B:hwpath:43/0/2/2/0/0/0/8/0/0/1:0/2/4/0 (lan19)
```

In the VM guest `pvoly2d`, the DIO functions (ports) are seen as lan19 (path 0/2/4/0) and IP address is configured for this port only.

```
# ioscan -kfnC lan
Class I  H/W Path  Driver   S/W State  H/W Type  Description
===== == ======== =======  =========  ========= ==============================
lan   19 0/2/4/0   iocxgbe  CLAIMED    INTERFACE HP AT094A 10GbE-SFP PCIe 2p 8Gb FC and 2p
1/10Gbe Adapter
```

```
# netstat -in
Name   Mtu  Network      Address       Ipkts Ierrs Opkts Oerrs  Coll
====== ==== ============ ============= ===== ===== ===== =====  ====
lan19  1500 15.0.0.0     15.213.156.42 4     0     4     0      0
```

```
lan3   1500 15.213.200.0 15.213.202.60 1060  0     171   0      0
lo0   32808 127.0.0.0    127.0.0.1     242   0     242   0      0
```

If you want to replace the NIC with another card of same model, you must initially run the `olrad(1M)` command with `-C` option which reports the resource usage and its criticality.

In this example, the criticality reported by Direct IO analysis is CRA_DATA_CRITICAL as IP is configured for a port whereas criticality returned by Mass Storage and Legacy AVIO Storage analysis is CRA_WARNING resulting in cumulative CRA result being CRA_DATA_CRITICAL. For more information about the CRA details, see the CRA log file `/var/adm/cra.log` on the VSP. The following is the snippet from the CRA log on the VSP containing the MASS STORAGE, legacy AVIO CRA REPORT and HPVM Direct I/O Analysis details.

```
CRA REPORT SUMMARY:
WARNING - One or more subsystems queried for Critical Resources Analysis(CRA)
reported DATA CRITICAL usage on some resources.
A DATA CRITICAL resource must be present to maintain some services up.
Forcing its removal may disrupt such services.

CRA DETAILED REPORT:

ANALYSIS SCOPE: MASS STORAGE
This report provides details of any critical mass storage hardware path usages
in the system.

RESULT: WARNING Some hardware paths to resources are affected

DETAILED REPORT: Analyzed the following mass storage hardware paths to detect
any critical usages in the system:
43/0/2/2/0/0/0/5/0/0/0
43/0/2/2/0/0/0/5/0/0/1

WARNINGS
 Affected Processes:
  PID:
  22525  hpvmapp using 64000/0xfa00/0x3 (/dev/rdisk/disk38)
  under the affected card(s)
  43/0/2/2/0/0/0/5/0/0/0

  PID:
  22256  hpvmapp using 64000/0xfa00/0x4 (/dev/rdisk/disk39)
  under the affected card(s)
  43/0/2/2/0/0/0/5/0/0/0

ANALYSIS SCOPE: HPVM Legacy AVIO storage
This report provides details on critical HPVM legacy AVIO storage
usage in the HPVM environment.

DETAILED HPVM legacy AVIO CRA REPORT:

WARNINGS
 Affected vPars or VM guest are:
 Guest instance with a boot disk with some paths affected: 2,5

ANALYSIS SCOPE:HPVM- DIRECT I/O
This report provides details of any critical Direct I/O hardware path usages in the system.

RESULT: DATA CRITICAL Direct I/O resources will be affected.

SYSTEM CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: NONE

DATA CRITICAL RESULTS
 Affected vPars or VM guest instances numbers are: 5

WARNING
 Affected vPars or VM guest instances numbers are: NONE

FAILURE
 vPars or VM guests that failed to perform CRA analysis are: NONE
```

In this scenario, where the CRA has returned DATA CRITICAL, if you choose to run the `Pre Replace` option of the `olrad (1M)` command (`olrad -r` option), the operation fails with CRA_DATA_CRITICAL error, reason being that doing this operation renders the VM guest where the FLA NIC ports are assigned, inaccessible to network. To verify that the FLA NIC is successfully suspended, following options of the `olrad(1M)` and `ioscan(1M)` commands can be used.

Further ioscan output inside each of the guest will also show the corresponding FLA NIC port in SUSPENDED state as in previous examples.

```
# olrad -q
Slot          Path          Link Max   Max   Link  Pwr  Occu Susp  OLAR OLD  Mode
                            Spd  Link  Link  Width
                                 Spd   Width
10-0-2-2-2-2  43/0/2/2/0/0  5.0  5.0   x8    x8    Off  Yes  Yes   Yes  Yes  PCIe

# ioscan -kfnC fc
Class     I  H/W Path        Driver S/W State   H/W Type     Description
========================================================================
fc        6  43/0/2/2/0/0/0/5/0/0/0  fclp   SUSPENDED   INTERFACE    HP AT094-60001 PCIe Fibre Channel
2-port 8Gb FC/2-port 10GBE Combo Adapter
/dev/fclp6
fc        7  43/0/2/2/0/0/0/5/0/0/1  fclp   SUSPENDED   INTERFACE    HP AT094-60001 PCIe Fibre Channel
2-port 8Gb FC/2-port 10GBE Combo Adapter
/dev/fclp7

# ioscan -kfnC hpvmdio
Class     I  H/W Path        Driver  S/W State   H/W Type     Description
=========================================================================
hpvmdio   2  43/0/2/2/0/0/0/8/0/0/0  hpvmdio   SUSPENDED   INTERFACE    HP AT094A 10GbE-SFP PCIe
2p 8Gb FC and 2p 1/10Gbe Adapter
 /dev/hpvmdio2
hpvmdio   4  43/0/2/2/0/0/0/8/0/0/1  hpvmdio   SUSPENDED   INTERFACE    HP AT094A 10GbE-SFP PCIe
2p 8Gb FC and 2p 1/10Gbe Adapter
 /dev/hpvmdio4
```

After the Combo card is replaced with a new one, the `Post Replace` option of the `olrad(1M)` command (`olrad -R`) can be used to resume usage of the card. To verify that the FLA NIC is successfully resumed, the options of the `olrad(1M)` and `ioscan` commands can be used.

AVIO LAN devices

In this scenario, there are two guests with host name `evolution` and `president`, each guest uses vswitch `testlan` which is backed by an NIC card (lan18) on olrad capable PCI slot. The example shows the behavior of `olrad -C` when the IP address is configured on vNIC and without an IP address.

Example 52 Configuration

The VSP has two guests with the following networking configuration.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type  OS Type State #VCPUs #Devs #Nets Memory
==================== ====  ===== ======= ===== ====== ===== ===== =======
evolution            2     SH    HPUX    On(OS)  4     5     18      22 GB
president            1     VP    HPUX    On(OS)  4     9     22   38912 MB
```

The VSP has the following vswitches and lan18 is associated with `testlan vswitch`

```
# hpvmnet
Name      Number State   MOde NamePPA MAC Address IPv4 Address
========= ====== ======= ==== ======= =========== =============
localnet  1 Up   Shared  N/A     N/A
sitelan   2 Up   Shared  lan0    0x2c4138869bde   15.213.202.188
testlan   3 Up   Shared  lan18   0x00237d6c1398
datalan   7 Up   Shared  lan8    0x78e3b5f53eea
```

Testlan vswitch is backed to lan18

```
# ioscan -kfNC lan
Class I  H/W Path       Driver  S/W State  H/W Type   Description
======= ============= ====== ========= ========= ===========================
lan   0  6/0/0/0/0/0/0  iexgbe  CLAIMED    INTERFACE  HP PCIe 2-p 10GbE Built-in
lan   1  6/0/0/0/0/0/1  iexgbe  CLAIMED    INTERFACE  HP PCIe 2-p 10GbE Built-in
lan   2  6/0/0/2/0/0/0  iexgbe  CLAIMED    INTERFACE  HP PCIe 2-p 10GbE Built-in
lan  14  47/0/0/0/0/0/0 iocxgbe CLAIMED    INTERFACE  HP AT111-60001 10Gb PCIe 2-port CNA (NIC) Adapter
lan  15  47/0/0/0/0/0/1 iocxgbe CLAIMED    INTERFACE  HP AT111-60001 10Gb PCIe 2-port CNA (NIC) Adapter
lan  16  47/0/0/2/0/0/0 iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan  17  47/0/0/2/0/0/1 iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan  18  47/0/1/0/0/0/0 iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
lan  19  47/0/1/0/0/0/1 iexgbe  CLAIMED    INTERFACE  HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
```

Guest Configuration:

President: Lan11 inside the guest is configured with IP address 192.168.1.7

```
President#  netstat -in
# netstat -in
Name   Mtu   Network         Address         Ipkts   Ierrs  Opkts  Oerrs  Coll
lan0   1500  15.213.200.0    15.213.202.248  354075  0      26177  0      0
lo0   32808  127.0.0.0       127.0.0.1       23678   0      23678  0      0
lan11  1500  192.0.0.0       192.168.1.7     0       0      0      0      0
```

Evolution: In this guest a VLAN is configured on interface lan3 and the VLAN interface is configured with an IP address 192.168.3.25.

```
# lanscan
Hardware  Station        Crd Hdw   Net-Interface    NM  MAC    HP-DLPI  DLPI
Path      Address        In# State NamePPA          ID  Type   Support  Mjr#
0/0/0/0   0x6AA6B9E88E26 0   UP    lan0    snap0     1  ETHER  Yes      119
0/0/7/0   0x0EF67B389C1E 1   UP    lan1    snap1     2  ETHER  Yes      119
0/1/1/0   0xC26D927F1E6B 3   UP    lan3    snap3     3  ETHER  Yes      119
VLAN5028  0xC26D927F1E6B 5028 UP   lan5028 snap5028 98  ETHER  Yes      119

# netstat -in
Name      Mtu  Network       Address         Ipkts  Ierrs Opkts  Oerrs Coll
lan5028   1500 192.168.3.0   192.168.3.25    32     0     48     0     0
lan0      1500 15.213.200.0  15.213.202.249  356667 0     28397  0     0
lo0      32808 127.0.0.0     127.0.0.1       22700  0     22700  0     0
```

Convert hardware path to slot ID.

```
# olrad -g 47/0/1/0/0/0/0
12-0-2-1-0-5

# olrad -C 12-0-2-1-0-5

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large configurations. It is recommended not to disrupt
this operation.]
```

```
CRA REPORT SUMMARY: CRA detected DATA CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.

#cat /var/adm/cra.log

ANALYSIS SCOPE: HPVM AVIO NETWORKING
This report provides details of any HPVM networking related usages for
a set of h/w paths in the system.

RESULT: DATA-CRITICAL resource usage detected.

DETAILED REPORT: Analyzed following hardware paths to detect any usages in the system:
47/0/1/0/0/0/0 (lan18)
47/0/1/0/0/0/1 (lan19)

DATA CRITICAL RESULTS:
  Affected vPars or VM guest instances numbers configured on vswitch backed by interface lan18
    2, 1, ----------- Guest effected

FAILURES:
 Guest Instance(2) Vswitch lan18: The guest CRA reported Data Critical Warning
 Guest Instance(1) Vswitch lan18: The guest CRA reported Data Critical Warning

NOTE : vswitch presence may show up as hpvmnetd activity

USEFUL HPVM/NETWORKING COMMANDS:
hpvmnet  lanadmin  lanscan  nwmgr  netstat  ifconfig  linkloop
```

When the IP address is unplumbed on the guest `president`

```
# netstat -in

Name  Mtu   Network       Address         Ipkts   Ierrs Opkts   Oerrs Coll
lan0   1500 15.213.200.0  15.213.202.248  358892  0     26525  0     0
lo0   32808 127.0.0.0     127.0.0.1       23905   0     23905  0     0

# olrad -C 12-0-2-1-0-5

# cat /var/adm/cra.log

Critical Resources Analysis(CRA) Report
Logged on: Mon Dec  2 13:23:43 2013

CRA REPORT SUMMARY:
WARNING - One or more subsystems queried for Critical Resources Analysis(CRA) reported
DATA CRITICAL usage on some resources.
A DATA CRITICAL resource must be present to maintain some services up.
Forcing its removal may disrupt such services.

CRA DETAILED REPORT:

ANALYSIS SCOPE: HPVM AVIO NETWORKING
This report provides details of any HPVM networking related usages for
a set of h/w paths in the system.

RESULT: DATA-CRITICAL resource usage detected.

DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
47/0/1/0/0/0/0 (lan18)
47/0/1/0/0/0/1 (lan19)

DATA CRITICAL RESULTS:
Affected vPars or VM guest instances numbers configured on vswitch backed by interface lan18 2,

FAILURES:
Guest Instance(2) Vswitch lan18: The guest CRA reported Data Critical Warning

NOTE : vswitch presence may show up as hpvmnetd activity

USEFUL HPVM/NETWORKING COMMANDS:
hpvmnet  lanadmin  lanscan  nwmgr  netstat  ifconfig  linkloop
```

Example 53 Configuration

In this scenario, the two guests `evolution` and `president` use `testlan` vswitch which is backed by an NIC card (lan18) on olrad capable PCI slot and shows the behavior of `olrad -C` when no VNIC is configured with any IP address.

```
# hpvmnet
Name     Number State   Mode      NamePPA MAC Address    IPv4 Address
======== ====== ======= ========= ======= ============== ===============
localnet 1      Up      Shared            N/A            N/A
sitelan  2      Up      Shared    lan0    0x2c4138869bde 15.213.202.188
testlan  3      Up      Shared    lan18   0x00237d6c1398
datalan  7      Up      Shared    lan8    0x78e3b5f53eea

# ioscan -funC lan | grep 18
lan 18  47/0/1/0/0/0/0  iexgbe  CLAIMED INTERFACE HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter
```

When no IP address is configured on the VNIC the HPVM AVIO networking returns SUCCESS, however the generic LAN will return DATA CRITICAL. In this case, you can assume that there is no VNIC usage.

```
Get the Slot ID of NIC card.
# olrad -g 47/0/1/0/0/0/0
12-0-1-1-0-5

# olrad -C 12-0-1-1-0-5

# cat /var/adm/cra.log

ANALYSIS SCOPE: NETWORKING □----------- Note that Analysis scope is Networking.
This report provides details of any networking related usages for
a set of h/w paths in the system.

RESULT: DATA-CRITICAL resource usage detected.

DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
47/0/1/0/0/0/0 (lan18)
47/0/1/0/0/0/1 (lan19)

DATA CRITICAL RESULTS:

 Interface lan18:        COMMAND hpvmnetd        PID 3221

 Interface lan18:        COMMAND hpvmnetd        PID 3221

USEFUL NETWORKING COMMANDS:
lanadmin  lanscan  nwmgr  netstat  ifconfig linkloop
```

Example 54 Configuration

In this scenario, the two guests `evolution` and `president` use `testlan` vswitch is backed to an NIC card (lan18) on olrad capable PCI slot. VNIC is configured with an IP and shows the behavior of suspend (`olrad -r` and `olrad -f -r`) and resume of a card.

Hardware path of lan18

```
# ioscan -funC lan | grep 18
lan      18  47/0/1/0/0/0/0  iexgbe    CLAIMED     INTERFACE    HP AM225-60001 PCIe 2-p 10GbE-SFP+ Adapter

Slot ID
# olrad -g 47/0/1/0/0/0/0
12-0-2-1-0-5

# olrad -r 12-0-2-1-0-5
Activity: Start of Prepare Replace
Target slot: 12-0-2-1-0-5

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA detected DATA CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.

CRA Error: resources associated with possible data loss
Target slot: 12-0-2-1-0-5
```

When there is a data critical, suspend of card will be stopped. Admin must analyze the `/var/adm/cra.log` before using the `-f` option to continue with the suspend.

```
# olrad -f -r 12-0-2-1-0-5
Activity: Start of Prepare Replace
Target slot: 12-0-2-1-0-5

Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA detected DATA CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.

CRA Error: resources associated with possible data loss
Target slot: 12-0-2-1-0-5

Activity: CRA being forced using -f option
Target slot: 12-0-2-1-0-5

Activity: End of Prepare Replace
Target slot: 12-0-2-1-0-5

Activity: Target slot powered off, drivers suspended, OK to replace the card
Target slot: 12-0-2-1-0-5

Resuming the Card.
# olrad  -R 12-0-2-1-0-5
Activity: Start of Post Replace
Target slot: 12-0-2-1-0-5

Activity: End of Post Replace
Target slot: 12-0-2-1-0-5

Activity: Target slot powered on, drivers resumed, OK to start using the card
Target slot: 12-0-2-1-0-5
```

AVIO storage devices

Example 55 Configuration

The VSP has an active VM guest configured with legacy AVIO backing stores.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type  OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ====  ======= ========= ====== ===== ===== =======
guest1             3     SH     HPUX    On(OS)    2      2     1     4 GB

# hpvmstatus -p 3 -d | grep avio_stor
disk:avio_stor:0,0,0:disk:/dev/rdisk/disk51
disk:avio_stor:1,0,0:disk:/dev/rdisk/disk52
```

The boot device of the guest is `/dev/rdisk/disk28`, and it is mapped to the VSP device `/dev/rdisk/disk51`

```
guest1# ioscan -kfNd gvsd
Class   I   H/W Path  Driver S/W State   H/W Type     Description
======= === ========  ====== =========   ==========   ====================
ext_bus 18  0/0/0/0   gvsd   CLAIMED     INTERFACE    HPVM AVIO Stor Adapter
ext_bus 21  0/1/0/0   gvsd   CLAIMED     INTERFACE    HPVM AVIO Stor Adapter
guest1# setboot
Primary bootpath : 0/0/0/0.0x0.0x0 (/dev/rdisk/disk28) -------------> Boot device
HA Alternate bootpath :
Alternate bootpath :

guest1# hpvmdevinfo

Device Type  Bus,Device,Target  Backing Store Type  Host Device Name  Virtual Machine Device Name
===========  =================  ==================  ================  ============================
disk         [0,0,0]            disk                /dev/rdisk/disk51 /dev/rdisk/disk28
disk         [1,0,0]            disk                /dev/rdisk/disk52 /dev/rdisk/disk29
```

With legacy AVIO devices, there is no multi-pathing capability within the guest, so the guest has one single path to each legacy AVIO device that it sees.

On the VSP, the two paths to the boot device of the guest are through the same FC port (`/dev/fclp3`), which is on the HBA card in slot 10-0-1-0-2-3.

```
# ioscan -m lun /dev/rdisk/disk51
Class I  Lun  H/W Path          Driver  S/W      State   H/W Type Health  Description
=======  ==== ================  ======= =======  ======  =======  ======= ===========
disk     51   4000/0xfa00/0x1   esdisk  CLAIMED  DEVICE  online   HP      HSV200
        42/0/0/2/0/0/0/4/0/0/0.0x50001fe15006c768.0x4002000000000000 ----> Both paths are through the same HBA
 port
        42/0/0/2/0/0/0/4/0/0/0.0x50001fe15006c76d.0x4002000000000000 ----> Both paths are through the same HBA
 port
             /dev/disk/disk51      /dev/disk/disk51_p2   /dev/rdisk/disk51     /dev/rdisk/disk51_p2
             /dev/disk/disk51_p1   /dev/disk/disk51_p3   /dev/rdisk/disk51_p1  /dev/rdisk/disk51_p3
```

OLR Operation:

```
# olrad -C 10-0-1-0-2-3
Critical Resource Analysis(CRA) in progress...
[NOTE: The CRA may take a few minutes to complete on large
configurations. It is recommended not to disrupt this operation.]

CRA REPORT SUMMARY: CRA detected SYSTEM CRITICAL usages.

Detailed CRA report is available in /var/adm/cra.log file.
```

The following is the snippet from the CRA log on the VSP:

```
ANALYSIS SCOPE: HPVM Legacy AVIO storage
This report provides details on critical HPVM legacy AVIO storage usage in the HPVM environment.

DETAILED HPVM legacy AVIO CRA REPORT:

SYSTEM CRITICAL RESULTS

Affected vPars or VM guest are:
  3 --------> VM guest guest1
```

If there was an additional path to `/dev/rdisk/disk51` on the VSP through an FC card on a slot other than 10-0-1-0-2-3, then, the CRA on slot 10-0-1-0-2-3 reports the severity as WARNING and the administrator can proceed with the OLR operation without having to bring down the VM guest `guest1`.

Time taken for CRA on a VSP

The default timeout value set for each guest OS to complete CRA requests issued to it, as part of the host PCI OLR operations initiated using the `olrad(1M)`. The value is two minutes. For guests with large, active, and I/O configurations this may be insufficient. Administrators can configure the timeout value by defining the parameter *`OLR_GUEST_RESP_TIMEOUT`* in `/etc/rc.config.d/hpvmconf` the timeout value must be specified in milliseconds.

Impact of PCI OLR on HPVM

PCI OLR operations require the temporary suspension of affected I/O traffic within guests. During this time, administrative operations that change the configuration or active status of running guests are not allowed. Operations such as `hpvmstart(1M)` or `vparboot(1M)`, CPU, memory or dynamic I/O OLAD using `hpvmmodify(1M)` or `vparmodify(1M)`, `hpvmstop(1M)`, `hpvmsuspend(1M)`, `hpvmmigrate(1M)`, and so on, are not allowed to run while a host PCI OLR operation is in progress. These operations are serialized using a software lock. If any one of these commands is running, attempts to run the same or any other commands that modify guest configuration or state are failed with a message stating `unable to get file lock`. In such cases, the operation may be retried after the first is completed.

Limitations of PCI OLR on SD2 VSPs

The following are the limitations of PCI OLR on SD2 VSPs:

- Online VM migration or resume of a guest that is using a resource backed by a suspended VSP I/O resource fails.
- Starting a guest that has a resource backed by a suspended VSP resource succeeds as long as the guest boot is not impacted.
- If a guest with a resource backed by a suspended VSP resource is booted, and the resource is resumed on the VSP at a later point, a guest reboot is be required before the guest resource is usable or online.
- If the OLR of a VSP resource impacts more than 32 I/O devices (vHBAs or vNICs) on any guest, the OLR VSP operation fails.
- CRA on the guest will not take non-NPIV devices into account, that are configured as primary boot (when the primary boot device is not the one on which the current boot occurred), secondary boot, and dump devices. But, a caution is displayed in the VSP CRA log.
- For non-NPIV resources, no resource usage will be logged within the guest CRA log.
- While a PCI OLR is in progress on the VSP, no other HPVM command that results in a guest state change can be executed (guest start, stop, migrate, suspend, resume, modify). Similarly, while an HPVM command that can change the state of a guest is in progress, a `PCI OLRAD` command cannot be executed on the VSP.
- PCI OLR is currently not supported on guests using vlan backed vswitches. When a CRA request is issued to `olrad` capable slot containing a physical NIC and the NIC has virtual LAN Interface (VLAN interfaces) backed to a vswitch always returns CRA_SUCCESS.
- A PCI OLR operation or a CRA on the VSP is not supported if any of the active vPars or VM guests have more than 32 impacted IO devices of a particular type (NPIV or AVIO LAN) on the I/O card.
- When a PCI OLR operation on the VSP reports a legacy AVIO resource as data critical for any of the active vPars or VM guests, the `olrad` command must not be retried with the `force` option without manually verifying if the primary or secondary boot device, swap, or dump device or cluster lock disk configured within the vPar or VM will be impacted.
- The CRA on the VSP cannot determine that a vPar or VM guest is in the middle of a recovery boot process. Hence, Hewlett Packard Enterprise recommends that one does not attempt

a PCI OLRAD operation on the VSP if any of the vPars or VM guests are in the middle of a recovery boot.

You must retry the operations after the recover boot is complete and the guest is back to stable state (that is, either shutdown has completed or recovery boot).

- The CRA on the VSP fails if any of the vPar or VM guests are in the middle of an operating system installation. You must retry the operation after the guest installation is complete and the guest is back to stable state (that is, either shutdown has completed or boot post installation).

12 Migrating VMs and vPars

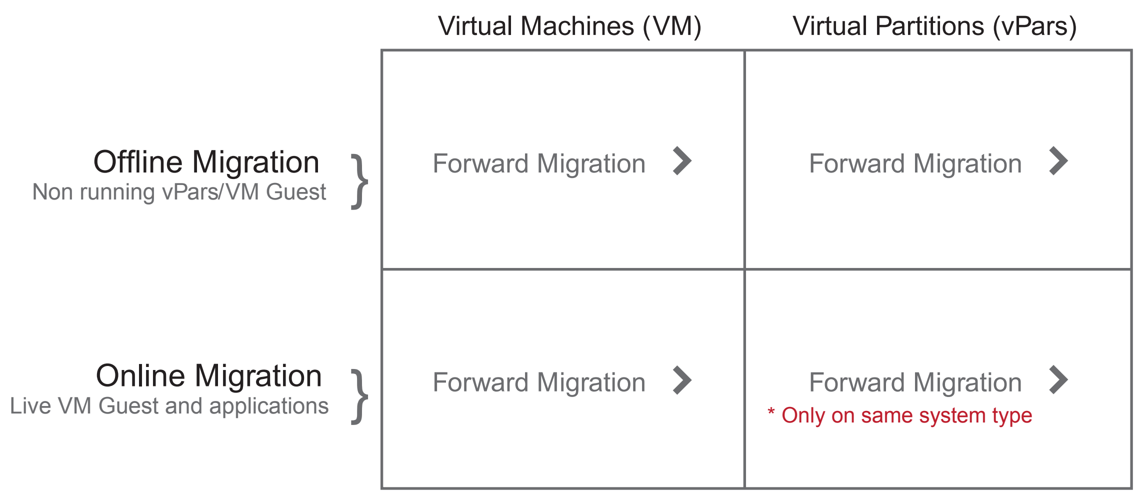
You can migrate either an offline vPar or VM, or a live online vPar and VM running a guest operating system and applications from a source VSP system to a target VSP system, using the `hpvmmigrate` command.

Introduction to migration

vPars and Integrity VM v6.4 allows the following types of migration:

- To migrate a VM or vPar from one VSP system to another, use the `hpvmmigrate` command. The VM can be a non-running VM guest, a vPar configuration (offline migration) or a running VM or vPar guest (online migration). Online migration enables a running VM or vPar and its applications to be migrated from one VSP to another without service interruption. All VM and vPar I/O connections to storage and networks remain active throughout the online migration, and it is not necessary to reboot VM or a vPar and restart applications.

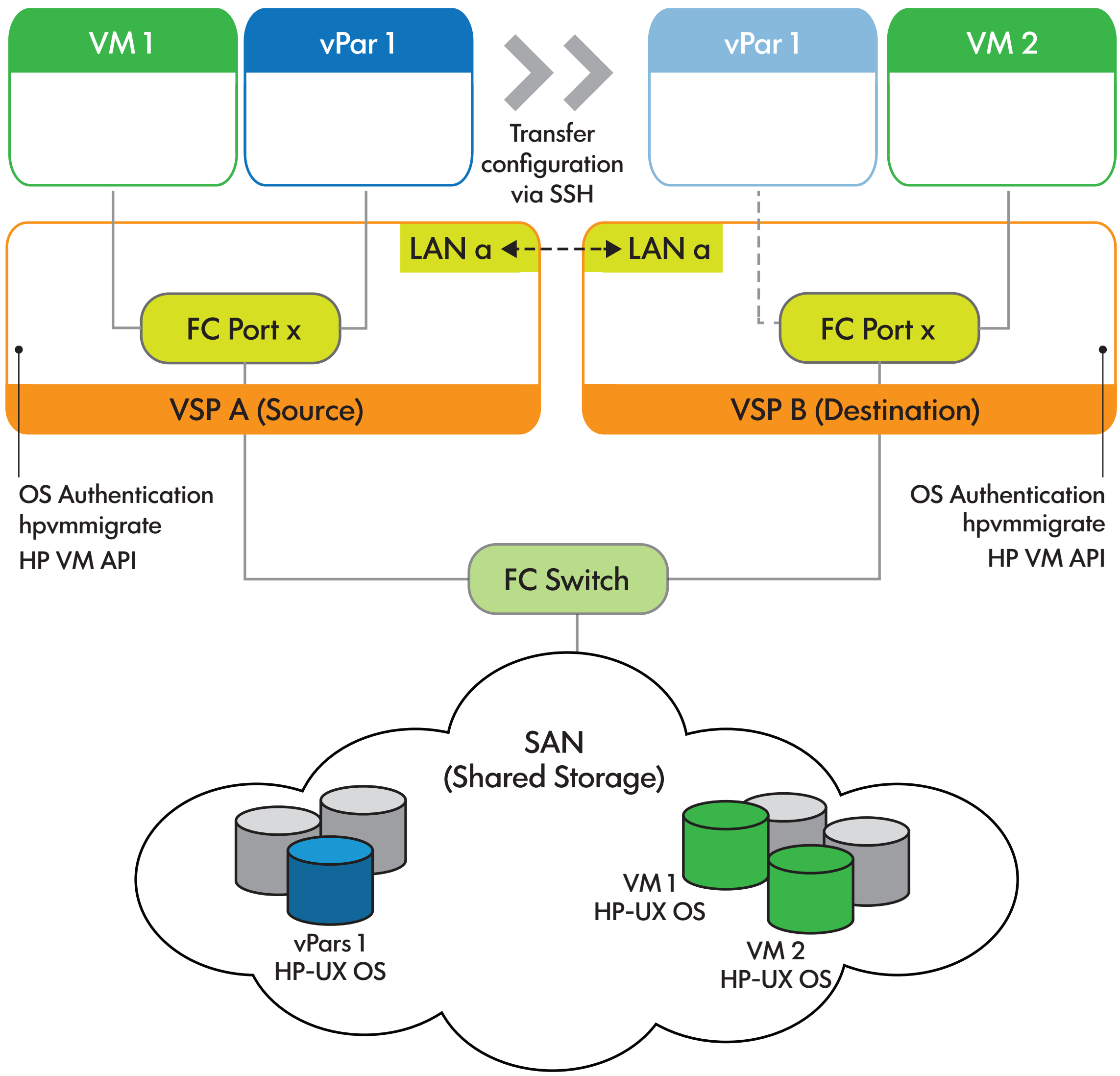
  Figure 17 Online and Offline forward migration possibilities

- To migrate a Serviceguard Packaged VM or vPar online, use the `cmmovevpkg` command. For more information, see the `cmmovevpkg(1M)` manpage or the serviceguard toolkit for integrity virtual servers user guide at **http://www.hpe.com/info/hpux-serviceguard-docs**.

Figure 18 (page 204) shows the process of migrating a guest from Host A to Host B offline.

Figure 18 Symmetric VSPs configured for guest migration



The VM or vPar migration environment includes a source machine and a target machine. Both must be running vPars and Integrity VM, be able to run the guests, conform to their operating system requirements and restrictions, and must be able to provide the allocated resources to the guest. If the guest uses 2 GB of memory on one machine, it must be able to use that amount on the other machine. Similarly, if the source machine can provide a guest with four vCPUs, the target machine must also be able to provide the same. To modify the virtual devices or network on the target host, use the `hpvmmodify` command.

To enable migration, all resources used by the guest must be configured symmetrically on both the source and target host. A symmetric configuration includes:

- A common LAN
- Identical subnet and vswitch connectivity
- Common access for SAN based storage
- Private, high-speed network connection (for Online VM or vPar Migration)

For guidelines about setting up storage for migrating VMs or vPars, see “VSP and VM or vPar configuration considerations” (page 211).

If the HP Capacity Advisor is used on the VM or vPar, you must collect utilization information before migrating the VM on vPar. The Capacity Advisor cannot continue to collect the utilization information for the VM or vPar during the migration.

Considerations for migrating an online VM or vPars

Following are the considerations to migrate an online VM or vPars:

- Vacating a VSP system—With online VM or vPar migration, you can migrate all VMs or vPars from a VSP to one or more VSPs without interrupting the workload activity on the VM and vPar. This is most often done for the maintenance of the VSP system—hardware, firmware, or software. You can configure the hardware that does not have hot-plug support. You can update the firmware, which requires the system to be shut down. You can also update software components that require a VSP reboot. A rolling upgrade of VSP software is possible by moving the running guests to another VSP, upgrading the VSP, and then migrating the guests back. Moving VMs or vPars while keeping active applications online allows greater flexibility in scheduling maintenance or upgrades, and minimizes the impact of unpredictable maintenance. For example, you can move online VMs or vPars in response to predictive failure alerts without interrupting your applications.
- Targeting a particular VSP—You might want to migrate an active VM or vPar workload to a particular VSP to take advantage of a particular resource or feature on that target VSP without losing application availability. If your current VSP resources become oversubscribed, you can migrate one or more of the VMs or vPars to other VSPs that have the remaining capacity. A potential target VSP might have a large quantity of RAM, CPUs, or I/O adapters, which might facilitate faster processing or greater I/O bandwidth while on that VSP. Another possibility is that, certain VSPs have special devices that are needed only temporarily by VM or vPar workloads. Because online VM or vPar migration enables VMs or vpars to be migrated without interrupting their workloads, it is convenient and practical to migrate VMs or vPars temporarily to certain VSPs to take advantage of particular resources and features when they are needed. This is especially true for workloads with well-understood cyclic resource requirements (for example, month-end processing).
- Balancing VSP workloads—You might want to segregate VMs or vPars to balance the workload on VSPs. For example, you might want to separate VMs or vPars whose workloads peak simultaneously. Perhaps you want to group workloads together that have similar special resource requirements. For example, you will run your multi-threaded applications on a VSP that has several CPUs in order to maximize the effectiveness of multi-way VMs or vPars. Online VM or vPar migration enables a new level of workload-to-resource alignment flexibility and agility where you can segregate or combine your workloads, without any interruption in application availability.
- Optimizing physical resource utilization—The online VM or vPar migration feature enables you to optimize the physical resources in use by running VM or vPar. You can move (or park) idle VM or vPar, near-idle VM or vPar, or VM or vPar with currently less-critical workloads on a smaller or less powerful machine. You can use the dynamic memory feature to reduce the amount of memory in use by the VMs and shrink CPU entitlements to more tightly packed VMs on a smaller VSP.

For more information about the online and offline migration support, see *HP-UX vPars and Integrity VM Release Notes* available at **http://www.hpe.com/info/hpux-hpvm-docs**.

To verify whether a guest can be migrated to the target VSP, use the `hpvmmigrate -s` option.

Considerations for migrating VMs or vPars offline

Following are the considerations to migrate a VM or vPar offline:

- The vPar or VM can be stopped; you must move the configuration information offline.
- Migrating the VM or vPar offline does not use the VSP resources (such as memory and CPUs) on the source and target VSPs.
- The vPar or VM might have local storage, logical volumes, or file-backed storage, which must be copied to the target VSP.
- The source and target VSPs might have different processor types that prevent online migration.
- You can migrate vPars or VMs offline between different processor families.

For more information about the migration path for offline migration, see *HP-UX vPars and Integrity VM Release Notes* available at **http://www.hpe.com/info/hpux-hpvm-docs**.

Offline migration of a vPar or VM guest with DIO functions assigned requires that each function is assigned a label using the `hpvmhwmgmt -L label` switch (See `hpvmhwmgmt(1M)` for the command syntax.). Additionally, for each DIO-capable function on the vPar or VM guest on the source VSP, there must be at least one DIO capable function on the target VSP.

A label can contain up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (-), the underscore (_), and the period (.), except that it might not be the string "." or "..". Labels apply only to DIO functions, and are added to the DIO pool on the source and target VSPs using the following command:

```
hpvmhwmgmt -p dio -a hwpath
```

If any DIO function in a vPar or VM does not have a label, offline migration fails. There might be more functions with the same label available on the target VSP than are needed to do a one-to-one matching of DIO functions on the target VSP, but there must be at least a one-to-one correspondence between each labeled function on the source vPar or VM and available DIO-capable functions on the target VSP.

Hewlett Packard Enterprise recommends that you assign labels to correspond to IP names, so that the network mapping on the source vPar or VM is preserved when the vPar or VM is migrated to the target VSP. It is not a requirement for offline migration to succeed, but failure to maintain the one-to-one correspondence with IP names might cause problems when the migrated vPar or VM is started.

If a target VSP contains multiple DIO-capable functions with the same label, it might be possible that offline migration picks the DIO-capable function which is used by another vPar or VM. In such cases, the vPar or VM that is migrated offline will not be able to power on if another vPar or VM assigned with the same DIO-capable function is already running. You must either manually change the DIO-capable function with an unused DIO-capable function or assign the DIO-capable functions with unique labels such that it maintains an exact one-to-one mapping between each labeled function on the source vPar or VM and available DIO-capable functions on the target VSP.

Label-matching is independent of whether the labels are assigned to DLA or FLA functions. However, offline migration first attempts a match of like-for-like function types. For more information about DLA and FLA distinction, see “Using direct I/O networking” (page 139).

Command line interface for migration

To migrate a VM or vPar to another VSP:

1. Set up SSH keys on both the source and target hosts, as described in “SSH setup between the VSPs” (page 215).
2. Present all SAN storage assigned to the VM or vPar to the target VSP (if it is not already there).

3. If using offline migration and the guest is booted, stop the guest on the source host, using the `hpvmstop` or `hpvmconsole` command. You can also use the `hpvmmigrate -d` command to stop the guest during the migration. This has an advantage in that, the resource checks are made on the target before the guest is stopped on the source. However, it is best to log into the guest and shut it down before starting an offline migration. This ensures that all guest data is properly flushed to the disks.

   For information about starting and stopping guests, see "Managing vPars and VMs using CLI" (page 239).
4. On the source host, enter the `hpvmmigrate` command, as described in "Using the `hpvmmigrate` command" (page 207). When migrating an online guest, there are several reasons why the migration might abort, leaving the guest running on the source host. The success or failure of migrations is reported by the `hpvmmigrate` command. Causes for the abortion include insufficient resources on the target host, excessively busy VSPs, a slow network connection, or busy guest. If such conditions exist, the migration attempt is aborted so the workload of the guest can continue running on the source host. This is not a serious problem because the migration can be re-attempted when conditions improve.
5. If migrating the guest offline, restart the guest on the target host using the `hpvmstart` or `hpvmconsole` command. You can also use the `hpvmmigrate -b` option with an offline migration to automatically restart the guest on the target.

If you do not use the `hpvmmigrate -D` option to remove the VM or vPar configuration on the source VSP, it is marked `Not Runnable`, and it is configured with all its devices. This protects the storage from unintended use by Integrity VM commands.

If you never intend to migrate the guest back to the source VSP, you can remove the VM or vPar configuration with the `hpvmremove` command. After the guest is removed from the VSP, you must unpresent the SAN storage of the guest and remove the associated device special files (using the `rmsf` command). If you cannot unpresent the storage, you must use the `hpvmdevmgmt -a rdev:/`*`device`* command for each device to mark them restricted.

The `hpvmmigrate` command verifies that the target host has sufficient resources (such as memory, network switches, and storage devices) for the guest to run. If the resources are insufficient or do not exist, or if other errors occur, the guest is not migrated to the target host.

After successfully migrating the guest, the `hpvmmigrate` command automatically disables the guest on the source host.

Using the `hpvmmigrate` command

You can migrate an online or an offline VM or vPar from a source VSP to a specified target VSP using the `hpvmmigrate` command. vPars and VMs can be migrated while OFF, and online guests can be migrated while ON and running. You can use the `-o` option with VMs or vPar to migrate an online guest, which involves copying all the configuration information of the VM or vPar and transferring the active guest memory and virtual CPU state. Omit the `-o` option to migrate the configuration information of the offline VM or vPar, and optionally local disk contents to the target VSP.

The resources that are defined in the configuration information of the VM or vPar are verified to determine whether the migrated VM or vPar can boot on the target VSP. If there is a problem, it is reported and the VM or vPar is not migrated. You can specify the `-F` (force) option to suppress the errors and force the VM migration to the target VSP.

CAUTION: The `-F` option is deprecated in Integrity VM commands. This option must be used only if instructed by HPE Support.

By default, Integrity VM or vPar retains the configuration and marks it `Not Runnable(NR)` on the source VSP after it is migrated successfully to the target VSP. Run the `hpvmstatus` command to make sure that the state of the VM or vPar is `Off(NR)` on the source VSP and the guest is `On(OS)` on the target VSP. The guest is running on the target VSP and is, therefore, considered `Runnable`.

This mechanism allows the same VM or vPar to be configured on multiple VSPs, while still preventing accidental booting of the same guest on multiple hosts simultaneously. At any given time, a VM or vPar must be `Runnable` on only one VSP to prevent the possibility of two VMs or vPars using the same SAN storage at the same time. You must use the `hpvmmodify` command, if necessary, to mark the VM or vPar `Runnable` on only the VSP, and `Not Runnable` on all other hosts that know the VM or vPar configuration information.

IMPORTANT: Mark a migrated VM or vPar as `Runnable` only in rare circumstances and with care. Inappropriate use can cause corrupt the disk.

When you run the `hpvmmigrate` command, you must specify the name of the guest to be migrated and the target VSP system.

Specify the guest using one of the following options:

- `-P` *`source-vm-name`* to specify the guest name
- `-p` *`source-vm_number`* to specify the VM number

Specify the target host by including the `-h` option and specifying one of the following:

- Target host alias for the private, high-speed network connection
- Target host IP address of the private, high-speed network connection

NOTE: If you migrate a VM or vPar that is managed by Matrix OE, use Capacity Advisor to collect utilization data before you migrate the VM or vPar. Otherwise, the utilization information about the VSP prior to the migration is lost.

Table 29 (page 208) lists the options that can be used with the `hpvmmigrate` command.

Table 29 Options to the `hpvmmigrate` command

Option	Description
`-A`	Attempts to abort an online VM or vPar migration.
`-b`	For offline migrations, causes the `hpvmmigrate` command to automatically boot the VM or vPar on the target after the migration process is complete. If the `-b` option is specified for an offline migration, all backing stores must be copied.
`-c`*`number-vcpus`*	For offline migrations, specifies the number of virtual CPUs for which this VM or vPar will be configured on the target.
`-C`	For offline migrations, physically copies the storage device specified with the `-m` option to the target VSP during the migration process. If specified before the first `-m` option, it applies to all `-m` options that specify an appropriate type of storage. This might take a long time to complete if a large amount of storage is to be copied.
`-d`	For offline migrations, causes the `hpvmmigrate` command to automatically shut down a running guest before migrating the VM or vPar configuration to the target VSP. Consider migrating the guest online by using the `-o` option instead.
`-D`	Deletes the VM or vPar from the source VSP after migrating the VM or vPar to the target VSP system. If not specified, the VM or vPar is marked `Not Runnable` on the source VSP after migration.
`-e [`*`:max-percent`*`]`	For offline migrations, specifies the percentage of CPU resources to which the VM's virtual CPUs is entitled. During peak system CPU load,

Table 29 Options to the `hpvmmigrate` command *(continued)*

Option	Description
	the entitlement is the guaranteed minimum allocation of CPU resources for this VM. The percent can be set to an integral value between 0 and 100. If the value specified is less than 5, the VM is allocated the minimum percentage of 5%. The default is 10%. Integrity VM reserves processing power for essential system functions such as logging, networking, and file system daemons. The `-e` and the `-E` options are mutually exclusive.
`-E` [*:max-cycles*]	For offline migrations, specifies the CPU entitlement of the VM in CPU cycles. The cycles are expressed as an integer followed by one of these units: • M (megahertz) • G (gigahertz) If no letter is specified, the default unit is megahertz. The `-e` and the `-E` options are mutually exclusive.
`-F`	Forces the migration of a VM or vPar, whether or not there are resource validation errors (such as resource conflict, resource nonexistence, and so on). Use the `-F` option rarely and with caution. This option ignores all resource validation errors, including oversubscribing of resources. **NOTE:** These errors can prevent the VM or vPar from booting on the target VSP. Any validation errors are logged in the Integrity VM or vPar command log. The `-F` option is deprecated in Integrity VM or vPar commands; this option must be used only if instructed by HPE Support.
`-h` *target-host-alias-or-IP-address*	Specifies the host alias or IP address of the target VSP machine to which the VM is migrated. The target machine must be a valid VSP and must be accessible by the source VSP. Almost all forms of the `hpvmmigrate` command require the `-h` option. For online migration, the parameter for the `-h` option must specify a private, dedicated, high-speed network link to the target VSP. If you specify a simple non-qualified host name, the `hpvmmigrate` command appends `-hpvm-migr` to the name and checks if a host alias is defined for a private network corresponding to the simple name. Online guest migration does not check to ensure the link is private, but using a private network is important for efficient and secure online migrations and to preserve the bandwidth of the regular site network.
`-H`	Displays information about how to use the `hpvmmigrate` command.
`-k`	Creates the VM or vPar configuration on the target VSP and marks it `Not Runnable`, but does not change the VM or vPar on the source VSP. This is used primarily to distribute VM or vPar configurations for Serviceguard.
`-l` *new-vm-label*	Specifies a descriptive label for the VM or vPar, which can be useful in identifying a specific VM or vPar in the verbose display of the `hpvmstatus` command. The label can contain up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (-), the underscore (_), and the period (.). To specify white space, the label must be quoted (" ").
`-m` *rsrc-with-absolute-path*	For offline migrations, specifies a resource of a VM or vPar for copying, translation, and so on. This option can be specified more than once. For more information about specifying VM or vPar storage and network resources, see *hpvmresources*(5).
`-n`	Quits after starting the migration in the background. If not specified, the `hpvmmigrate` command continues to run interactively and reports the migration status until the migration is complete.

Table 29 Options to the `hpvmmigrate` command *(continued)*

Option	Description
`-N` *`new-vm-name`*	Specifies the new name for the VM or vPar being migrated. The *`new-vm-name`* can be up to 255 alphanumeric characters, including A-Z, a-z, 0–9, the dash (-), the underscore character (_), and the period (.). The VM or vPar name must not start with a dash (-). If the VM or vPar name exists on the target VSP, the VM or vPar must have the same UUID as the source VM or vPar, and the VM or vPar on the target must be marked `Not Runnable`.
`-o`	Specifies an online guest migration. To be compatible for online migrations, both the source and the target VSP must have the same processor family (as reported by the `machinfo` command). To maintain online guest network connectivity, a vswitch with the same name and connected to the same subnet must be configured on the target VSP. Also, only whole disk backing storage consisting of SAN LUNs, and null backing store DVD devices, are supported for online migration guest storage.
`-p` *`source-vm-number`*	Specifies the unique number of the VM to be migrated. To view the *`source-vm-number`*, run the `hpvmstatus` command. Most forms of the `hpvmmigrate` command require either the `-p` option or the `-P` option.
`-P` *`source-vm-name`*	Specifies the unique name of the VM or vPar to be migrated. Most forms of the `hpvmmigrate` command require either the `-p` option or the `-P` option.
`-q`	Displays fewer informative messages. Some potential error conditions are still reported.
`-Q`	For online migrations, sets the non-interactive mode. Assuming that the output device is not a terminal.
`-r` *`amount`*	For offline migration, specifies the amount of memory available to this VM or vPar. The size is expressed as an integer, optionally followed by one of these letters: • M (megabytes) • G (gigabytes) If the letter is omitted, the default unit is megabytes.
`-s`	Indicates that the migration must not occur, but the `hpvmmigrate` command must check whether or not the migration is possible. Because VMs or vPars and their hosts are dynamic, a successful `-s` trial does not always guarantee a subsequent successful migration. The `hpvmmigrate` command with the `-o`, `-s`, and `-h` options (but without a `-p` or `-P` option) verifies host connectivity, licensing, and CPU compatibility for online migration.
`-t`	For offline migrations, translates the storage device names specified with the `-m` option by comparing WWIDs. To compare WWIDs, the storage resources must be present and available on both the source and the target VSPs. If you specify the `-t` option before the first `-m` option, the `-t` option applies to all `-m` options. The `-t` option overrides the`-T` option for storage resources specified with the `-m` option. Device translation is automatic for online migration.
`-T`	For offline migrations, specifies that devices must not be translated.
`-v`	Displays the version of the `hpvmmigrate` command.
`-w`	For online migrations, bypasses all vswitch connectivity checks. Use the `-w` option only if you are certain that the source and target vswitches

Table 29 Options to the `hpvmmigrate` command *(continued)*

Option	Description
	are connected to the same subnet; otherwise, your online guest will lose network connectivity after migrating.
`-Y`	Suppresses encryption negotiations and sends guest memory data.
`-y`	Requires encryption negotiation and sends guest memory data with protection.

NOTE: You must follow the configuration steps listed in "Considerations for migrating an online VM or vPars" (page 205) to migrate VMs or vPar online that are using logical volume backing stores.

Before enabling the guest on the source, check the target to ensure that the guest was not migrated there.

It is rare but possible that a guest is marked `Not Runnable` after a failed offline migration. If this occurs, use the following command to return the guest to the registered state:

```
# hpvmmodify -P guestname -x register_status=enabled
```

VSP and VM or vPar configuration considerations

This section discusses the configuration information required for a successful migration and how to chose the hosts and guests that can participate in online VM and vPar migration. Effective migration of online guests among VSPs depends on proper configuration of the networks and storage that is connected to the VSP and used by the online guests. The `hpvmmigrate` command verifies that the source and target hosts provide the guest with symmetric accessibility to network and storage resources. If you set up the configuration on both hosts before you migrate the guest, the migration task is easier and faster.

To migrate guests among a group of VSP servers, the VSPs require common access to storage devices, networks, and virtual switch configurations. In the case of legacy AVIO devices, pathnames to storage need not be identical; however, the same LUNs assigned to a guest must be presented to both the source and the target VSPs. In the case of NPIV, for migration across VSPs on a shared FC fabric, the target VSPs must have FC ports that can access all the targets ports that the NPIV HBA had access to on the source VSP. In case of offline migration across disjoint fabrics, appropriate FC ports on the source and target VSP must be labeled identically for proper placement of NPIV HBAs. There must be equal access to guest storage and equal network reachability on both the source and the target VSPs.The network on the target VSP must be able to make all the same network connections that can be used by the guest on the source VSP.

A vswitch of the same name, connected to the same network must be available on the source and target VSP servers. The `hpvmmigrate` command verifies connectivity before migration. You can use the `hpvmmigrate -w` option to bypass the vswitch connectivity checks, but only use `-w` if you are certain that the source and target vswitches are connected to the same subnet. Otherwise, your guest will lose network connectivity after migrating.

For online migration, in addition to sharing the same LAN segment for normal guest connectivity, the VSPs must be connected with a private 1 GbE (or faster) network for efficient VSP-to-VSP communications and for secure guest memory transfer. Hewlett Packard Enterprise strongly recommends using NTP for time synchronization on all VSPs and guests to maintain consistent time accuracy.

Using Network Time Protocol (NTP) with HP-UX Virtualization

Hewlett Packard Enterprise recommends using NTP with HP-UX Virtualization to keep time-of-day clocks in sync and correct. You can use the `xntpd` command on HP-UX to synchronize time.

NTP configuration on a VSP

On each VSP, NTP must be configured as it would be on any typical (non-virtual) system. In `/etc/ntp.conf` file, specify a drift file and one or more high quality time servers:

```
driftfile /etc/ntp.drift
server <A-HIGH-QUALITY-TIME-SERVER> prefer # a preferred time source
server <ANOTHER-HIGH-QUALITY-TIME-SERVER>  # a backup time source
server <YET-ANOTHER-HIGH-QUALITY-TIME-SERVER>
```

The local clock must also be configured as a fall back if necessary:

```
server 127.127.1.0                   # use local clock as backup
fudge 127.127.1.0 stratum 10         # show poor quality
```

If you have a group of VSPs that you would like to synchronize, you can add "peer" references in the `/etc/ntp.conf` file for each of those associated VSPs, so there is mutual synchronization:

```
peer <AN-ASSOCIATED-VM-HOST>
peer <ANOTHER-ASSOCIATED-VM-HOST>
peer <YET-ANOTHER-ASSOCIATED-VM-HOST>
```

After configuring the `/etc/ntp.conf` file of the VSP, assuming the NTP is already enabled, (that is, the XNTPD variable in `/etc/rc.config.d/netdaemons` is set to 1, as in export XNTPD-1), you can run the `/sbin/init.d/xntpd start` command to restart the `xntpd` command on the HP-UX VSP.

NTP configuration on a vPar and Integrity VM guests

NTP was not designed to run in a virtualized environment. Consequently, you must be careful in using NTP within vPar and Integrity VM guests. Using the default NTP configuration on guests might result in NTP instability and failure to synchronize or in apparent lost time within the guest. To avoid these virtualization related NTP issues, all guests must get time directly from the VSP. Further, guests must not serve time to any other systems.

You can monitor NTP status by using the `ntpq -p` command and noting the *`offset`* and *`disp`* values. Both values will be under 100. For information about how to check NTP stability, see *HP-UX Internet Services Administrator’s Guide*.

You can improve time stability within guests by tuning NTP to poll more frequently for time corrections. The default NTP values for the *`minpoll`* and *`maxpoll`* intervals are 6 (64 seconds) and 10 (1024 seconds) respectively. NTP adjusts the current polling interval depending on network quality and delays. A VM guest uses a virtual LAN that can cause NTP to set the polling value incorrectly. To mitigate this issue, use the *`minpoll`* and *`maxpoll`* directives in the `ntp.conf` file to change the polling intervals.

Start with *`minpoll`* at 4 (16 seconds) and *`maxpoll`* at 6 (64 seconds) and then reduce *`maxpoll`* towards 4 if necessary to force shorter polling intervals. Hewlett Packard Enterprise recommends that guests are never allowed to deliver time. For this reason, the local clock (server 127.127.1.0) or an `ntp.drift` file must not be configured on guests. The `ntp.conf` file for guests may be as simple as the single line:

```
server <VM-HOST-SERVER-NAME> minpoll 4 maxpoll 6
```

After configuring the `/etc/ntp.conf` file of the guest, assuming NTP is already enabled (that is, the XNTPD variable in `/etc/rc.config.d/netdaemons` is set to 1, as in export XNTPD=1), you can run the following commands on an HP-UX guest to sync its time with the VSP and restart the `xntpd` command:

```
/sbin/init.d/xntpd stop

/usr/sbin/ntpdate -b <VM-HOST-SERVER-NAME>

/sbin/init.d/xntpd start
```

NOTE: For guests that are on a different subnet than the VSP, the VSP might not be the best source of time if there is another accurate time server available with less network latency. In different subnets, measure latency from the guest to various time servers using the `ping` and `traceroute` commands to determine the potential time server that has the least network latency. Using the VSP might be the best solution, but this depends on your local network topology and the relative network distance to alternate time servers. If it appears best to use an alternate (non VSP) time server, it might be helpful for the alternate time server and the VSP to use each other for peer mutual time synchronization.

VSP requirements and setup

For migrating VMs, see "VSP requirements and setup" (page 224) and for migrating vPars, see " VSP requirements and setup" (page 231).

VSP processors for online migration

For VSP processors for online migration of VMs, see "VSP processors for online migration" (page 224) and for For VSP processors for online migration of vPars, see " VSP processors for online migration" (page 231).

Private network setup

Source and target VSP systems should be connected with a dedicated, high-speed private network. To use the private network during a migration, specify the name of the private network connection in the `hpvmmigrate -h` option. As a helpful convention, if you specify a simple non-qualified host name, the `hpvmmigrate` command appends `-hpvm-migr` to the name and checks if a host alias is defined for a private network corresponding to the simple name. If so, that host-alias is used (that is, *`host`*`-hpvm-migr` is used instead of *`host`*).

To set up a private network between two systems, identify the physical network interfaces that are to be used for the private network. Then, connect those ports to the same network switch, or cable them directly to each other with a cross-over cable if these two VSP systems are the only two systems that migrates guests. Also, BladeSystems in the same enclosure can be connected directly together without an external switch or cable.

Assign private network IP addresses to those interfaces by editing the `/etc/hosts`, `/etc/nsswitch.conf` file, and `/etc/rc.config.d/netconf` on each host. Private (non-routable) IP addresses in the range of 10.0.0.0 to 10.255.255.255 are good choices to use. (See the chapter on Network Addressing for assistance with subnetworking configuration in the current version of the **HP-UX LAN Administrator's Guide**).

In the following example, VSP system `host2` is using network interface lan3 as its private network to connect to VSP `host1`:

Address aliases from `/etc/hosts` on the `host1` and `host2` systems:

```
127.0.0.1        localhost          loopback
15.17.81.141     host1              host1.alg.hp.com
15.17.81.142     host2              host2.alg.hp.com
10.3.81.141      host1-hpvm-migr
10.3.81.142      host2-hpvm-migr
```

Excerpt from `/etc/nsswitch.conf` on the VSP systems:

```
hosts:     files dns
ipnodes:   files dns
```

Excerpt from `/etc/rc.config.d/netconf` on the `host2` system:

```
INTERFACE_NAME[3]=lan3
IP_ADDRESS[3]=10.3.81.142
SUBNET_MASK[3]=255.255.252.0
BROADCAST_ADDRESS[3]=""
```

```
INTERFACE_STATE[3]=""
DHCP_ENABLE[3]=0
INTERFACE_MODULES[3]=""
```

Example output from `netstat` on the `host2` VSP system:

```
# netstat -in
Name      Mtu  Network         Address         Ipkts ...
lan3      1500 10.3.80.0       10.3.81.142     1022313379 ...
lan0      1500 15.17.80.0      15.17.81.142    2420913 ...
lo0      32808 127.0.0.0       127.0.0.1       123762 ...
```

You can also use the `nwmgr` command to help verify the connection. The following example uses the `nwmgr` command on `host1` to get the Station Address (MAC):

```
# nwmgr

Name/          Interface Station          Sub-     Interface      Related
ClassInstance  State     Address          system   Type           Interface
============== ========= ============== ======== ============== =========
lan2           UP        0x001E0B5C0572 igelan   1000Base-SX
lan0           UP        0x001E0B5C05C0 igelan   1000Base-SX
lan1           DOWN      0x001E0B5C05C1 igelan   1000Base-SX
lan3           UP        0x001E0B5C0573 igelan   1000Base-SX
lan900         DOWN      0x000000000000 hp_apa   hp_apa
lan901         DOWN      0x000000000000 hp_apa   hp_apa
lan902         DOWN      0x000000000000 hp_apa   hp_apa
lan903         DOWN      0x000000000000 hp_apa   hp_apa
lan904         DOWN      0x000000000000 hp_apa   hp_apa
```

The following example on `host2` tests the connection to Station Address 0x001E0B5CO573 of `host1`:

```
# nwmgr --diag -A dest=0x001E0B5C0573 -c lan3
lan3: Link check succeeded.
```

You can use the `ssh` and the `env` commands to verify whether the private network connection is working properly between two VSP systems, and whether you are using the correct network interfaces. For example:

```
# ssh host1-hpvm-migr env | grep -i connection
SSH_CONNECTION=10.3.81.142 52215 10.3.81.141 22
```

NOTE: Because Integrity VM disables the TSO and CKO capabilities on the IP address of the LAN interface (resulting in poorer than expected VM Host data-transfer performance), Hewlett Packard Enterprise recommends that you dedicate a LAN interface solely for online VM and vPar migration data transfer to improve data transfer time. That is, to receive the best performance on host-to-remote data transfers on a LAN interface, do not configure a vswitch over it.

Conventions for using *`target-hpvm-migr`* names for private networks

If the name specified for the `hpvmmigrate -h` option is a simple `basename`, the `hpvmmigrate` command concatenates its conventional private network suffix `-hpvm-migr` to the basename and first verifies whether that name can be resolved. A simple `basename` is a reasonably short string with no specified domain hierarchy (for example, period (.) in the name). The simple `basename` cannot contain the conventional suffix `—hpvm-migr` either. You must add the alias *`target`*`-hpvm-migr` to `/etc/hosts` that maps to the private IP network address for VSP target and modify `/etc/nsswitch.conf`, so lookups reference `/etc/host` before using DNS. (The resolution check is done by looking up the modified name with the `gethostbyname` function, so DNS is used if there is no alias in `/etc/hosts`.)

Because this is a convention implemented locally on each host, administrators can or cannot use it. If this convention is configured correctly, both *target* and *target*-hpvm-migr resolve to the proper address. For example:

- hpvmmigrate -h host39 — Look up host39-hpvm-migr first, and if not found, look up host39.
- hpvmmigrate -h host39-hpvm-migr — Look up host39-hpvm-migr.
- hpvmmigrate -h host39.atl — Look up host39.atl.

The *target*.fully.qualified.domain-name will not be modified.

By following this convention, defining an alias with suffix –hpvm-migr for the private network connections, you block the site network for online migrations in case someone accidentally specifies the *hostname* of the target VSP for the hpvmmigrate -h option.

Using NTP on VSPs

Hewlett Packard Enterprise strongly recommends using NTP to synchronize clocks for online VM and vPar migration environments. In addition to a typical NTP configuration, all the potential VSPs must use each other as mutual peer NTP servers to help maintain time consistency between hosts.

For more information about NTP, see "Using Network Time Protocol (NTP) with HP-UX Virtualization" (page 211).

SSH setup between the VSPs

Only superusers can run the hpvmmigrate command. The migration of a guest is controlled by a set of secure remote operations that must be enabled on both systems. The hpvmmigrate command requires HP-UX SSH to be set up on both the source and target host systems, to provide a secure communication path between VSPs. SSH is installed on HP-UX systems by default. The passwords-based and host-based authentication are not supported. SSH security must be set up so that superusers can use ssh commands between the source and target VSPs without interactive passwords.

The hpvmmigrate command uses SSH public-key based authentication between the source and destination hosts. To enable secure communication between the source and target hosts, you must generate SSH keys on both systems. You must have root privileges to generate and set up the SSH keys required for guest migration. You can do this by using the secsetup script provided by Integrity VM.

Run the following command on both the source and target hosts:

```
# /opt/hpvm/bin/secsetup -r other hostname
```

Instead of using secsetup, SSH keys can be generated manually on the systems by using the ssh-keygen command. The ssh-keygen command generates, manages, and converts authentication keys for SSH. For information about manual SSH key generation, see the ssh-keygen command HP-UX manpage.

Troubleshooting SSH key setup

If SSH is installed on both the source and the target system, you can run the ssh command on the source host to establish a connection with the target host without providing password. This ability ensures that SSH keys are set up between the two hosts. If SSH keys are not set up, the hpvmmigrate command displays an error message indicating that the SSH setup must be verified.

If the secsetup script does not work correctly, verify the permissions on root / to ensure that superusers have write permissions. For example,

```
# ll -d /
drwxr-xr-x 20 root root 8192 Apr 29 06:25 /
```

If the root directory of the VSP has different permissions than displayed in the example, use the chmod command to correct them.

```
# chmod 755 /
```

If a VSP is reinstalled at some point after using the secsetup script to configure SSH keys, you might receive warning messages from ssh commands about keys changed, or bad keys in your known_hosts file. In this case, use the ssh-keygen -R *hostname* command to remove obsolete keys from the *known_hosts* file, and then use the secsetup command again to configure new keys.

If you set up SSH security between VSPs before adding the conventional –hpvm-migr host alias to the /etc/hosts file and you do not run secsetup command on the host-alias addresses, the hpvmmigrate command fails with the message, Incorrect initial message, when it attempts to use the conventional host alias.

A workaround is to run SSH once manually (for example, ssh <hostname>-hpvm-migr *date*) and enter **yes** to the question about whether or not you must continue. This action adds <hostname>-hpvm-migr to the list of known hosts, and subsequent hpvmmigrate commands find the proper host key.

Using a third-party SSH

The hpvmmigrate command uses HP-UX native SSH command for secure communication between VSPs. To use an incompatible SSH command with the hpvmmigrate command, make sure your version of SSH is set up for host-based authentication without requiring interactive passwords. Then, set the SSHEXECPATH environment variable (in /etc/rc.config.d/hpvmconf) to invoke a command or shell script similar to the one provided in alt_ssh_example.

Customize alt_ssh_example script for use in your environment, with your version of SSH to translate all the HP-UX SSH specific options to run your alternate SSH command, and to achieve similar behavior. The command or shell script must have permissions similar to a real ssh executable -- it must be writable only by the file owner. The hpvmmigrate command expects to use the HP-UX ssh command as in the following:

```
ssh -e none -o BatchMode=yes -T -x target-host-alias exec hpvmmigrate -#
```

See the alt_ssh_example comments for explanations of the -e, -o, -T, and -x options. With an alternate version of SSH, you might not need some of the HP-UX specific options; or, there might be different options that achieve the same effect; or, perhaps some alternate SSH configuration mechanism can be used eliminating the need for some of the HP-UX specific SSH options.

VM requirements and setup

For information on VM requirements and setup while online migration of VM, see “VM requirements and setup” (page 225).

Setting online migration phase time-out values

For information on setting online migration phase time-out values while online migration of VM, see “Setting online migration phase time-out values” (page 226).

Migrations might time out and must be restarted

To protect a workload of the guest, the online VM and vPar migration feature has limits for the amount of time that a migrating guest can remain in various phases of a migration. There are several capacity and resource-related reasons an attempted online migration might time out and abort, leaving the guest running on the source host. Potential causes include:

- Insufficient resources on the target host
- Excessively busy VSPs
- A slow network connection
- An extremely busy guest

If such conditions exist, the attempted migration is aborted, so the workload of the guest can continue running on the source VSP. This is not a serious problem, because the guest continues to run on the source, and you can re-attempt the migration when conditions improve.

Offline or Online migration can also be retried by adjusting the following hpvmmigrate timeout parameters in the `/etc/rc.config.d/hpvmconf` file.

- HPVMMIGRATE_CONNECT_TIMEOUT— Specifies the timeout value used to check whether the target host is reachable or not. The default is 1000 milliseconds.
- HPVMMIGRATE_SSHCONNECT_TIMEOUT— Specifies the timeout value used for ssh connection. The default is 30000 milliseconds.
- HPVMMIGRATE_NETWORK_TIMEOUT— Specifies the network timeout value for the handshake and initial message exchanges. The default is 15000 milliseconds.
- HPVMMIGRATE_CREATE_TIMEOUT— Specifies the network timeout value for message exchanges while guest creation at target. The default is 120000 milliseconds.
- HPVMMIGRATE_START_TIMEOUT— Specifies the network timeout value for message exchanges during guest start. The default is 120000 milliseconds.

When these variables are not defined then the default values are considered. These variables will be defined in milliseconds in the `/etc/rc.config.d/hpvmconf` file. For example, HPVMMIGRATE_SSHCONNECT_TIMEOUT=35000.

Sharing guest storage device

The guest storage device shareable attribute is not propagated to the target VSP during an online migration. After the first guest that is configured to use the shared storage is online migrated to the target, enable the shared attribute for the device to avoid online migration failures for other guests that share the device. You can use the `hpvmstatus` command to determine the device special filename of the shared device on the target and the `hpvmdevmgmt` command to mark the device shareable. For example:

```
hpvmstatus -P vm_name -d
hpvmdevmgmt -m gdev:/dev/rdisk/disknnn:attr:SHARE=YES
```

For online and offline migration, device special files (DSFs) assigned to VMs do not need to match on source and target VSPs. The `hpvmmigrate` command converts from DSF on the source VSP to WWID and then DSF on the target VSP. You can use the `ioscan -C disk -P wwid` command to find out whether the disks of the VM are presented to both VSPs. If you find stale DSFs and stale entries in your Integrity VM device management database, use the `insf -e` command and the `hpvmdevmgmt` command to repair the HP-UX VSP system.

WARNING! Do not physically rearrange controllers on the host systems to make the paths the same. This can lead to stale DSFs and stale entries in the Integrity VM device management database.

Do not mark disks `SHARE=YES` for devices assigned to VMs that migrate (unless more than one VM shares the storage on the same VSP). Marking a device `SHARE=YES` can lead to more than one VM using the device at the same time and can lead to disk corruption.

Selecting physical HBA ports during migration with NPIV HBAs

Starting vPars and Integrity VM v6.2, the `hpvmmigrate` command attempts to take into account redundancy and multi-pathing aspects in addition to balancing the count of NPIV HBA across available HBA ports while selecting the HBA ports on which a guest NPIV HBAs will be placed.

The following rules apply:

- The pFCs chosen on the target host depends on the following:
  - The number of pFCs on the target host.
  - The number of active NPIV HBAs that each of them already has.
  - The FC connectivity of the pFCs to the FC fabric (that is, to which physical switch and fabric they are connected).
- For each guest NPIV HBA, an HBA port on the target is selected based on the following criteria:
  - An attempt is made to distribute the NPIV HBAs of the guest, first across eligible HBA cards, and then across eligible HBA ports on the target.
  - Of these, when selecting an HBA port, the first preference is for one that is connected to the same physical switch as on the source host.
  - Of all such eligible HBA ports, the first preference is for the one with the least number of active NPIV HBA instances.

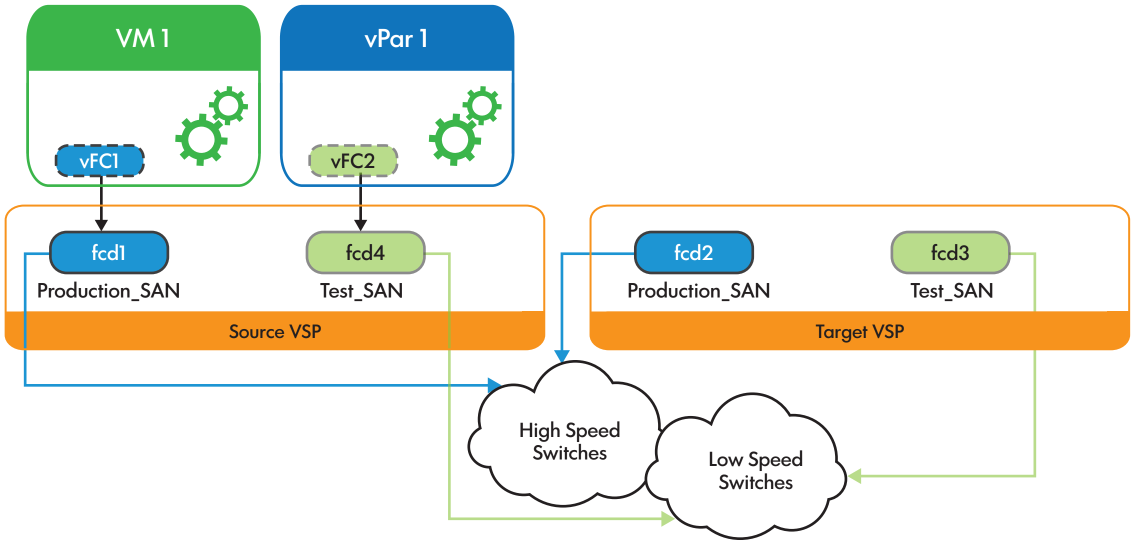
From vPars and Integrity VM v6.3.5 onwards, the administrator can label NPIV resources to achieve and maintain SAN level isolation during migration of vPars and VM guests across VSPs. For more information about configuring labels for predictable placement of NPIV HBAs on the target FC ports of VSP during migration, see “NPIV pools” (page 106).

SAN isolation with NPIV HBA during guest migration

This section describes the configuration steps required to ensure SAN level isolation across migrations.

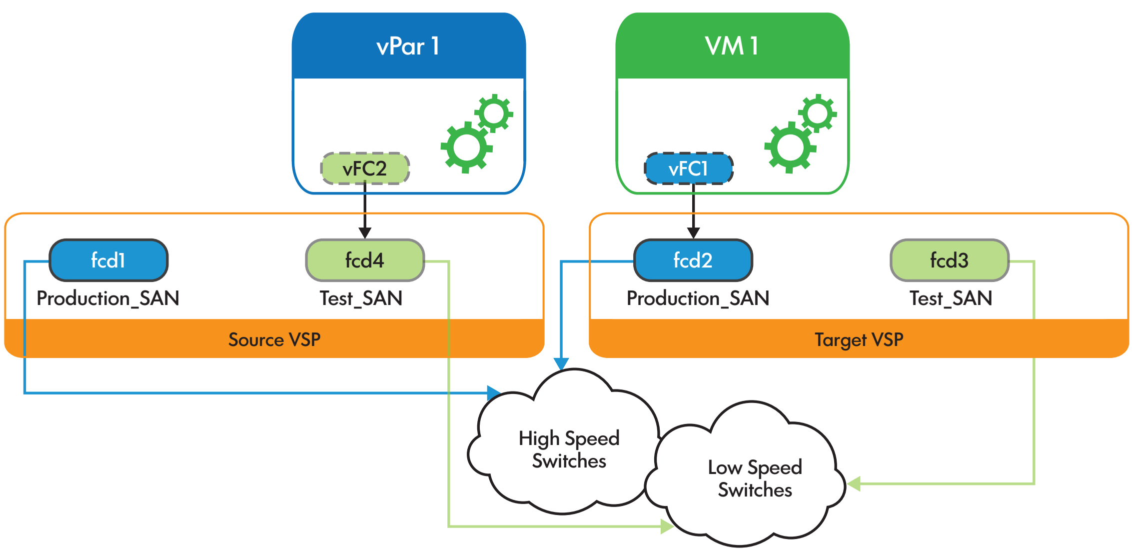
1. On all the VSPs on which a set of vPars and VM guests with NPIV resources can potentially run, identify the NPIV capable FC ports that need to be grouped together (this can be based on bandwidth capabilities of the FC ports or the fabric to the FC port is connected).
2. On each of the VSPs, add these ports into the NPIV pool and assign an appropriate label. Ports with similar characteristics on all VSPs must have the same label associated with them.
3. By not assigning a FC port to any group, the administrator can indicate that when nothing is available in a specific pool, an eligible port can be picked from the DEFAULT_POOL.

Figure 19 Before migration—SAN isolation with NPIV HBA



In Figure 20, fcd1 has been labeled with “Production_SAN” and fcd4 as “Test_SAN”. VM1 has an NPIV HBA that is backed by the FCD device `/dev/fcd1`. On the target VSP, fcd2 is connected to the high speed switches, whereas fcd3 is connected to a low speed switch. To ensure that as far as possible, the NPIV HBA belonging to VM1 gets placed on fcd2 during migration, fcd2 on the target VSP is labeled such that it matches label of fcd1 on the source VSP.

Figure 20 After migration—SAN isolation with NPIV HBA is preserved



As seen in Figure 20, post migration, the NPIV HBA of VM1 has been placed on fcd2.

NOTE:

- NPIV capable FC ports can be added or removed from pools while they are in use by active guests.
- The label associated with an NPIV capable FC port on the VSP can be changed while the port has vHBAs configured on it and in use by active guests. These changes do not impact the operation of the vPar or VM in any way.
- The migration fails ONLY if there is no NPIV resource with matching connectivity requirements (as explained in “Selecting physical HBA ports during migration with NPIV HBAs” (page 218)) amongst all the NPIV capable FC ports on the target VSP.

When the source and target VSPs have FC ports with labels configured, then in addition to meet the selection criteria as explained in “Selecting physical HBA ports during migration with NPIV HBAs” (page 218), the following rules apply for selection of FC port on the target VSP:

- If the NPIV HBA being considered for placement is part of the NPIV pool on the source, then,
  - Firstly, the rules (explained in “Selecting physical HBA ports during migration with NPIV HBAs” (page 218)) will be applied on the NPIV resources with the matching label on the target.
  - Next, the rules will be applied on NPIV resources with no NPIV label on the target VSP.
  - Lastly, the rules will be applied on NPIV resources with any NPIV label.
- If the vFC being considered is not a part of the NPIV pool on the source, then,
  - Firstly, the rules will be applied on the NPIV resources with no NPIV label.
  - Next, the rules will be applied on the remaining NPIV resources with some NPIV label or other.

NOTE:

- Migration will not fail if on the target VSP, a NPIV capable FC port with a matching label is unavailable.
- Migration will only fail if there is no NPIV FC port on the target matching selection rules as in “Selecting physical HBA ports during migration with NPIV HBAs” (page 218).

Bandwidth management for NPIV HBAs during guest migration

If the pFC on the source is labeled, the migration algorithm attempts to select a physical FC (pFC) on the target within the same NPIV pool as the source virtual FC (vFC). It also checks the bandwidth availability of the authorized pFCs. If both the FC ports (FC1 and FC2) have the required bandwidth on the target VSP, and FC1 label matches with the source VSP, but FC2 does not match. In this case, the preference is given to FC1.

When the NPIV HBAs are configured with bandwidth entitlement, then to meet the selection criteria rules, which are explained in “Selecting physical HBA ports during migration with NPIV HBAs” (page 226), the following rules apply for selection of FC port on the target VSP:

Case 1

If vFC is considered as part of an NPIV pool on the source and the vFC has bandwidth entitlement, and the migration option `ignore_npiv_entitlement` is disabled then:

- These rules are applied on the NPIV resources on the target that belong to the matching NPIV pool, and also that meets the bandwidth requirement. The number of active bandwidth entitled NPIV HBA that exists on the pFC are also checked.
- If appropriate pFC is not found in the NPIV pool on the target, then apply these rules, which do not belong to any pool.
- The same checks are applied on NPIV resources that belong to any other pool to find an appropriate pFC that meets the bandwidth requirement.
- If no such authorized pFC is found, migration fails.

If the vFC being considered is not part of an NPIV pool on the source, then:

- These rules are applied on the NPIV resources on the target that do not belong to any pool.
- If appropriate pFC is not found, then these rules are applied on the remaining NPIV resources that belongs to any other pool to find an appropriate pFC that meets the bandwidth requirement.
- If no such authorized pFC is found, migration fails.

Case 2

If the vFC is considered is part of an NPIV pool on the source and the vFC has the bandwidth entitlement, and the guest migration option `ignore_npiv_entitlement` is enabled then:

- These rules are applied on the NPIV resources on the target that belong to the matching NPIV pool to find appropriate pFC, and to create the vFC with the matching bandwidth entitlement. The number of active bandwidth entitled NPIV HBA is also checked.
- If appropriate NPIV resource is found on the target with the matching pool, but it does not have the bandwidth to suffice the source vFC, or the limit on number of active NPIV HBA with bandwidth entitlement is reached, then more preference is given for the NPIV pool label match. In this case, the bandwidth entitlement of the vFC is ignored, and the HBA is created without bandwidth entitlement on physical HBA that has the Quality of Service (QOS) mode disabled.
- If appropriate NPIV resource is not found in the NPIV pool on the target, then apply these rules on NPIV resources that do not belong to any pool.
- These rules are applied on NPIV resources that belong to any other pools to find appropriate pFC.
- If no matching bandwidth entitlement or limit on number of active NPIV HBA with bandwidth entitlement is reached, then the migration succeeds with best selected pFC ignoring the bandwidth entitlement. vFC is created on pFC that has the QOS mode disabled.

If the vFC being considered is not part of an NPIV pool on the source, then:

- These rules are applied on the NPIV resources that do not belong to any pool.
- These rules are on the remaining NPIV resources that belongs to some pool or the other.

Migration across disjoint fabric

Migration across “disjoint fabrics” refers to migration of vPar or VM guest configured with NPIV HBAs across two VSP hosts that are connected to a different set of FC switches or SAN fabrics.

Prior to vPars and Integrity VM v6.4, the prerequisite for migration of guests configured with NPIV was the source and the target VSPs must have FC ports connected to the same SAN fabric. Starting with vPars and Integrity VM v6.4, the `hpvmmigrate` command will allow migration of a guest configured with NPIV even when the source and target VSP hosts are connected to disjoint fabrics. This capability comes in handy when trying to migrate a guest configuration across sites configured for disaster recovery.

Each new guest option, “npiv_migration” is introduced with `hpvmmodify` and `vparmodify`commands to specify that the VM guest/ vPar migration is being attempted across disjoint fabrics. The selection criteria for placement of NPIV HBAs on the target VSP host differs based on this new option.

When the guest option ‘npiv_migration’ is set to fabric, the selection criteria for placement of NPIV HBAs during vPar or VM migration will be connectivity to the same fabric. In this case, the NPIV migrations work in the same way when vPars or Integrity VM v6.3.5 is used, and the source and the target VSPs have to be connected to the same SAN fabric for migrations to succeed.

When the guest option ‘npiv_migration’ is set to label, the selection criteria for placement of NPIV HBAs during the vPar or VM migration will be matching NPIV labels. Appropriate physical FC ports on the source and target VSP hosts must have matching labels assigned to them. This selection criteria must be used for NPIV migrations across VSP hosts connected to separate or disjoint SAN fabrics.

This section describes the configuration steps to migrate the guest to the target VSP when the fabric is different or disjoint:

1. All the VSPs where a set of vPars and VM guests with NPIV resources can potentially run and identify the NPIV capable FC ports need to be grouped together based on the storage connectivity.
2. Add the ports into the NPIV pool and assign an appropriate label to each of the VSPs. Ports that are connected to the same storage or replicated storage at the target must have the same labels associated with them.
3. Set the guest option ‘npiv_migration=label’ using the `hpvmmodify` or `vparmodify` command.
4. For migration across disjoint fabric, the NPIV labels are used to identify the FC ports for the placement of the NPIV HBA on the target VSP. If the FC ports are not labelled appropriately, the vPar or VM guest startup might fail on the target VSP.

NOTE:

1. If npiv_migration is set to label, all FC ports used by the migrating VM guest or vPar must be labeled. If not, the migration will fail.
2. If npiv_migration is set to label and a VM guest is migrated online, the npiv_migration option will be ignored and behavior defaults to check for fabric connectivity.
3. By default, npiv_migration is set to fabric.

When migrating across disjoint fabrics, all the rules explained in “Selecting physical HBA ports during migration with NPIV HBAs” (page 218) will be applied on the NPIV resources with the matching label on the target for placement of NPIV HBAs.

NOTE:

- Migration will fail if an NPIV capable FC port with a matching label is unavailable on the target VSP.
- The selection criteria applies only to offline vPar or VM migrations.

Using NTP on the VM or vPar guests

Hewlett Packard Enterprise strongly recommends using NTP for online VM and vPar migration environments. Each guest must include all potential VSPs as servers in `ntp.conf` file so that the current local VSP can be used as a time source. Whether migrating or not, guests must not be used as time servers. To maintain reliable time synchronization on a guest, it might be necessary to reduce the NTP polling interval, so the guest checks the time more frequently with the NTP server.

Marking a guest not runnable

On all VSPs that have a VM or vPar configured, the VM or vPar must be marked `Runnable` on only one VSP at a time. While migrating online guests, unexpected errors or guest resets or aborts must not cause your guest to be incorrectly marked `Runnable` or `Not Runnable`.

To verify the `Runnable` state of a VM or vPar, use the `hpvmstatus` command to see that the guest is `Runnable` on only one VSP and `Not Runnable` on all other VSPs. If the `Runnable` state of a VM or vPar is not correct on a VSP, use the `hpvmmodify` command to correct it.

To mark a guest `Not Runnable`, use the following command:

```
# hpvmmodify -P guestname -x runnable_status=disabled
```

To mark a guest `Runnable`, use the following command:

```
# hpvmmodify -P guestname -x runnable_status=enabled
```

WARNING! You must be careful when marking a guest `Runnable` when it was previously `Not Runnable`. Ensure this guest is `Not Runnable` and definitely not actually running on any other VSP.

Inter family online migration support

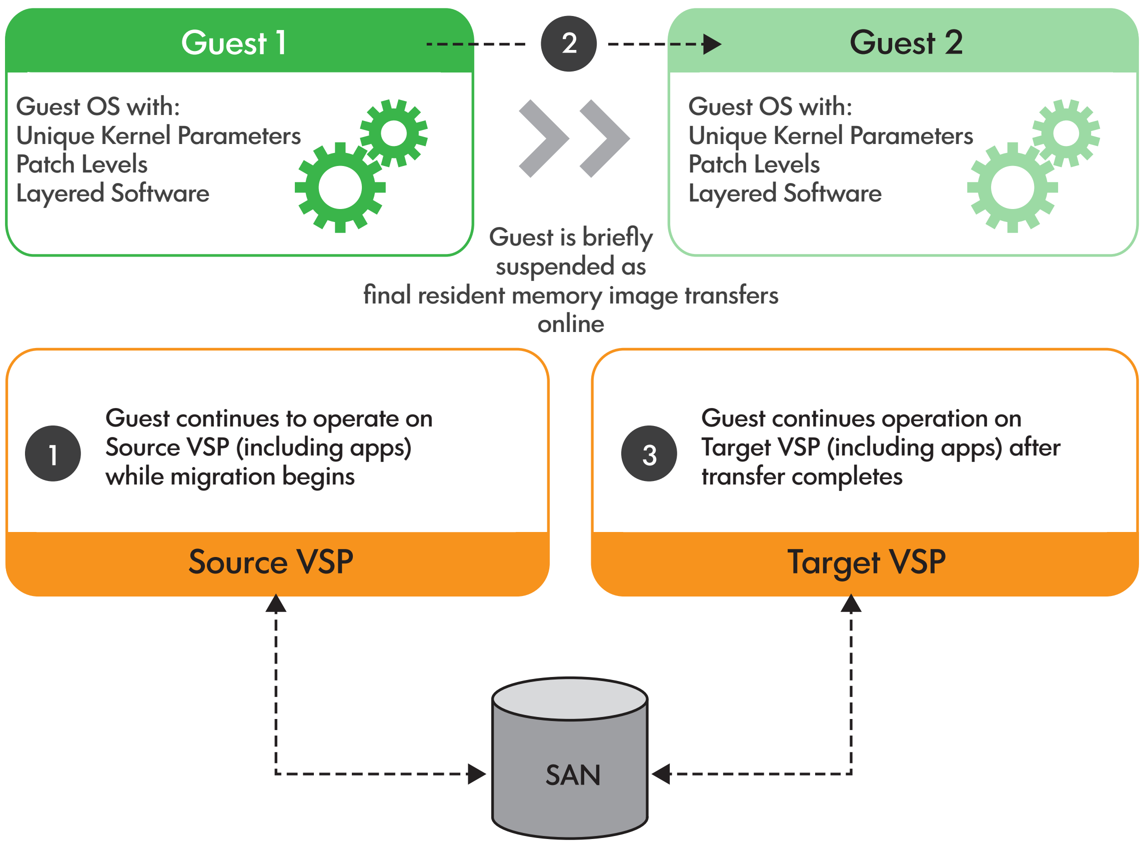
For more information, see “Inter family online migration support” (page 229).

13 Migrating VMs

VSP requirements and setup

All the latest HP-UX patches that Integrity VM requires, and any other required Integrity VM patches must be installed. For more information about vPars and Integrity VM installation, including supported VSP operating system versions, patches, and other system requirements, see *HP-UX vPars and Integrity VM Release Notes*, available at **http://www.hpe.com/info/hpux-hpvm-docs**. Required patches are available at **http://www.hpe.com/support/hpesc** .

Figure 21 Online VM migration from source to target



VSP processors for online migration

VSPs can be different Integrity server models with different number of processors, different I/O adapters and configurations, different amounts of memory, different firmware revisions, and so on. In particular, guests can migrate between radically different size, capacity, and power VSPs. However, for online migration, all the eligible VSP servers in a group must have equivalent architecture implementations. The processors on the source and destination VSPs must all be within one of the following groups:

- All Integrity VM supported variants of the Itanium 2 processor before 9000
- Itanium 2 9000 and the Itanium 2 9100
- Itanium 9300
- Itanium 9500

NOTE: Starting with HP-UX vPars and Integrity VM v6.3, the interprocessor family migration between 9300 and 9500 series is supported. For more information about online migration support, see “Inter family online migration support” (page 223).

Different processor frequencies and cache sizes are supported for OVMM. Table 30 (page 225) lists the recent Itanium processors showing different values for processor family.

Table 30 Itanium processor families

Family	Model	Series
31	0	Itanium 2
31	1	Itanium 2
31	2	Itanium 2
32	0	Itanium 9000
32	1	Itanium 9100
32	2	Itanium 9300
33	0	Itanium 9500

You can lookup processor `Family` as shown in the following example output from the `machinfo -v` command. (As more processors families and models are added, more specific capability requirements might be necessary.) The systems `host19` and `host20` in this example are compatible for migration, because they have the same processor `family (32)` and therefore they belong to the same processor group as defined earlier.

```
# hostname
host19
# machinfo -v
CPU info:
 12 Intel(R) Itanium 2 9000 series processors (1.6 GHz, 24 MB)
          533 MT/s bus, CPU version C2
          24 logical processors (2 per socket)

          Vendor identification:        GenuineIntel
          Processor version info:       0x0000000020000704
                  Family 32, model 0, stepping 7
          Processor capabilities:       0x0000000000000005
                  Implements long branch
                  Implements -byte atomic operations
          . . .

# hostname
host20
# machinfo -v
CPU info:
  4 Intel(R) Itanium 2 9000 series processors (1.6 GHz, 24 MB)
          533 MT/s bus, CPU version C2
          8 logical processors (2 per socket)

          Vendor identification:        GenuineIntel
          Processor version info:       0x0000000020000704
                  Family 32, model 0, stepping 7
          Processor capabilities:       0x0000000000000005
                  Implements long branch
                  Implements -byte atomic operations
          . . .
```

VM requirements and setup

Online VM Migration is supported on HP-UX 11i v2 and HP-UX 11i v3 guests. All memory sizes and virtual CPU configurations for the current version of Integrity VM are supported. As with all guest OS installations, the latest revision of a matching guest kit must be installed.

Setting online migration phase time-out values

To protect the workload of the guest, the online migration software limits the amount of time spent in each migration phase. The phases of an online migration are:

- Initialization phase— Establishes connections, carries out various checks, starts the target guest.
- Copy phase— Tracks writes to guest memory and copies all of guest memory.
- I/O quiesce phase— Queues new I/O requests and waits for outstanding I/O to complete.
- Frozen phase— Stops the virtual CPUs and copies modified memory and guest state.

For example, if a guest stops I/O to storage for long, it can experience I/O errors and applications can fail or the operating system can crash. If a guest is frozen for long, external network connections to the guest can time out and network connections can be dropped.

Network time-outs are troublesome for certain UDP applications that are not resilient enough to tolerate packets being delayed and dropped. If you run UDP applications that assume fast network packet turnaround, you must reduce the frozen phase time-out value, which might cause online migrations to abort more often. However, it will preserve the integrity of the network connections to the guest. The trade-off is that your migration might abort if conditions are not appropriate for fast and efficient migrations.

If necessary, you can adjust the following migration time outs with the `hpvmmodify -x` command:

- migrate_init_phase_timeout— Specifies the maximum number of seconds the online migration spends during the initial phase of the migration. The default is 90 seconds.
- migrate_copy_phase_timeout— Specifies the maximum number of seconds the online migration spends during the full-copy phase. The default is infinite.
- migrate_io_quiesce_phase_timeout— Specifies the maximum number of seconds the migration spends during the quiesce phase. The default is 15 seconds.
- migrate_frozen_phase_timeout— Specifies the maximum number of seconds the migration spends during the freezing phase. The default is 60 seconds.

Sharing guest storage device

For more information on sharing guest storage device, see "Sharing guest storage device" (page 217).

Selecting physical HBA ports during migration with NPIV HBAs

For more details on selecting physical HBA ports during migration with NPIV HBAs, see "Selecting physical HBA ports during migration with NPIV HBAs" (page 218).

Using NTP on the VM guests

For more information on using NTP on VM guests, see "Using NTP on the VM or vPar guests" (page 222).

Marking a guest not runnable

For more information on marking a guest not runnable, see "Marking a guest not runnable" (page 223).

Examples of the `hpvmmigrate` command

The following command displays the version number of the `hpvmmigrate` command:

```
# hpvmmigrate -v
hpvmmigrate: Version B.06.30
```

Online Migration

The OVMM feature is initiated with the `-o` option to the `hpvmmigrate` command. The following example shows migration of a guest to another VSP. The guest name is vm3. The target VSP is called host2, and the private network of the target VSP is called `host2-hpvm-migr` (that is, `host2-hpvm-migr` is an alias for the private network defined in `/etc/hosts`).

NOTE: The `hpvmmigrate` command does not check whether you are using a private network to migrate your guest. Using a private network is important for security, and to maintain the performance of public network of your site.

To migrate guest `vm3` to VSP `host2`:

```
# hpvmmigrate -o -P vm3 -h host2
```

The `hpvmmigrate` command displays status as various phases of migration completion. Output messages that are indented from the left margin are from the remote target VSP.

To prevent data getting over written on the SAN storage of the guest, the Integrity VM software helps to prevent you from accidentally running the same guest on more than one VSP simultaneously. If the `hpvmmigrate -D` option is not specified, the guest is marked `Not Runnable (NR)` on the source VSP after online migration is finished. This prevents the VM from booting on the original source VSP while it is running on the target VSP. If the `hpvmmigrate -D` option is used, unpresent the SAN storage of the guest from the source VSP as soon as migration completes, thus avoiding accidental usage of the storage on that VSP.

Using the `hpvmstatus` command to view migration details

To view the current state of all VMs on the VSP, use the `hpvmstatus` command. Many states are related to online VM and vPar migration:

- `On (OS)` — The guest is On and running the guest operating system. It is considered `Runnable`.
- `Off (NR)` — The VM is not booted and is `Not Runnable`.
- `On (MGS)` — The guest is On and running a guest operating system. It is the source of an online migration to another VSP.
- `On (MGT)` — The VM is On, but not yet running a guest operating system. It is the target of an online migration from another VSP.

You can use the `hpvmstatus -P` and `-V` options to get detailed migration status about a particular VM. If the guest is actively migrating, the `hpvmstatus` command shows the phase information about online VM and vPar migration phases.

Options to `hpvmmodify` command for online migration

To change the online migration phase timeout values, you can use the `hpvmmodify -x` option. For a list of time-out phases, see "Setting online migration phase time-out values" (page 226).

Use the `hpvmmodify -x online_migration=disabled` option to prevent a particular VM from migrating online. This is important if the guest is running software that is sensitive to external network monitoring with short timing intervals, such as Serviceguard.

NOTE: A transient network error might cause the vswitch connectivity check of the `hpvmmigrate` command to report a failure. If the connectivity check fails, retry the migration by rerunning the `hpvmmigrate` command.

If the network connectivity check of the `hpvmmigrate` command continues to fail, verify the vswitch and network configuration, and test connectivity with the `nwmgr` command.

If the vswitch connectivity required by the guest on the target VSP is properly configured and verified, you can use the `hpvmmigrate -w` option to bypass vswitch connectivity checks.

The Online VM migration feature is supported with Serviceguard packaged guests. For more information, see *Serviceguard Toolkit for Integrity Virtual Servers User Guide* at **http://www.hpe.com/info/hpux-serviceguard-docs**.

Using the `hpvminfo` command in the guest

The `hpvminfo` command is a part of the Integrity VM guest kit and must be installed on all the guests. In the case of Integrity VM v6.3, if VirtualBase B.06.30 is installed on the guest, the guest kit need not be installed. You can use the `hpvminfo -V` option to view information about the guest and the current VSP.

Following is a shell script using the `hpvminfo -M` option (for machine-readable output) that you can run on any Unix guest to know when an online migration has occurred. The script gets the guest name (G), and the current host (H1), and then begins an infinite loop testing and reporting whether the host on which it is running has changed. Terminate the shell script with a ^C.

```
G=$(hpvminfo -M | awk -F : '{print $12;}')
H1=$(hpvminfo -M | awk -F : '{print $7;}')
echo $(date) $G: Current host is $H1
while true
do
   H2=$(hpvminfo -M | awk -F : '{print $7;}')
   if [ "$H1" != "$H2" ]; then H1=$H2; echo $(date) $G: host is now $H2; fi
done
```

Following is a sample output from this script:

```
Tue Aug 26 10:52:39 PDT 2008 vm6: Current host is host2
Tue Aug 26 10:53:36 PDT 2008 vm6: host is now host1
Tue Aug 26 10:54:28 PDT 2008 vm6: host is now host2
Tue Aug 26 10:55:19 PDT 2008 vm6: host is now host1
```

Restrictions and limitations of online VM migration

Administrators must configure certain aspects of VSPs and guests for online migration capability. Integration with automated workload placement, management, and load balancing tools are not supported. More automated and more convenient management of distributed Integrity VM guests might follow in subsequent Integrity VM releases.

A dedicated high-speed network must not be on the data center, work site, company, or “public” LAN. Online migration can also swamp the network while a migration is in progress. Using the network site for migration traffic can also create peaks of network activity that might affect network performance. Using a high-speed network is desirable to minimize guest memory transfer time and allows your guest to migrate smoothly.

The following devices are supported for guest storage for online guest migration:

- Whole disk backing stores consisting of SAN LUNs
- Ejected file-backed DVDs
- SLVM volumes
- NFS-mounted backing stores

- NPIV backing stores
- Cluster DSF
- DMP Nodes

File backing stores that are not NFS-mounted and attached devices are not supported for online guest migration.

Following are the mandatory conditions while migrating a vPar or VM with cDSF as backing store:

- The source and the destination must belong to the same Cluster DSF group. (cmsetdsfgroup(1M))can be used to find whether the source and destination belong to same Cluster DSF group or not.
- The source and destination must have HPVM v6.3 or later.

Only one online migration to or from a VSP can be performed at a time. Also, be aware of the state of the guest while migrating it online. If the guest is in the On (EFI) state and no guest operating system is booted, the online migration fails with an error. If the guest is shutting down, restarting or crashing while migrating, the online migration aborts when the hpvmmigrate command can no longer communicate with the guest.

For more information about online migration, see HP-UX vPars and Integrity VM v6.3 Release Notes available at **http://www.hpe.com/info/hpux-hpvm-docs**.

NOTE: Online migration is not supported with DIO devices.

In the case of NPIV, for migration across VSPs on a shared FC fabric, the target VSPs must have FC ports that can access all the targets ports that the NPIV HBA has access to on the source VSP. In case of offline migration across disjoint fabrics, appropriate FC ports on the source and target VSP must be labeled identically for proper placement of NPIV HBAs.

If zoning is required, the FC switch must use World-Wide-Name (WWN) based zoning as opposed to Port based zoning.

Hewlett Packard Enterprise recommends that you do not use port based zoning or mixed zoning (using both WWN based and port based zoning on the same fabric), because migration might fail under certain scenarios.

Inter family online migration support

Starting with vPars and Integrity VM v6.3, Online migration of VM guests between VSP systems running Itanium 9300 processor and Itanium 9500 processor is supported. In other words, a VM guests running on a BL860c i2 or SD2 i2 or rx2800 i2 can be migrated online to a system BL860c i4 or SD2 i4 or rx2800 i4 systems. Additionally, online migration from an Itanium 9500 processor system to Itanium 9300 processor system is also supported.

The following configuration must be set up to enable this feature:

1. VSP running on Itanium 9500 processor
   a. Ensure the VSP is installed with vPars and Integrity VM v6.3 or later.
   b. Enable the tunable `mdep_reduce_rse_size=1`

      To set the kernel tunable, enter the following:

      ```
      # kctune mdep_reduce_rse_size=1
      ```

After enabling the tunable, the VSP must be restarted for the tunable to take effect.

2. VSP running on Itanium 9300 processor
   a. vPars and Integrity VM v6.2 with PHSS_43648 (PK2) or vPars and Integrity VM v6.3 or later.

⚠ **WARNING!** This feature must not be attempted on any other VSP configurations other than those listed.

3. Integrity VM configuration

   Enable the hpvm tunable `ogm_tukwila_poulson`. To set the tunable, enter the following `hpvmmodify -p <guest_id> -x ogm_tukwila_poulson=enabled` command on the desired guest. You must enable this tunable before starting the guest. If the tunable is enabled on a live guest, the change will be updated in the guest configuration; however, it will take effect only on the guest next reboot.

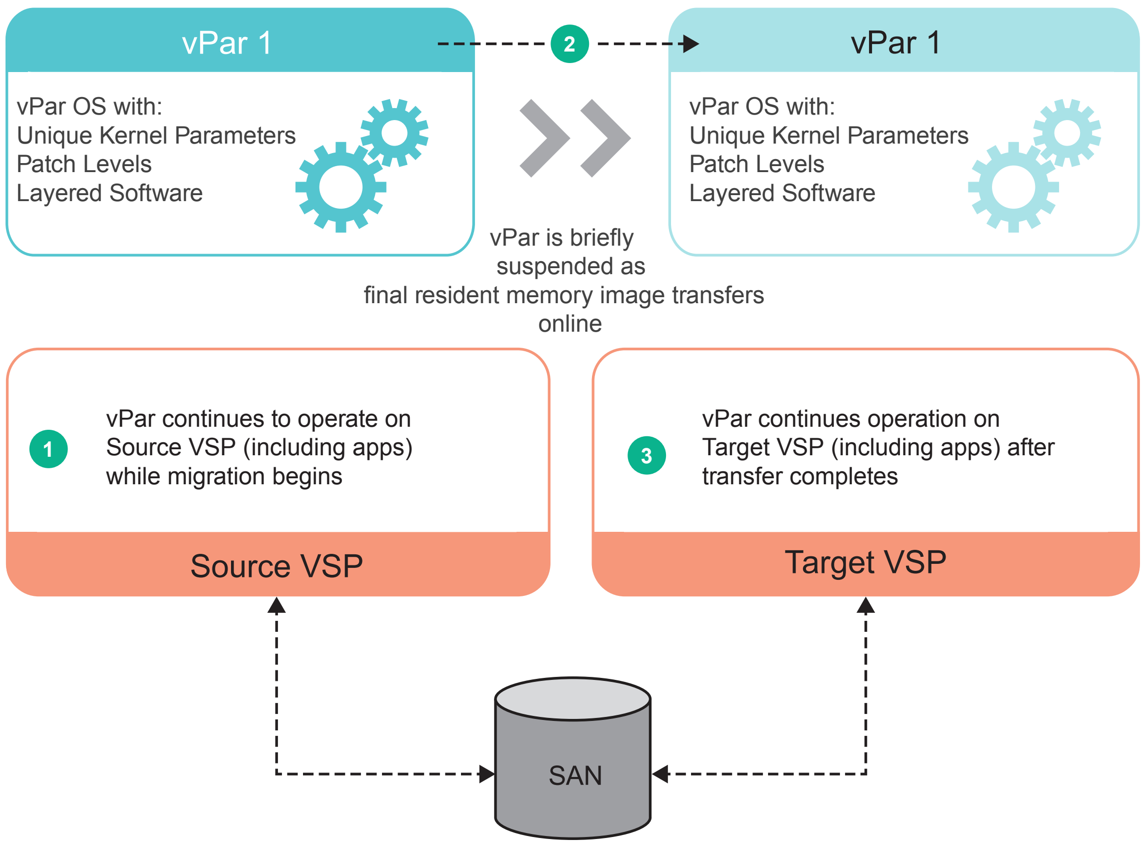
NOTE: After an inter family Online guest migration from Itanium 9300 to Itanium 9500 systems, the following limitations must be taken care:

1. The `caliper(1)` command can fail or show incorrect data on the Itanium 9500 systems. Hence, Hewlett Packard Enterprise recommends not to run the `caliper` command in the VM guest after guest migration.
2. The `machinfo(1)` command shows the source guest data. Upto vPars and Integrity VM v6.2, online guest migration was only supported among same family processors. With this feature, as online guest migration is being supported across processor family, running the `machinfo` command on the Itanium 9500 system after migration still displays it as an Itanium 9300 system.
3. After enabling `ogm_tukwila_poulson` tunable, if the guest is offline migrated to a VSP with vPars and Integrity VM version lower than 6.2 PK2 or the VSP is downgraded to a lower version, you can use the `hpvmmodify -P <Guest name> -x tunables=otpe=default` command to unset the tunable. This command removes the `ogm_tukwila_poulson` tunable entry from the configuration file. An attempt to perform Itanium 9300 systems to Itanium 9500 systems online guest migration from v6.2 base or v6.2 PK1 to v6.3 with the tunable entry in the guest configuration may result in the guest panic or online guest migration to abort.

14 Migrating vPars

The online vPar migration feature enables user to move the running vPar, its OS, and its applications to an identical VSP system without service interruption. While the vPar is moved from one VSP to another, the guest OS and all of its applications remain active, without requiring an OS reboot or application restart. The I/O activity freezes for a minimal amount of time (depending on size of guest and other parameters), but never shuts down during the migration process.

Figure 22 Online vPar migration from source to target



VSP requirements and setup

For more information about configuring VSP, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.

VSP processors for online migration

For online vPar migration, VSPs need to be the same Integrity server models with same number of processors. Integrity server models may have different I/O adapters and configurations, different amounts of memory, different firmware revisions, and so on. In particular, vPars cannot be migrated online between radically different size, capacity, and power VSPs. For online migration, all the eligible VSP servers in a group must have equivalent architecture implementations. The processors on the source and destination VSPs must all be within:

- Itanium 9500

NOTE: Inter process family migration between 9300 and 9500 series is not supported for online vPar migration.

Different processor frequencies and cache sizes are not supported for online vPar migration.

Table 31 (page 232) lists the recent Itanium processors showing different values for processor family.

Table 31 Itanium processor families

Family	Model	Series
33	0	Itanium(R) Processor 9540
33	0	Itanium(R) Processor 9560

You can lookup processor Family as shown in the following example output from the `machinfo -v` command. (As more processors families and models are added, more specific capability requirements might be necessary.) The systems sd2-host1 and sd2-host2 in this example are compatible for migration, because they have the same processor family (33) and therefore they belong to the same processor group as defined earlier.

```
# hostname
sd2-host1

# machinfo -v
CPU info:
   Intel(R)  Itanium(R)  Processor 9540 (2.13 GHz, 24 MB)
   8 cores, 16 logical processors per socket
   6.39 GT/s QPI, CPU version D0
          Active processor count:
          7 sockets
          56 cores (8 per socket)
          56 logical processors (8 per socket)
          LCPU attribute is disabled

          Vendor identification:        GenuineIntel
          Processor version info:       0x0000000021000404
                  Family 33, model 0, stepping 4
...

# hostname
sd2-host2

# machinfo -v
CPU info:
   Intel(R)  Itanium(R)  Processor 9540 (2.13 GHz, 24 MB)
   8 cores, 16 logical processors per socket
   6.39 GT/s QPI, CPU version D0
          Active processor count:
          8 sockets
          64 cores (8 per socket)
          64 logical processors (8 per socket)
          LCPU attribute is disabled

          Vendor identification:        GenuineIntel
          Processor version info:       0x0000000021000404
                  Family 33, model 0, stepping 4
...
```

Private network setup

For more information on private network setup, see “Private network setup” (page 213).

Conventions for using target-hpvm-migr names for private networks

For more information on conventions for using target-hpvm-migr names for private networks, see “Conventions for using *target-hpvm-migr* names for private networks” (page 214).

NTP Usage on VSPs

For more information on using NTP on VSPs, see "Using NTP on VSPs" (page 215)

vPar requirements and setup

The guest OS must have the VirtualBase bundle installed to work in a VSP environment. For more information on installing VirtualBase on a vPar or VM Guest, see "Installing VirtualBase on a vPar or VM Guest" (page 28).

NOTE: For more information on configuring vPar, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.

Setting online migration phase time-out values

To protect the workload of the guest, the online migration software limits the amount of time spent in each migration phase. The phases of an online migration are:

- Initialization phase— Establishes connections, carries out various checks, and starts the target guest.
- Copy phase— Tracks write to guest memory and copies all of the guest memory.
- I/O quiesce phase— Queues new I/O requests and waits for outstanding I/O to complete.
- Frozen phase— Stops the virtual CPUs and copies modified memory and guest state.

For example, if a guest stops I/O to storage for long, it can experience I/O errors and applications can fail or the operating system can crash. If a guest is frozen for long, external network connections to the guest can time out and network connections can be dropped.

Network time-outs are troublesome for certain UDP applications that are not resilient enough to tolerate packets being delayed and dropped. If you run UDP applications that assume fast network packet turnaround, you must reduce the frozen phase time-out value, which might cause online migrations to abort more often. However, it will preserve the integrity of the network connections to the guest. The trade-off is that your migration might abort if conditions are not appropriate for fast and efficient migrations.

If necessary, you can adjust the following migration time outs with the `hpvmmodify -x` command:

- **migrate_init_phase_timeout**— Specifies the maximum number of seconds the online migration spends during the initial phase of the migration. The default is 180 seconds.
- **migrate_copy_phase_timeout**— Specifies the maximum number of seconds the online migration spends during the full-copy phase. The default is infinite.
- **migrate_io_quiesce_phase_timeout**— Specifies the maximum number of seconds the migration spends during the quiesce phase. The default is 200 seconds.
- **migrate_frozen_phase_timeout**— Specifies the maximum number of seconds the migration spends during the freezing phase. The default is 200 seconds.

Migrations might time out and must be restarted

For more information, see "Migrations might time out and must be restarted" (page 216).

Sharing guest storage device

For more information on sharing guest storage device, see "Sharing guest storage device" (page 217).

Selecting physical HBA ports during migration with NPIV HBAs

For more information on selecting physical HBA ports during migration with NPIV HBAs, see "Selecting physical HBA ports during migration with NPIV HBAs" (page 218).

Using NTP on the VM and vPar guests

For more information, see using NTP on the VM and vPar guests"Using NTP on the VM or vPar guests" (page 222).

Marking a guest not runnable

For more information, see marking a guest not runnable "Marking a guest not runnable" (page 223).

Examples of the `hpvmmigrate` command

The following command displays the version number of the `hpvmmigrate` command:

```
# hpvmmigrate -v
hpvmmigrate: Version B.06.40.00
```

Offline Migration — An example

The following example illustrates how to migrate the guest named VPAR1, residing on the host named HostA, to the target host (HostB). On the system named HostA, enter the following command:

```
# hpvmmigrate –P VPAR1 –h HostB
```

This example specifies:

- The name of the vPar (`-P VPAR1`)
- The name of the target host (`-h HostB`)

Online Migration — An example

The online vPar migration feature is initiated with the `–o` option to the `hpvmmigrate` command. The following example shows migration of a guest to another VSP host. The guest name is vpar1. The target VSP is called sd2-host2 and the private network of the target VSP is called host2–hpvm-migr (that is, host2–hpvm-migr is an alias for the private network defined in /etc/hosts).

NOTE: The `hpvmmigrate` command does not check whether you are using a private network to migrate your guest. Using a private network is important for security, and to maintain the performance of public network of your site.

To migrate guest vpar1 to VSP host sd2-host2:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar1                    1 VP   HPUX    On (OS)       32     2     1 65536 MB
vpar2                    2 VP   HPUX    On (OS)        8     1     1 65344 MB
vpar3                    3 VP   HPUX    On (OS)        1     1     1 8192 MB

# hpvmmigrate -P vpar1 -o -h sd2-host2
hpvmmigrate: Connected to target VSP using '15.213.244.52'
hpvmmigrate: Starting vPar/VM 'vpar1' on target VSP host '15.213.244.52'
    (C) Copyright 2000 - 2016 Hewlett-Packard Development Company, L.P.

hpvmmigrate: Init phase (step 4) - progress 0%
    ......Creation of VM minor device 1
    Device file = /var/opt/hpvm/uuids/3047965e-e03f-11e5-8010-84349712dd06/vm_dev
hpvmmigrate: Init phase (step 4) - progress 0%
    guestStatsStartThread: Started guestStatsCollectLoop - thread = 6

      allocating datalogger memory: FF800000-FFA00000 (2048KB) ramBaseLog 600000110be00000
      allocating firmware RAM (fff00000-100000000, 1024KB) ramBaseFw 600000110bd00000
    Starting event polling thread

    Online migration initiated by source 'sd2-host1' (targethost)
    Target:0: online migration started with encryption algorithm AES-128-CBC.
hpvmmigrate: Init phase (step 5) - progress 0%
    Target:1: online migration started with encryption algorithm AES-128-CBC.
    Target:2: online migration started with encryption algorithm AES-128-CBC.
hpvmmigrate: Init phase (step 22) - progress 60%
    Event: configuration file renamed to
/var/opt/hpvm/uuids/3047965e-e03f-11e5-8010-84349712dd06/vmm_config.current
hpvmmigrate: Init phase completed successfully.
hpvmmigrate: Copy phase completed successfully.
hpvmmigrate: I/O quiesce phase completed successfully.
hpvmmigrate: Frozen phase completed successfully.
```

```
hpvmmigrate: vPar/VM migrated successfully.
```

The `hpvmmigrate` command displays status as various phases of migration completion. Output messages that are indented from the left margin are from the remote target VSP.

To prevent data getting over written on the SAN storage of the guest, the Integrity VM software helps to prevent you from accidentally running the same guest on more than one VSP simultaneously. If the `hpvmmigrate -D` option is not specified, the guest is marked Not Runnable (NR) on the source VSP after online migration is finished. This prevents the vPar from booting on the original source VSP while it is running on the target VSP. If the `hpvmmigrate -D` option is used, remove the SAN storage of the guest from the source VSP as soon as migration completes, thus avoiding accidental usage of the storage on that VSP.

Using the `hpvmstatus` command to view migration details

To view the current state of all vPars on the VSP, use the `hpvmstatus` command. Many states are related to online vPar migration.

- On (OS)— The guest is On and running the guest operating system. It is considered Runnable.
- Off (NR)— The vPar is not booted and is Not Runnable.
- On (MGS)— The guest is On and running a guest operating system. It is the source of an online migration to another VSP.
- On (MGT)— The vPar is On, but not yet running a guest operating system. It is the target of an online migration from another VSP.

You can use the `hpvmstatus -P` and `-V` options to get detailed migration status about a particular vPar. If the guest is actively migrating, the `hpvmstatus` command shows the phase information about online vPar migration phases.

`hpvmmodify` options command for online migration

Setting phase time-out values

To change the online migration phase timeout values, you can use the `hpvmmodify -x` option. For more information on list of time-out phases, see “Setting online migration phase time-out values” (page 233).

Disable online vPar migration

Use the `hpvmmodify -x online_migration=disabled` option to prevent a particular vPar from migrating online. This is important if the guest is running software that is sensitive to external network monitoring with short timing intervals, like Serviceguard.

Enabling force_vpar_migration

During online vPar migration, the resource agent tries to maintain the same vPar topology on the target VSP as seen on the source VSP. If the resource agent cannot allocate resources based on the source topology of vPar, then migration is aborted.

You can force resource agent to allocate any resources of any topology on the target VSP if not able to get the requested source vPar topology. To enable this feature, you can enable `force_vpar_migration` option as follows:

```
# hpvmmodify -P vpar1 -x force_vpar_migration=enabled
```

This option is disabled by default. It can be changed dynamically on an active vPar. It does not have any effect on VM.

Using the `hpvminfo` command in the guest

The `hpvminfo` command is a part of the Integrity VM guest kit and must be installed on all the guests. You can use the `hpvminfo -V` option to view information about the guest and the current VSP.

Following is a shell script using the `hpvminfo -M` option (for machine-readable output) that you can run on any UNIX guest to know when an online migration has occurred. The script gets the guest name (G), and the current host (H1), and then begins an infinite loop testing and reporting whether the host on which it is running has changed. Terminate the shell script with a ^C.

```
G=$(hpvminfo -M | awk -F : '{print $12;}')

H1=$(hpvminfo -M | awk -F : '{print $7;}')

echo $(date) $G: Current host is $H1

while true
do
    H2=$(hpvminfo -M | awk -F : '{print $7;}')
    if [ "$H1" != "$H2" ]; then H1=$H2; echo $(date) $G: host is now $H2;
    fi
done
```

Following is a sample output from this script:

```
Thu Mar 17 22:26:59 IST 2016 vpar1: Current host is sd2-host1
Thu Mar 17 22:29:32 IST 2016 vpar1: host is now is sd2-host2
```

Multi-socket memory copy enhancement

To reduce the memory copy time for vPar with multiple dedicated processor cores, HPE has developed a multi-socket memory copy technique that creates multiple memory clients so that the memory copy can be performed in parallel. It significantly reduces the amount of time for the memory copy. This capability is enabled for vPar guests, but not available for VM guests.

In multi-socket memory copy, memory copy is done by threads that are bound to dedicated cores of VSP.

A maximum of four cores can be used for multi-socket memory copy.

Use the `hpvmhwmgmt (1M)` command to add extra cores to the VSP pool on both the source and target VSPs.

NOTE: Hewlett Packard Enterprise recommends to allocate equal number of cores to the VSP pool on both the source and target VSPs.

Hewlett Packard Enterprise recommends assigning at least two dedicated cores to both the source and target VSPs to take advantage of multi-socket memory copy enhancement, and improve performance of online vPar migration. For more information on benefits gained in performance while doing online vPar migration, see “Add extra cores to the VSP pool to take advantage of multi-socket memory copy” (page 238).

Restrictions and limitations of online vPar migration

Following restrictions and limitations are applicable to online vPar migration.

Memory restrictions

- Before migrating vPar, a minimum of 30% of vPar memory must be available. Run glance(1) to find the percentage of available vPar memory before migration.
- It is mandatory to configure vPar with 100% base memory. If not done, it may result in migration failure due to memory fragmentation, or other memory constraints.

- Online migration of memory is not supported on an online migrated vPar guest. It is supported after an online migrated vPar is rebooted.
- The minimum vPar memory supported for online vPar migration is 8GB. The maximum vPar memory supported for online vPar migration is 64GB.
- Up to four successive migrations are supported. Currently, there is no way to detect the number of successful successive online vPar migrations. This feature will be provided in a future release of the product.

Processor restrictions

- Online vPar migration is supported on Intel Itanium i4 processors. They must all report the same processor family output for the HP-UX command `machinfo -v`.
- Processor frequency should be identical on source and target VSPs.
- Hyper-threading setting must be same on both the source and target VSPs.
- The maximum number of vPar CPUs supported for online vPar migration is 32.

NOTE: Online vPar migration is not supported on Itanium i2 processors. Inter family online migration is not supported for vPars.

Platform restrictions

- Source and target platforms should be identical. For more information on the source and target VSP support matrix, see Source and destination VSPs for online vPar migration of *HP-UX vPars and Integrity VM v6.4 Release Notes* available at **http://www.hpe.com/info/hpux-hpvm-docs**.

Miscellaneous

- ktracer (1M) should not be run when an online vPar migration is in progress.
- loratune (1M) is not supported on an online migrated vPar.
- When an online vPar migration is initiated, cimserver is stopped and restarted either in case of abort, or after a successful migration, where diagnostic logs are removed. If there is offline vPar migration, diagnostic logs are retained. For more information, see **http://www.hpe.com/info/hpux-diagnostics-sfm-docs**.

NOTE: Restrictions and limitations applicable to online VM migration are also applicable to online vPar migration. For more information, see "Restrictions and limitations of online VM migration" (page 228).

Recommendations

Following guidelines are recommended to reduce the overall duration of an online vPar migration.

Increase the base page size of vPar

- Increasing base page size from 4k to 8k can reduce duration of online vPar migration by up to 20%.
- Increasing base page from 4k to 16k can reduce duration of online vPar migration by up to 40%.

Add extra cores to the VSP pool to take advantage of multi-socket memory copy

- Allocating a total of two CPU cores to the VSP's CPU pool can reduce the duration of online vPar migration by up to 20%.
- Allocating a total of four CPU cores to the VSP's CPU pool can reduce the duration of online vPar migration by up to 40%.

Private network setup

For more information, see "Private network setup" (page 213).

NOTE: For online migration, in addition to sharing the same LAN segment for normal guest connectivity, Hewlett Packard Enterprise recommends that the VSPs must be connected with a private 10 GbE (or faster) network for efficient VSP-to-VSP communications and for secure guest memory transfer.

Memory considerations

HPE recommends that customers ramp down their workloads before initiating live migration. Initiating online vPar migration with heavy load may also result in a migration failure.

It is also possible, though less likely, that systems that were running with a heavy load earlier and are running idle now, may see migration failures due to memory fragmentation. These restrictions will be removed in a future release of the product.

15 Managing vPars and VMs using CLI

To manage a vPar and VM guest, connect to the vPar and VM guest using a remote connection, and use the operating system administration procedures appropriate to the guest OS. vPars and Integrity VM provides utilities for managing vPars and VM guests from the VSP and from inside the vPar and VM guest. This chapter describes how to manage vPars and VM guests using Integrity VM commands and utilities.

Monitoring guests

To view the information about all the vPars and VM guests configured on the VSP, enter the `hpvmstatus` command.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type    OS Type  State   #VCPUs #Devs  #Nets   Memory
==================== ===== ======= ======= ======  ===== =====  ======  ========
vPar0002                 2   VP     HPUX     Off         3      0      0      2048 MB
guest1                   3   SH     HPUX     On (OS)     4      0      0        10 GB
ux1                      1   SH     HPUX     Off         4      2      1         3 GB
```

The vPar and VM guest status is displayed in the `State` column and indicates whether the vPar or VM guest is powered off or on. When the vPar or VM guest is on, the status also includes one of the following:

- `EFI` indicates the vPar or VM guest is running normally in EFI.
- `OS` indicates the vPar or VM guest is running normally in the operating system.
- `ATTN` indicates the guest is not responding to interrupts.

Table 32 (page 239) lists the options that can be used with the `hpvmstatus` command.

Table 32 Options to the `hpvmstatus` command

Option	Description
`-v`	Displays the version of the Integrity VM product that is running on the VSP.
`-V`	Displays detailed information about the specified VM or about all the VMs if you do not specify one using either the `-p` or `-P` option.
`-M`	Specifies the display output to be in machine-readable format.
`-X`	Specifies the display output to be in XML format.
`-P` *`vm-name`*	Specifies the name of the VM.
`-p` *`vm-number`*	Specifies the number of the VM.
`-D`	Displays the resource allocated to the specified VM. You must include either the `-p` option or the `-P` option.
`-e`	Displays the event log for the VSP or the specified VM. The event log records all changes to VM configurations.
`-r`	Displays the memory and virtual CPU resource allocation for the VMs (or for the specified VM if you use the `-p` option or the `-P` option). This option displays the entitlement and virtual CPUs parameters configured for the VM and the current usage of those resources.
`-d`	Displays the devices allocated to the VM you specify using either the `-p` option or the `-P` option.
`-S`	Displays the scheduler mode for the VSP. `CAPPED` indicates that gWLM is managing the node. NORMAL indicates that the node is not being managed by gWLM.
`-s`	Displays the current VSP resources.
`-m`	If Serviceguard is installed, displays information about the multiple-server environment.

Table 32 Options to the `hpvmstatus` command *(continued)*

Option	Description
`-R`	Displays the resource reservation settings of the VMs.
`-L`	Displays the changes from the current configuration.
`-i`	When used with the `-P` option, prints statistics collected by the monitor.
`-C`	Displays whether the guests prefer clm, ilm, or none.
`-A`	Displays the guest configuration differences between the next start and the last start guest configurations.

For example, to view the detailed information about the `host1` VM, enter the following command:

```
# hpvmstatus -P vm001 -V
[Virtual Machine Details]
Virtual Machine Name : vm001
Virtual Machine UUID : dee4c3a6-33d8-11e2-8d00-3c4a92c4ef92
Virtual Machine ID : 3
Virtual Machine Label :
VM's Model Name : server Integrity Virtual Machine
VM's Serial Number : VM01247004
VM's Config Version : 6.20.0
VM's Config Label   : HPVM B.06.20 LR ccipf opt Thu Mar 14 2013 12h23m34s IST
Virtual Machine Type : Shared
Has reserved resources  : No
Configuration is active : Yes
Operating System : HPUX
OS Version Number :
State : Off
Start type : Manual
Console type : vt100-plus
Guest's hostname :
Guest's vNIC IP Preference :
Guest's IPv4 address :
EFI location          : /opt/hpvm/guest-images/common/efi
Pattern File location : /opt/hpvm/guest-images/common/patterns.vmmpat
vPar/VM revision : 14
Running on serverid : 0
Running on pid : 0
Application controllers : NONE
Distributed : 1
Effective serverid : 0
Graceful stop timeout : 120
Runnable status : Runnable
Modify status : Modify
Visible status : Visible

[Online Migration Details]
Online migration status : Enabled
Init phase timeout : 90 seconds
Copy phase timeout : Infinite
I/O quiesce phase timeout: 15 seconds
Frozen phase timeout : 60 seconds

[Suspend/Resume Details]
Suspend status : Disabled

[Remote Console]
Remote Console not configured

[Authorized Administrators]
Oper Groups :
Admin Groups :
Oper Users :
Admin Users :
        faizan

[Tunables]
```

```
[Virtual CPU Details]
Number Virtual CPUs : 1
Minimum Virtual CPUs : 1
Maximum Virtual CPUs : 32
Percent Entitlement : 10.0%
Maximum Entitlement : 100.0%

[Memory Details]
Total memory : 2 GB
Minimum memory limit : 512 MB
Maximum memory limit : 256 GB
Reserved memory       : 64 MB
Minimum reserved limit : 32 MB
Maximum reserved limit : 256 GB
VHPT Size : 1 MB
Overhead memory : 128 MB

[Dynamic Memory Information]
    NOTE: Dynamic data unavailable, configured values only
Type : driver
Minimum memory : 512 MB
Target memory : 2048 MB
Memory entitlement : Not specified
Maximum memory : 2048 MB


[Storage Interface Details]
Device type : disk
Adapter type : avio_stor
Ioscan format : 0/0/0/0.0.0
Bus : 0
Device : 0
Function : 0
Target : 0
Lun : 0
Physical Storage type : disk
Physical Device : /dev/rdisk/disk25


[Network Interface Details]

[Direct I/O Interface Details]

[Misc Interface Details]
Device type : serial
Adapter type : com1
Physical Storage type : tty
Physical Device : console
```

To view the VSP system resource, use the `-s` option with the `hpvmstatus` command. For example:

```
# hpvmstatus -s
[HPVM Server System Resources]

         vPar/VM types supported by this VSP = vPar, Shared
         Processor speed = 1330 Mhz
         Total physical memory = 32659 Mbytes
         Total number of operable system cores = 8
         CPU cores allocated for VSP = 1
         CPU cores allocated for vPars and VMs = 7
         CPU cores currently in use or reserved for later use = 1
         Available VSP memory = 1290 Mbytes
         Available swap space = 6750 Mbytes
         Total memory allocated for vPars and VMs = 27392 Mbytes
         Memory in use by vPars and VMs = 1600 Mbytes
         Available memory for vPars and VMs = 25792 Mbytes
         Available memory for 6 (max avail.) CPU VM  = 25088 Mbytes
         Available memory for 6 (max avail.) CPU vPar = 25664 Mbytes
```

```
Maximum vcpus for an HP-UX virtual machine = 7
Maximum vcpus for an OpenVMS virtual machine = 7
Maximum available vcpus for a VM = 6
Available CPU cores for a virtual partition = 6
Available entitlement for a 1 way virtual machine = 1330 Mhz
Available entitlement for a 2 way virtual machine = 1330 Mhz
Available entitlement for a 3 way virtual machine = 1330 Mhz
Available entitlement for a 4 way virtual machine = 1330 Mhz
Available entitlement for a 5 way virtual machine = 1330 Mhz
Available entitlement for a 6 way virtual machine = 1330 Mhz
```

Specific display output from some Integrity VM tools, such as the `hpvmstatus` command, is subject to occasional changes of form and content. Program scripts must always use machine-readable output options (for example, `hpvmstatus -M`) whenever available to avoid future script maintenance.

Monitoring Integrity VM performance

Guest and VSP performance information is displayed by the VSP `hpvmsar` command. In the `hpvmsar` command one of the displays can be shown in a GUI-type format with four different styles. For information about these styles, *hpvmsar* manpage. Some `hpvmsar` command options can be used only for HP-UX guests.

Table 33 Options to the `hpvmsar` command

Option	Display description
`-a`	Default Guest and Host CPU usage display in text or GUI modes for all running guests.
`-A`	Default Guest and Host CPU usage display in text or GUI modes for all guests whether they are running or stopped.
`-D`	Host to Guest Storage Utilization display
`-F`	Integrity VM core Memory Metrics display
`-G`	Guest Dynamic Memory, Swap, Paging display
`-H`	Host Memory, Swap, Paging display
`-I`	Guest Interrupt display
`-N`	Guest AVIO Network traffic by vswitch display
`-S`	Vswitch AVIO Network traffic by Port display

Removing and recreating a vPar or VM guest

If you remove a vPar or VM guest configuration and recreate it using the `vparcreate` or `hpvmcreate` command, the newly created vPar or VM guest might not have the same hardware paths for network and storage devices. This can change the LAN instance number. In such a case, you must update the `netconf` file with the new instance number. When the LAN instance number is incorrect, the network is inaccessible and startup scripts hang until timeout.

NOTE: You might have to mount all the file systems using the `mountall` command to access the editor that is required to modify the `netconf` file. To avoid the long boot time, boot to single user mode and modify the `netconf` file with the new LAN instance number.

Specifying VM type

Use the `-x vm_type=type` option of the `hpvmcreate` command to specify the VM type. A VM that shares CPU resources with other VM types can be the *`shared`* type. While a VM that has exclusive access to CPU resources is the *vpar*. By default, the `hpvmcreate` command creates a *shared* type guest.

NOTE: When creating a vPar using the `hpvmcreate` command, resource reservations and AutoBoot are not set by default, as is the default when using the `vparcreate` command. The following two commands are functionally equivalent:

```
vparcreate -P vparName
```

```
hpvmcreate -P vparName -B auto -x vm_type=vpar -x resources_reserved=true
```

Transformation between VM and vPar

A VM can be transformed into a vPar by setting its *vm_type* attribute to *vpar* using the `hpvmmodify` command. Also a vPar, created using the `vparcreate` or `hpvmcreate` command can be transformed into a VM by changing the *vm_type* attribute to *shared*. While making the transformation between VM and vPar, additional changes might be required for VM or vPar configuration to get the expected or default behaviour of a VM or vPar. The VM or vPar must be shutdown before making the transformation.

Resource Reservations

For a vPar created using the `vparcreate` command, the *resources_reserved* attribute is disabled by default for a VM and enabled by default for a vPar. For more information about resource reservation, "Reserved resources and resource over-commitment" (page 54). During the transformation between VM and vPar, the *resources_reserved* attribute must be changed to get the desired VM or vPar behaviour.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vm1                   9    SH   HPUX    On (OS)    2     1     1     2 GB
vpar1                11    VP   HPUX    On (OS)    1     1     1     2 GB
```

To transform a VM into a vPar,

```
# hpvmstop -P vm1 -g
```

```
# hpvmmodify -P vm1 -x vm_type=vpar -x resources_reserved=true
```

```
# hpvmstart -P vm1
```

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vm1                   9    VP   HPUX    On (OS)    2     1     1     2 GB
vpar1                11    VP   HPUX    On (OS)    1     1     1     2 GB
```

Similarly, to transform a vPar into a VM,

```
# hpvmstop -P vpar1 -g
```

```
# hpvmmodify -P vpar1 -x vm_type=shared -x resources_reserved=false
```

```
# hpvmstart -P vpar1
```

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vm1                   9    VP   HPUX    On (OS)   2      1     1     2 GB
vpar1                11    SH   HPUX    On (OS)   1      1     1     2 GB
```

CPU entitlement

When a vPar is created, its CPU entitlement (`-e`) is automatically set to (forced) 100%. Converting a VM to a vPar automatically adjusts its entitlement to 100%, while converting a vPar to a VM does not change the entitlement (it remains 100% unless modified using the `-e` option).

Memory

During type conversion, the base and floating memory values for vPar or VM guest are as follows:

Case 1: Memory parameters for shared or VM guest.

For any memory modification for shared guest, the entire memory is considered as base memory. Therefore, you cannot specify base and floating memory values. However, when an offline vPar is converted to a VM guest, the base and floating memory configuration values are retained until the values are modified.

Case 2: Memory parameters for vPar.

By default, entire memory for vPar is treated as base memory, with the following exceptions:

- Exception 1 – User explicitly specifies separate values for base and floating memory at the time of create or modify.
- Exception 2 – When an offline vPar is converted to a VM guest, and again converted back to vPar. In such a case, the base and floating memory configuration values are retained only if the values are not modified on the VM guest.

Dynamic memory

The dynamic memory parameters of a VM will not be retained when a VM is transformed to vPar and the same is transformed back to VM. The dynamic memory will be enabled by default after transforming a vPar to VM and making *resources_reserved* attribute to “false”. If you keep the value of *resources_reserved* attribute to “true” for a VM then dynamic memory will be disabled.

Guest hardware paths

After the transformation between VM and vPar, the new vPar or VM guest might not have the same hardware paths for network and storage devices. This can change the LAN instance number. To preserve the LAN instance number, Matrix OE Portable Image product must be installed and enabled on the guest OS before making the transformation.

After the transformation of guests, Matrix OE portable image product must be disabled in the guest using the following command:

```
# /opt/network/bin/hpuxpitool -d
       * Future operations will ask whether to update the backup.
       * The requested changes have been applied to the currently
         running configuration.
Tunable                       Value  Expression  Changes
gio_portable_image  (before)      1  1           Immed
                    (now)         0  0
```

Guest OS Power state management

Power state management is not supported on a VM. However, it is supported on vPar. This feature gets enabled or disabled during the first boot depending on the Guest type. After transforming VM into a vPar, the `pstatectl` command does not work. It gives the following error message:

```
# pstatectl info
pstaatectl: Could not open /dev/pwr (No such file or directory)
Missing device special file or unsupported platform
```

This issue is resolved in HP-UX 11i v3 March 2013 release and also available in PHCO_43231.

Guest OS version

VM supports HP-UX 11i v2 and 11i v3 as guest OS, where as vPar supports only HP-UX 11i v3 version. Transforming a VM, which is installed with HP-UX 11i v2 version, requires a guest OS upgrade from HP-UX 11i v2 to HP-UX 11i v3.

Mix mode support for VM and vPar environment

Starting v6.2, mixed mode support is provided in the virtualized environment. VM and vPars can be configured and run on the VSP at the same time. All the resources on the VSP are available for both VM and vPars configuration. You can use the `hpvmstatus -s` to view the resources available for VMs and vPars, and it also shows the vPar or VM types supported by the VSP.

```
# hpvmstatus -s
[HPVM Server System Resources]
vPar/VM types supported by this VSP = vPar, Shared
        ...
        Total number of operable system cores = 8
        CPU cores allocated for VSP = 1
        CPU cores allocated for vPars and VMs = 7
        ...
        ...
        Total memory allocated for vPars and VMs = 27392 Mbytes
        Memory in use by vPars and VMs = 1600 Mbytes
        Available memory for vPars and VMs = 25792 Mbytes
        Available memory for 6 (max avail.) CPU VM = 25088 Mbytes
        Available memory for 6 (max avail.) CPU vPar = 25664 Mbytes
        ...
        Maximum available vcpus for a VM = 6
        Available CPU cores for a virtual partition = 6
        Available entitlement for a 1 way virtual machine = 1330 Mhz
        Available entitlement for a 2 way virtual machine = 1330 Mhz
        ...
```

The `hpvmstatus` shows the status of both VM and vPar which are configured on the same VSP.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vm1                    9    SH   HPUX    On (OS)   1      1     1     2 GB
vpar1                 11    VP   HPUX    On (OS)   1      1     0     2 GB
```

The VM named “vm1” and a vPar named “vpar1” are running on the VSP concurrently.

On the same VSP, you can create a new VM or vPar, modify the existing VM or vPar while the other guest types are still running.

For example, stop the VM guest:

```
# hpvmstop -P vm1 -g
hpvmstop: Stop the virtual machine 'vm1'? [n/y]: y
```

Modify the VM guest to change total vcpu to 2:

```
# hpvmmodify -P vm1 -c 2
```

Start the VM guest again:

```
# hpvmstart -P vm1
....
....
```

All the previous operations on the VM are done while the vPar is still running.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vm1                    9     SH   HPUX    On (OS)   2      1     1     2 GB
vpar1                 11     VP   HPUX    On (OS)   1      1     0     2 GB
```

Specifying guest operating system type

Use the `-O os_type` option of the `hpvmcreate` or `hpvmmodify` command to specify the type of operating system that runs on the vPar or VM.

For example, to specify `hpux` as the guest operating system:

```
# hpvmcreate -P host1 -O hpux
```

For specific information about installing HP-UX guests, “Installing HP-UX vPars and Integrity VM” (page 22).

If you do not specify the operating system type, it defaults to `UNKNOWN`. When you install the operating system and boot the guest, this guest configuration parameter is automatically set to the appropriate operating system type.

When a running guest transitions from running in the machine console to running in the operating system, the operating system type is detected. If the operating system type is different from the information in the configuration file of the guest, it is automatically updated to reflect the current operating system.

Creating VM labels

The `-l` option of the `hpvmcreate` or `hpvmmodify` command specifies the label of the VM. The VM label is a descriptive label unique to a VM or vPar. The label can be useful in identifying a specific VM in the output displayed by the `hpvmstatus -V` command. The label can contain up to 255 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). If white space is desired, the label must be quoted ("").

For example, to create a VM with a label “Virtual Machine number one”, run the following command:

```
# hpvmcreate -P vm001 -l "Virtual Machine number One"
# hpvmstatus -P vm001 -V
[Virtual Machines Details]
Virtual Machine Name  : vm001
Virtual Machine UUID  : 24a7bfa4-b1b2-11e2-8400-b499ba6430e0
Virtual Machine ID    : 1
Virtual Machine Label : Virtual Machine number One
..
..
```

Specifying the VM boot attribute

The `-B` option of the `hpvmcreate` or `hpvmmodify` command specifies the startup behavior of the VM. The `start_attr` attribute can have the following (case-insensitive) values:

- `auto`: Automatically start the VM when Integrity VM is initialized on the host.
- `manual`: Manually start the VM.

For example, to create a VM with `start_attr` attribute as manual, run the following:

```
# hpvmcreate -P vm001 -B manual
```

Alternatively, you can modify the `start_attr` attribute of an existing VM or vPar using the `hpvmmodify` command:

```
# hpvmmodify -P vm001 -B auto
```

If the `start_attr` attribute is set to `auto`, the VM is started when Integrity VM is initialized. This is the default. This occurs when the VSP system is booted, and when the Integrity VM software is stopped and restarted on a running VSP. For example, when you upgrade Integrity VM to a new version on a running system, the software is started automatically. The VSP attempts to start all VMs for which the attribute is set to `auto`. If the resources are insufficient, some VMs might fail to start.

If the attribute is set to `manual`, the VM is not started automatically when Integrity VM is initialized on the VSP. You can start the VM manually with the `hpvmstart` command or through its virtual console.

The `-B` option does not set the console of the VM to enable booting when the VM is started. This function must be set with the console of the VM.

In addition to automatically starting guests when Integrity VM starts, this feature also determines a startup order to best utilize VSP processor and memory resources. On cellular systems with CLM configured, the goal is to start the guests so that CLM is utilized first. For each guest with the `start_attr` attribute set to `auto`, the startup order is based on `resources_reserved` attribute and a memory weight and a processor weight added together. A guest with `resources_reserved` attribute set to `true`, gets the highest priority while deciding the boot order.

A rough estimate of the memory weight calculation is:

100 * guest memory size / available host memory + 2 (if the guest resources can fit into available CLM of the cell and processors)

A rough estimate of the processor weight calculation is:

(minimum guest cpu entitlement * number of virtual processors) / (100 * number of host processors)

Guests are expected to start in order of highest weight to lowest. You can adjust the order by setting the `sched_preference` attribute. If a guest fails to start for any reason, the sequence continues with the next guest. For memory placement on a non cell-based system or cell-based system with all ILM configured, the boot order has little affect.

In general, on these configurations, the largest guests boot first. On cell-based systems with CLM configured, expected memory placement depends on the calculated weights, the `sched_preference` setting, and the VSP memory configuration:

- If `sched_preference` is not set, or set to “cell” and the guest resources fit into one cell, CLM is used.
- If there is not enough CLM and there is enough ILM, ILM is used.
- If `sched_preference` is set to “ilm” and there is enough ILM, ILM is used.
- If there is not enough ILM, the memory is allocated from all cells (striped).
- If there is insufficient ILM but the guest resources fit into one cell, CLM is used. Otherwise, the memory is striped.

Creating guest administrators and operators

vPars and Integrity VM provides secure access to guest machine consoles. When you create a VM, you can specify groups and user accounts to have administration or operator privileges on that guest. These users are allowed to log in to the VSP using their own user accounts, and to use the `hpvmconsole` command to perform system administration tasks on the guest VM.

A captive virtual console account is a special-purpose user account created on the VSP for each guest administrator or operator. These types of user accounts use the `/opt/hpvm/bin/hpvmconsole` directory for a shell, and the desired per-guest directory of the guest for a home directory. For virtual console access, the account also requires a password, and access to its associated guest.

Before you create a VM, use the `useradd` command to create user accounts for virtual console access. For example, the following command adds the user account `testme1`:

```
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
 -c "Console access to guest 'testme'" \
 -d /var/opt/hpvm/guests/testme \
 testme1
```

Do not use the hpvmsys group for user accounts. This group is used for security isolation between components of Integrity VM.

These types of console users are specified as either admin (guest administrators) or oper (guest operators). Guest operators can access the VM console, shut down and reboot the guest, view system status, transfer control to another guest operator or administrator, and set system identification. The guest administrator has all these capabilities and the ability to use the virtual console say commands (restricted to use by Hewlett Packard Enterprise field support specialists).

To specify guest administrators and operators, use the hpvmcreate, hpvmmodify, hpvmmigrate, and hpvmclone commands. To assign administrator and operator privileges to a user group, include the -g option. To assign administrator and operator privileges to a specific user, use the -u option.

NOTE: Console users cannot use the su command to change from one privilege level to another. Per-user checks are based on login account identifiers, not on UUIDs.

The following command creates the VM named testme with the administrator named testme1:

```
# hpvmcreate -P testme -u testme1:admin
```

Guest operators and administrators need access to the hpvmconsole command to control the VM. If you do not want the same users to have access to the VSP, you can restrict use of the hpvmconsole command to only guest console access by creating a restricted account for that purpose. To do so:

1. Use the useradd command and set up an /etc/passwd entry for each guest on the VSP. The user name of the account must be the same as the guest name, and must have no more than eight characters. For example:

   ```
   # useradd -d /var/opt/hpvm/guests/host1 \
   -c 'host1 console' -s /opt/hpvm/bin/hpvmconsole host1
   ```

   This example uses the following options:

   - The -d option specifies the home directory for the host1 account.
   - The -c option specifies a comment text string that describes the account.
   - The -s option specifies the path for the shell of the new account.

2. Use the passwd command to set a password for the account. For example:

   ```
   # passwd host1
   ```

3. Use the hpvmmodify command to provide the user with guest administration privileges:

   ```
   # hpvmmodify -P winguest1 -u host1:admin
   ```

A guest administrator can now access the host1 virtual console by using the ssh command or telnet command on the VSP and logging in to the host1 account. The guest administrator cannot use the su command.

NOTE: For security reasons, Hewlett Packard Enterprise strongly recommends that you do not include /opt/hpvm/bin/hpvmconsole, the virtual console image, in /etc/shells. Doing so opens two security vulnerabilities:

- It allows ftp access to the account.
- It allows a general user to select the image with the chsh command.

The following is an example session of remote access to the host1 virtual console on the VSP myhost:

```
# telnet host1

Trying .xx.yy.zz...
Connected to host1.rose.com.
```

```
Escape character is '^]'.

HP-UX host B.11.31 U ia64 (ta)

login: guest1
Password:
Please wait...checking for disk quotas


   MP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[host1] vMP>
```

The virtual console interface displays raw characters for the `CL` and `CO` commands, including the attempts of the guest to query the console terminal for its type and characteristics. As a result, the terminal answers those queries, which can cause the terminal setup communication to interfere with the virtual console commands. Interactive users can clear the screen. However, this situation can be a problem for noninteractive or scripted use of the console.

Administrator account names

The virtual console administrator name can be any valid HP-UX login name. To continue accessing the virtual console, existing guest console accounts must be added to the authorization list for the associated guest using the `usermod` command. This allows multiple accounts to map to the guest and requires the account names to be valid HP-UX login strings.

The guest configuration file (set using the `-u` and `-g` options to the `hpvmcreate`, `hpvmmodify`, and `hpvmclone` commands) determines the authorization of access to the virtual console. This controlled access allows you to temporarily block access by using the `hpvmmodify` command to change the virtual console administrator account name.

vPars or VM user accounts

The configuration for captive `hpvmconsole` guest user accounts supports additional access controls and configurations. This change requires that the guest user accounts have the correct home directory. To ensure that the user continues to have administrative console access, use the following command:

```
# hpvmmodify -P compass1 -u compass1:admin
```

Using the virtual console

Each vPar or VM guest has its own virtual console from which you can power on or off the vPar or VM guest, boot the guest operating system or shut it down, and so on. The `hpvmconsole` command connects to the virtual console of a specified vPar or VM guest.

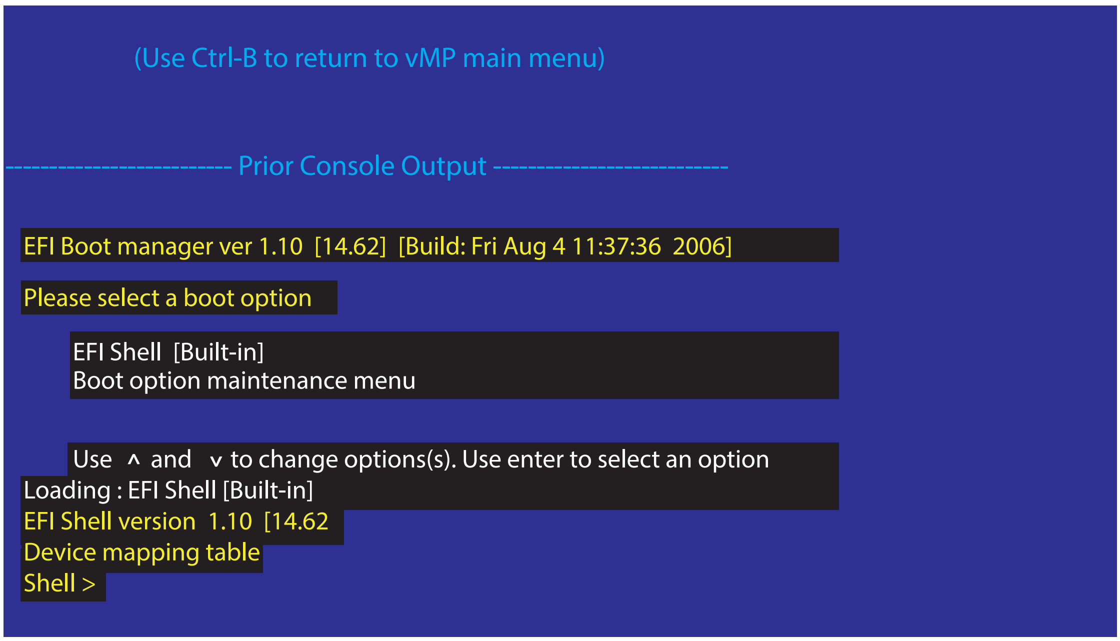
To start the virtual console for the guest named `host1`, enter the following command:

```
# hpvmconsole -P host1


   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
```

```
          SL: Show Event Logs
          VM: Virtual Machine Menu
          HE: Main Help Menu
           X: Exit Connection

[host1] vMP>
```

To return to the virtual console when the display is in the EFI, press **Ctr+B**. Use the `co` command to open the virtual console. For example:

```
[host1] vMP> co
```



You can pass a command to the vPar or VM guest console using the `-c` option with the `hpvmconsole` command. For example, to start a VM named `host1`, enter the following command:

```
# hpvmconsole -P host1 -c "pc -on"
```

Table 34 (page 250) lists the options that can be used with the `hpvmconsole` command.

Table 34 Options to the `hpvmconsole` command

Option	Description
`-P` *vm-name*	Specifies the name of the VM console.
`-p` *vm-number*	Specifies the number of the VM console.
`-c` *command*	Specifies a machine console command to run on the VM.
`-e` *echar*	Specifies an alternate interrupt character. The default interrupt character is **Ctrl+B**, unless the session is on the `/dev/console` of the VSP, in which case, use the **Ctrl+X**.
`-f`	Follows the console output after reaching EOF on standard input. Used for scripting.
`-i`	Interacts with the console. Used for scripting.
`-q`	Makes scripted operations less verbose.

To get information about using the virtual console, enter the `HE` command. For example:

```
[host1] vMP> he
==== vMP Help: Main Menu ================================ (Admin) ============

HPVM B.06.30  ccipf opt Thu Dec 01 2011
(C) Copyright 2000 - 2011 Hewlett-Packard Development Company, L.P.
```

```
        Virtual Management Processor (vMP) Help System

    Enter a command at the help prompt:
     OVerview  - Launch the help overview
     LIst      - Show the list of vMP commands
     <COMMAND> - Enter the command name for help on an individual command
     TOPics    - Show all vMP Help topics and commands
     HElp      - Display this screen
     Q         - Quit help
```

For more information about using the `hpvmconsole` command, see *hpvmconsole*(1M).

Using the virtual iLO Remote Console

The vPars and Integrity VM virtual iLO Remote Console allows you access to the guest console by logging into a specific IP address. You can assign each guest a virtual iLO Remote Console IP address with which the end user can connect using either telnet or SSH. After login authentication, the guest console is immediately available. The user is no longer required to know the VSP machine IP address or guest name. Instead, the user must know only the virtual iLO Remote Console IP Address. The virtual iLO Remote Console IP stays the same even after an OVMM. There is also no need to manually run any command, such as the `hpvmconsole` command.

The following sections describe:

- Configuring a virtual iLO Remote Console
- Choosing the virtual iLO Remote Console IP address
- Deleting a virtual iLO Remote Console
- Getting the virtual iLO Remote Console settings of a guest

Configuring, deleting, and obtaining status of a virtual iLO Remote Console

You can assign a virtual iLO Remote Console IP address when you create, modify, or clone a guest, using the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` commands:

- `hpvmcreate -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*
- `hpvmmodify -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*
- `hpvmclone -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*

For example:

```
# hpvmmodify -P guestname -K 16.92.81.68 -L 255.255.252.0
```

NOTE: Only IPv4 addresses are supported, not IPv6.

The virtual iLO Remote Console IP address must be unique and different from both the Host IP address and the Guest IP address. The virtual iLO Remote Console IP address must not be configured in advance. When the virtual iLO Remote Console is created, Integrity VM automatically creates an alias interface for the IP address. For example, if you create the virtual iLO Remote Console:

```
# hpvmmodify -P guestname -K 16.92.81.68 -L 255.255.252.0
```

Integrity VM configures the IP alias in a similar manner as if you specified the `ifconfig` command:

```
"ifconfig lan0:274485572 16.92.81.68 netmask 255.255.252.0"
```

To view the alias interface that Integrity VM creates, run the `netstat` command:

```
# netstat -rn
Routing tables
Destination            Gateway           Flags    Refs   Interface          Pmtu
127.0.0.1              127.0.0.1         UH       0      lo0                32808
16.92.81.68            16.92.81.68       UH       0      lan1:274485572     32808
16.92.80.101           16.92.80.101      UH       0      lan1               32808
127.0.0.0              127.0.0.1         U        0      lo0                32808
default                16.92.80.101      U        0      lan1                1500
```

To delete a virtual iLO Remote Console, specify 0 as the IP address. For example:

```
# hpvmmodify -P guestname -K 0
```

To obtain the virtual iLO Remote Console settings of a guest, use the `hpvmstatus` command. For example:

```
# hpvmstatus -P guestname
....
[Remote Console]
Remote Console Ip Address:      16.92.81.68
Remote Console Net Mask:        255.255.252.0
```

When users connect to the virtual iLO Remote Console IP address, they must log in using the standard `telnet` or `ssh` system authentication. After authenticating, the users receive immediate access to the guest console:

```
# ssh -l guest1admin 16.92.81.68
Password:
   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[guest1] vMP>
```

The username used to access and log into the virtual iLO Remote Console must have guest administrator or operator privileges. The following example creates a guest administrator name **guest1admin** for the guest **guest1**. The `hpvmmodify -u` option is used to grant the guest administrator privilege:

```
# useradd -d /var/opt/hpvm/guests/guest1 -c 'guest1 console' guest1admin
# passwd guest1admin
# hpvmmodify -P guest1 -u guest1admin:admin
# hpvmmodify -P pqsvm53  -K  xxx.xxx.xxx.xxx  -L  xxx.xxx.xxx.xxx
# telnet xxx.xxx.xxx.xxx
```

For more information about guest administrators and operators, see "Creating guest administrators and operators" (page 247).

When a guest is migrated from one to another VSP using OVMM, the Integrity VM virtual iLO Remote Console is also migrated to the new VSP. Before migration, the virtual iLO Remote Console process is running on only the source VSP. After migration, the virtual iLO Remote Console process is stopped on the source VSP. Any client that was connected to that virtual iLO Remote Console is disconnected. A new virtual iLO Remote Console process is started on the target VSP. New client connections to the virtual iLO Remote Console IP address are now sent to the virtual iLO Remote Console process on the new VSP.

Integrity VM virtual iLO Remote Console limitations

The following are the virtual iLO Remote Console features:

- By default, only SSH is supported.

  To add telnet support for virtual iLO Remote Console, you must install two additional HP-UX enhancement patches, one for telnetd and one for the login (`/usr/bin/login`) command. If you try to telnet to the virtual iLO Remote Console without these patches, an error message is sent to the telnet client, and the connection is closed.

  Install the following patches on the VSP:

  - PHCO_41595
  - PHNE_41452

- The SSH server host keys of the virtual iLO Remote Console can change.

  When an SSH client connects to an SSH server, the client downloads the host keys of the server and retains a local copy (usually in a file such as `~/.ssh/known_hosts`). On subsequent connections, the SSH client verifies that the host key sent by the server matches the local copy. If the keys do not match, the SSH client prints an error message.

  The virtual iLO Remote Console uses the SSH server host keys of the host system. If the guest is migrated to another host system (using OVMM), these host keys change. When an end user does an SSH connection, an error message is displayed. The end user must manually delete the local copy of the host key. For additional information, *ssh*(1).

- Guest Administrator accounts are not migrated during OVMM.

  Guest administrator accounts on the source VSP system are not automatically migrated to the target VSP system during OVMM. You must manually add guest administrator accounts to the target VSP system, using the same `useradd` commands that are used on the source system. For information about creating Guest Administrator and Operator accounts, "Creating guest administrators and operators" (page 247).

- The virtual iLO Remote Console does not support rlogin connections.

Guest configuration files

When the guest is created, the VSP creates the guest configuration file `/var/opt/hpvm/guests/`*`guestname`*.

Integrity VM creates up to three guest configuration files:

- The `vmm_config.current` file contains the guest configuration that is currently set.
- The `vmm_config.prev` file contains the last known guest configuration settings.
- The `vmm_config.next` file contains the configuration settings that have changed since the guest was started. To initiate these changes, you must reboot the guest.

CAUTION: Never modify the guest configuration files manually. Always use the appropriate Integrity VM command (`hpvmmodify` or `hpvmdevmgmt`) to modify guest configuration parameters. Directly modifying the guest configuration files can cause guests to fail in unexpected ways.

Managing dynamic memory from the VSP

On the VSP, the dynamic memory feature is included with Integrity VM. You can manage dynamic memory on the VSP using the `-x` option with the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command. The `-x` option associates a variety of configuration parameters with the guest, including dynamic memory and network management for the guests. Table 35 (page 254) lists the `-x` keywords used for dynamic memory.

Table 35 Dynamic memory control command options

Keyword value pair	Description
dynamic_memory_control={1|0}	Specifies whether a privileged user on the guest (such as root) can change the dynamic memory values while the guest is running. To disable guest-side dynamic memory control, specify 0 (zero). If the guest is not active, the only effect is the modification of the guest configuration file. On the running guest, the change takes effect immediately.
ram_dyn_type={none|any|driver}	Specifies the type of dynamic memory control for the guest. When this configuration parameter is set to none, dynamic memory is disabled. If the guest is running with dynamic memory enabled, and you set this value to none, the guest configuration file is modified to remove all dynamic memory ranges and control information. When this configuration parameter is set to any, the next boot of the guest determines whether or not dynamic memory is enabled on the guest. If the dynamic memory driver is loaded, the value of this parameter is changed to driver. If none of the drivers are loaded or found, the value is not changed. When this configuration parameter is set to driver, guest dynamic memory controls and ranges are functional. Depending on the current or default settings, messages might be displayed indicating a resetting of the dynamic memory range values to match the current memory range settings. If you change the available guest memory value (using the -r option), the dynamic memory values are validated for range, and modified.
ram_dyn_min=*amount*	Specifies the minimum amount of memory that can be dynamically allocated to the guest. The ram_dyn_min value must be greater than the minimum memory (displayed by the hpvmstatus command) and less than the ram_dyn_max value.
ram_dyn_max=*amount*	Specifies the maximum amount of memory that can be dynamically allocated to the guest. The value of ram_dyn_max must be greater than the value of ram_dyn_min.
ram_dyn_target_start=*amount*	Specifies the amount of memory that the dynamic memory driver attempts to access when the guest starts. The value of the ram_dyn_target_start must be greater than the ram_dyn_min parameter and less than or equal to the ram_dyn_max parameter. When the guest starts, it initially has access to the guest memory size (specified by the -r option); later, the dynamic memory driver reduces the memory to the value of the ram_dyn_target_start parameter. The ram_dyn_entitlement and amr_enable options must be set to enable adjustments.
ram_dyn_entitlement=*amount*	Specifies the minimum guaranteed amount of memory.
amr_enable={0|1}	Specifies whether adjustments can be made.
amr_chunk_size=*amount*	Specifies the increment amount for changes in memory size (default is 256 MB). Larger values result in faster memory size growth.
ram_target={0|start|*amount*}	Sets the current memory size for the guest. The ram_target keyword is valid on the hpvmmodify and hpvmmgmt commands only. When you specify 0 (zero), the dynamic memory driver reduces the memory on the guest to a comfortable minimum without forcing guest memory to be paged out. This minimum value changes over time as the operating needs of the guest changes. When you specify start, the guest dynamic memory size grows to the allocated value specified using the -r option. This parameter is dynamic and can be used only on an active guest.

Configuring a VM to use dynamic memory

By default, dynamic memory is enabled. To configure a VM to use dynamic memory, use the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command. With the command, include the following `-x` option to set initial values:

```
-x ram_dyn_type = any | driver
-x ram_dyn_min = minimum size for memory size changes
-x ram_dyn_max = maximum size for memory size changes
```

You can configure a VM to reduce its memory size early in a boot process, making the VM available but maintaining lower memory overhead on the VSP system. Use the following `-x` option to enable this feature:

```
-x ram_dyn_target_start = memory size after boot
```

CAUTION: You must not set `ram_dyn_target_start` to a low value. If you set this value to a lower value it results in a huge memory pressure on the guest during boot process which leads to guest crash or a hung state. For more information about memory, see "Specify sufficient VM memory" (page 259).

You can supply several dynamic memory keywords on the same command line. For example, to enable dynamic memory and to configure the guest named `host1` to reduce its size early in the boot process, enter the following command:

```
# hpvmmodify -P host1 -r 6G \
-x ram_dyn_type=any \
-x ram_dyn_min=1222M \
-x ram_dyn_max=6G \
-x ram_dyn_target_start=2G
```

This command specifies the following values:

- The VM memory size is set to 6 GB.
- Dynamic memory is enabled using any dynamic memory support available.
- The minimum amount of memory that the VM can have is 1222 MB.
- The maximum amount of memory that the VM can have is 6 GB.
- The memory size to reduce to after it boots is 2 GB.

If the VM is running when the dynamic memory feature is configured for the first time, the VM must be rebooted for the configuration changes to take effect.

Viewing dynamic memory on the VSP

You can view the dynamic memory parameters and status for each guest by using the standard Integrity VM commands. For example, for the guest named `host1`, the `hpvmstatus` command displays the following information about dynamic memory:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
  Type                     :  driver
  Minimum memory           :  1222 MB
  Target memory            :  2103 MB
  Memory entitlement       :  Not specified
  Maximum memory           :  6144 MB
  Current memory           :  2103 MB
  Comfortable minimum      :    27 MB
  Boot memory              :  6135 MB
  Free memory              :   125 MB
  Available memory         :   286 MB
```

```
  Memory pressure         :     0
  Memory chunksize        : 65536 KB
  Driver Mode(s)          : STARTED ENABLED
  AMR state               : DISABLED
.
.
.
```

Table 36 (page 256) lists the dynamic memory characteristics displayed by the `hpvmstatus` and `hpvmmgmt` commands.

Table 36 Dynamic memory characteristics

Characteristic	Setting	Description
Type	`none`	No dynamic memory support.
	`any`	Dynamic memory is configured on the host, but the dynamic memory subsystem on the guest has not started and reported the implementation type.
	`driver`	Dynamic memory is implemented in a driver and does not use Guest OS Online Add or Delete features.
	`OLAD`	Dynamic memory is implemented using Guest OS Online Add or Delete features.
Minimum memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	The lower bounds for `ram_target` and `ram_dyn_target_start`.
Target memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	The target memory size of the guest, set using `ram_target` or `ram_dyn_target_start`.
Maximum memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	The upper bounds for `ram_target` and `ram_dyn_target_start`.
Current memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	The current memory size of the guest (usually equal to target memory).
Comfortable minimum	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	A value for `ram_target` which can be used to reduce the guest memory but allow the guest sufficient memory resources to continue running a minimal workload.
Boot memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	Size of physical memory in the VM presented to the guest OS.
Free memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	Amount of free memory in the guest.
Available memory	*`value`*`M` (for megabytes) or *`value`*`G` (for gigabytes)	Amount of memory in the guest allocated by user processes but not locked. This memory is available for paging.
Memory pressure	*`value`*	A value between 0 and 100 used as an indicator of memory deficit and paging. The higher the number, the longer the

Table 36 Dynamic memory characteristics *(continued)*

Characteristic	Setting	Description
		system is in a memory deficit. A memory pressure value approaching 100 usually means the system is hung.
Memory chunksize	*value*	The allocation chunk size used by dynamic memory when increasing and decreasing guest memory (as described in "Specify sufficient VM memory" (page 259).
Driver mode(s)	`started`	Dynamic memory can change guest memory size.
	`enabled`	Control that overrides `started`.
	`guestctl`	Guest-side control is enabled.

The following example shows active usage of the VSP and guests dynamic memory usage values, along with the guest memory utilization. The current swapping and paging, and translation address memory misses per second of the guests are included. For a description of each column displayed, the *hpvmsar* manpage. The dash (-) in the example indicates the guest named `ux2` is not currently booted.

```
# hpvmsar -G -A
HP-UX witch4 B.11.31 U ia64    10/22/10

10:02:28   GUEST  GTOTMEM(MB) HDYNRCLM(MB) GCURMEM(MB) GCURFREE(MB)  GSWAP  GPAGE GADDRTMISS/s
10:02:30   ux1           8186            0        8186         5956      0      0            0
           ux2              -            -           -            -      -      -            -
10:02:31   ux1           8186            0        8186         5956      0      0            0
           ux2              -            -           -            -      -      -            -
10:02:32   ux1           8186            0        8186         5956      0      0            0
           ux2              -            -           -            -      -      -            -
```

Modifying a memory size of the VM on the VSP

After dynamic memory is configured, you can change a memory size of the VM to any value between the minimum size (**`ram_dyn_min`**) and the maximum size (**`ram_dyn_max`**) in increments of the chunk size (64 MB). Use the `-x` option with the `hpvmmodify` command to change the memory size:

```
# hpvmmodify -P host1 -x ram_target = new memory size
```

Managing dynamic memory from the guest

Dynamic memory management from the guest is disabled by default and must be enabled from the VSP. If the feature is not enabled, you can view dynamic memory information, but cannot change the memory size.

Use the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command and include the `-x dynamic_memory_control` option. Specify `1` as the argument to the option. For example, on the VSP system, enter the following command to enable dynamic memory control on the guest named `host1`:

```
# hpvmmodify -P host1 -x dynamic_memory_control=1
```

Viewing dynamic memory information from the guest

Use the `hpvmmgmt` command on the HP-UX guest to manage and view the dynamic memory information. This command is installed when you install the VirtualBase software, as described in "Installing VirtualBase on a vPar or VM Guest" (page 28).

Table 37 (page 258) lists the options that can be used with the `hpvmmgmt` command.

Table 37 Options to the `hpvmmgmt` command

`-l` *`type`*	Specifies the type of data for which you want to view more information. For *`type`*, enter `ram`.
`-l` *`type`* `-t` *`interval`*	Allows you to continually watch and check the dynamic ram values. For the *`interval`*, specify the number of seconds between fetches of live data.
`-t` *`interval`*	Allows the `hpvmmgmt` command to continuously refetch the requested type of data using the value specified for the *`interval`* parameter.
`-c` *`num`*	Specifies the number of virtual CPUs to be enabled on the guest.
`-v`	Displays the version number of the `hpvmmgmt` command.
`-V`	Displays detailed information (verbose mode) about the VMs.
`-M`	Displays verbose attribute and resource information in a machine-readable format.
`-X`	Displays verbose attribute and resource information in the XML format.
`-x ram_target={0 | start | `*`amount`*`}`	Specifies the guest RAM target, where: • 0 indicates the guest dynamic memory is reduced to a comfortable minimum value. • `start` indicates the guest dynamic memory is set back to the boot time value. • *`amount`* is a specific target memory size for the guest.

For example, on the guest, use the `hpvmmgmt` command to view the dynamic memory information. Enter the following command:

```
# hpvmmgmt -l ram

  [Dynamic Memory Information]
  =======================================
  Type                     : driver
  Current memory           : 6135 MB
  Target memory            : 6135 MB
  Comfortable minimum      :   27 MB
```

To view more information, include the `-V` option. For example:

```
# hpvmmgmt -V -l ram
  [Dynamic Memory Information]
  =======================================
  Type                    :  driver
  Current memory          :  2103 MB
  Target memory           :  2103 MB
  Comfortable minimum     :  2423 MB
  Minimum memory          :  1222 MB
  Maximum memory          :  6144 MB
  Boot memory             :  6135 MB
  Free memory             :   124 MB
  Available memory        :   286 MB
  Memory pressure         :    12
  Memory chunksize        : 65536 KB
  Driver Mode(s)          : STARTED ENABLED GUESTCTL
```

Modifying memory size of VM from the guest

After the dynamic memory feature is configured and enabled, you can modify a memory size of the VM to any value between the minimum size (**`ram_dyn_min`**) and the maximum size (**ram_dyn_max**) in increments of the chunk size (64 MB). Use the `-x` option with the `hpvmmgmt` command:

```
# hpvmmgmt -x ram_target=memory_size
```

For example, to change the guest memory size to 4 GB, enter the following command:

```
# hpvmmgmt -x ram_target=4096M
Attempting to increase memory from 2103 MB to 4096 MB.
Successfully began to change ram_target to 4096 MB.
```

Troubleshooting dynamic memory problems

This section describes how to solve problems in the use of dynamic memory.

Dynamic memory restrictions

Use of dynamic memory is subject to the following restrictions:

- The memory size of a VM cannot be increased to a value above its original boot size (as specified with the `-r` option).
- If the VM memory has become fragmented, attempting to reduce the size of the VM might fail or might take a very long time. If you cannot reduce the size of the VM to the desired size, abort the operation by setting a new target size.
- Increasing the size of a VM requires free memory on the VSP. If the VSP memory is insufficient, the operation might take a very long time to complete, and might fail.
- If the values of `ram_target` and `ram_dyn_target_start` are not within the values of `ram_dyn_min` and `ram_dyn_max`, a warning message is displayed.

VM resource considerations

During normal operation of a system that has a workload running on it, the large pages might become fragmented over time. This is true on the VSP and on a VM running the HP-UX operating system. If the memory of the VM is fragmented, the dynamic memory subsystem is unable to reduce the size of guest. This is due to the minimum chunk size used for the reduction. If dynamic memory cannot remove at least 64 MB of physically contiguous guest memory, the size is not reduced.

Specify sufficient VM memory

If you set the value of `ram_dyn_target_start` small, the guest operating system of the VM might hang or crash while booting. In this case, the VM does not have access to sufficient amount of memory. As a rule, do not decrease the memory allocated to an HP-UX guest by more than 75% of its allocated memory size. Do not reduce the memory of a VM configured with 2 GB of memory by more than 50%.

If the VM crashes while booting on the VSP, use the `hpvmmodify` command to increase the value of the `ram_dyn_target_start` parameter. For example, to increase the memory size for the VM named `host1`, enter the following command on the VSP:

```
# hpvmmodify -P host1 -x ram_dyn_target_start=2GB
```

After you set this parameter, reboot the VM.

If the VM hangs, on the VSP, use the `hpvmstatus` command to verify the memory statistics on the VM. For example:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
  Type                    :  driver
  Minimum memory          :  1222 MB
```

```
  Target memory          :  2103 MB
  Maximum memory         :  6144 MB
  Current memory         :  2103 MB
  Comfortable minimum    :    27 MB
  Boot memory            :  6135 MB
  Free memory            :     0 MB
  Available memory       :   286 MB
  Memory pressure        :   100
  Memory chunksize       : 65536 KB
  Driver Mode(s)         : STARTED ENABLED
.
.
.
```

An indication of this problem is a small or zero amount of free memory and a large memory pressure value (100). If these indicators are present, use the `hpvmmodify` command on the VSP to increase the memory size of the VM. The VM then boots normally.

Actual memory allocations

If you specify a value for the `ram_target` or `ram_dyn_target_start` parameter that results in a change in memory size that is not a multiple of 64 MB, the target value is reset.

For example, if you specify 6 GB of memory, the HP-UX guest actually has access to 6135 MB of memory. If you attempt to set the memory size to 2048 MB, the amount of memory actually removed is 4087 MB. This is not a multiple of 64 MB, so the target memory size is reset to 2103 MB.

Enabling dynamic memory on the VM and on the VSP

The VirtualBase software must be installed on the VM before you can use dynamic memory parameters on the VSP system. For example, if the VirtualBase software is not installed, the `hpvmstatus` command displays the following:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
    NOTE: Dynamic data unavailable, configured values only
Type                    : driver
Minimum memory          : 1024 MB
Target memory           : 2048 MB
Maximum memory          : 3072 MB
.
.
.
```

If you attempt to modify the dynamic memory of the VM from the VSP, the following errors are displayed:

```
# hpvmmodify -x ram_target=2048M -P host1

hpvmmodify: ERROR (host1): Query to dynamic memory driver failed: Function is not available.
hpvmmodify: Failed to set ram_target.
hpvmmodify: Unable to modify the guest.
```

If you attempt to modify the dynamic memory from the VM, the following errors occur:

```
# hpvmmgmt -V -l ram
Dynamic memory driver not found on guest.
hpvmmgmt: Unable to continue.
# hpvmmgmt -x ram_target=2048
Failed to open dynamic memory driver, error: No such device.
Failed to set dynamic value error: No such device
hpvmmgmt: Unable to continue.
```

For information about installing the VirtualBase software, Section (page 28).

Upgrading the VirtualBase software when upgrading Integrity VM

The dynamic memory software has two components— the VSP support and the HP-UX guest support. These two components must be at the same version level for dynamic memory to function. When you upgrade Integrity VM, you must also install the new VirtualBase kit on the guest. (You must also upgrade the guest operating system if it is no longer supported.) During this upgrade process, dynamic memory might not function.

If there is a version mismatch, a message is written to the `syslog` file (`/var/adm/syslog/syslog.log`) of the VSP when the guest starts. For example:

```
vmunix: (hpvmdvr) Dynamic memory version mismatch Guest 5.
Please update the guest kit
```

This example indicates that the VirtualBase software kit on VM number 5 is out of date. To determine which guest is number 5, use the `hpvmstatus` command. In the following example, guest 5 is named `dale`:

```
# hpvmstatus

Virtual Machine Name VM #  Type    OS Type   State   #VCPUs #Devs  #Nets   Memory
==================== ===== ======= ========= ======  =====  =====  ======= ========
chip                   1    SH       HPUX    On (OS)    1      1      1      2 GB
dale                   5    SH       HPUX    On (OS)    1      0      0      2 GB
```

For information about installing the VirtualBase software, “Installing VirtualBase on a vPar or VM Guest” (page 28).

Automatic memory reallocation

Automatic memory reallocation is an optional feature of Integrity VM that allows automated changes in the amount of physical memory in use by VMs based on memory load conditions. Automatic memory reallocation is available only on guests that support dynamic memory.

To use automatic memory reallocation, the VM must have the VirtualBase software installed, because this is required for dynamic memory. For vPar or VM guest VirtualBase software installation instructions, “Installing VirtualBase on a vPar or VM Guest” (page 28).

Enabling automatic memory reallocation on the VSP

On the VSP, the automatic memory reallocation software is included with Integrity VM. The automatic memory reallocation daemon (`hpvmamrd`) is enabled by default. To disable automatic memory reallocation, the following line must be included in the `/etc/rc.config.d/hpvmconf file`: `HPVMAMRENABLE=0`. When `HPVMAMRENABLE=0` is not set in `hpvmconf`, `hpvmamrd` is automatically started and stopped when Integrity VM is started and stopped.

When running, `hpvmamrd` monitors the state of VMs that are enabled for automatic memory reallocation. Every 10 seconds, `hpvmamrd` examines the state of relevant VMs, and takes action within the parameters. It also takes action when an attempt is made to boot a VM that requires more physical memory than is currently available.

Enabling automatic memory reallocation on a VM

By default, VMs are not enabled for automatic memory reallocation. Only VMs that support dynamic memory can use automatic memory reallocation. Use the following `-x` options to enable automatic memory reallocation on a VM:

`-x amr_enable`

`-x ram_dyn_entitlement=`*`minimum memory size in MB`*

This option is supported on running VMs. If this is executed for a VM that does not support dynamic memory an error does not occur, but the command is ignored. A VM that does not have a value for `ram_dyn_entitlement` is also ignored by automatic memory reallocation. A VM that is enabled for automatic memory reallocation does not support manual dynamic memory operations from the VM. It does not support manual dynamic memory operations from the VSP that would cause the VM to shrink below its entitlement.

Viewing automatic memory reallocation

You can view automatic memory reallocation parameters and status for each VM by using the standard Integrity VM commands. The `hpvmstatus` command displays the following information about automatic memory reallocation:

```
# hpvmstatus -r
[Virtual Machine Resource Entitlement]
[Virtual CPU entitlement]
                                                       Percent    Cumulative
Virtual Machine Name VM #  #VCPUs Entitlement Maximum  Usage      Usage
==================== ===== ====== =========== =======  =======   ================
guest0                1      2       10.0%     100.0%    2.0%       237
guest1                2      2       10.0%     100.0%    2.5%     28863


Virtual Machine        DynMem  Memory  DynMem DynMem DynMem  Comfort Total  Free   Avail  Mem   AMR   AMR
Name1           VM #   Min     Entitle Max    Target Current Min     Memory Memory Memory Press Chunk State
=============== ====   ======  ======= ====== ====== ======= ======= ====== ====== ====== ===== ===== =======
guest0          1      512MB   2GB     5GB    5114MB 5114MB  1722MB  5GB    3534MB  324MB 0        0B DISABLED
guest1          2        1GB   2GB     4GB    2106MB 2106MB  1594MB  4GB     801MB  282MB 0      400B ENABLED
```

Online Memory Migration for vPar

Online Memory Migration is supported on HP-UX 11i v3 vPar starting HP-UX vPars and Integrity VM v6.2. This means that memory can be added and deleted from a live vPar without reboot.

Command options for base or floating memory configuration

Table 38 (page 262) lists the new and modified options for the `vpar` and `hpvm` commands.

Table 38 Options to `vpar` and `hpvm` commands

Command	Option	Description
`hpvmcreate`	`-a mem::`*`<amount>`* `-x` *`vm_type=vpar`*	Amount of memory added as base memory for the new vPar.
	`-a mem::`*`<amount>`*`:b -x` *`vm_type=vpar`*	Amount of memory added as base memory for the new vPar.
	`-a mem::`*`<amount>`*`:f -x` *`vm_type=vpar`*	Amount of memory added as floating memory for the new vPar.
`vparcreate`	`-a mem::`*`<amount>`*`-a mem::`*`<amount>`*`:b`	Amount of memory added as base memory.
	`-a mem::`*`<amount>`*`:f`	Amount of memory added as floating memory.
hpvmmodify	`-a mem::`*`<amount>`*`-a mem::`*`<amount>`*`:b`	Increment the base memory by the specified amount to the given vPar.
	`-d mem::`*`<amount>`*`-d mem::`*`<amount>`*`:b`	Decrement the base memory by the specified amount to the given vPar.
	`-m mem::`*`<amount>`*`-m mem::`*`<amount>`*`:b`	Modify the base memory with the specified amount to the given vPar.
	`-a mem::`*`<amount>`*`:f`	Increment the floating memory by the specified amount to the given vPar.
	`-d mem::`*`<amount>`*`:f`	Decrement the floating memory by the specified amount to the given vPar.
	`-m mem::`*`<amount>`*`:f`	Modify the floating memory with the specified amount to the given vPar.

Table 38 Options to `vpar` and `hpvm` commands *(continued)*

Command	Option	Description
	`-R`	Cancel the pending memory migration operation.
`vparmodify`	`-a mem::`*<amount>*`-a mem::`*<amount>*`:b`	Increment the base memory by the specified amount.
	`-d mem::`*<amount>*`-d mem::`*<amount>*`:b`	Decrement the base memory by the specified amount.
	`-m mem::`*<amount>*`-m mem::`*<amount>*`:b`	Modify the base memory with the specified amount.
	`-a mem::`*<amount>*`:f`	Increment the floating memory by the specified amount.
	`-d mem::`*<amount>*`:f`	Decrement the floating memory by the specified amount.
	`-m mem::`*<amount>*`:f`	Modify the floating memory with the specified amount.
	`-C`	Cancel the pending memory migration operation.

Base or floating memory configuration rules

The following lists some of the base or floating memory configuration rules:

- When an attribute is not specified, the memory defaults to base. Hence, base memory can be added or deleted without specifying any attribute or by explicitly including the ‘:b’ attribute. The following lists syntaxes to add base memory:

  ```
  # vparcreate -p <vpar>  -a mem::<amount>[:b] ...
  # vparmodify -p <vpar>  -a mem::<amount>[:b] ...
  ```

  Alternatively,

  ```
  # hpvmcreate -P <vPar_name> -x vm_type=vpar -a mem::<amount>[:b]...
  # hpvmmodify -P <vPar_name> -a mem::<amount>[:b]...
  ```

- Floating memory requires explicit specification of the attribute ‘:f’ during add or delete. The following lists syntaxes to add floating memory:

  ```
  # vparcreate -p <vPar> -a mem::<amount>:f ...
  # vparmodify -p <vPar> -a mem::<amount>:f ...
  ```

  Alternatively,

  ```
  # hpvmcreate -x vm_type=vpar  -P <vPar_name> -a mem::<amount>:f ...
  # hpvmmodify -P <vPar_name> -a mem::<amount>:f ...
  ```

- Both base and floating memory can be added when the partition is up or down. But, to delete base memory, the partition must be down.
- Floating memory can be added or deleted when the partition is up or down.
- Base and floating memory can be added or deleted in one command line:

  ```
  # vparmodify -p <vPar> -a mem::<amount>:b -a mem::<amount>:f ...
  ```

  Alternatively,

  ```
  # hpvmmodify -P <vPar_name> -a mem::<amount>:b -a mem::<amount>:f ...
  ```

- Memory add and delete cannot be performed in the same command when the partition is Online. For example, if the vPar is online, the add and delete operations must be separated into two commands as follows:

  ```
  # vparmodify -p <vPar> -a mem::<amount>:b
  # vparmodify -p <vPar> -d mem::<amount>:f ...
  ```

  Alternatively,

  ```
  # hpvmmodify -P <vPar_name> -a mem::<amount>:b
  # hpvmmodify -P <vPar_name> -d mem::<amount>:f
  ```

- A memory add or delete and CPU add or delete operation cannot be performed in the same command when the vPar is Online. Hence, memory add or delete and CPU add or delete must be separated into two commands as follows:

  ```
  # vparmodify -p <vPar> -a cpu::<cores>
  # vparmodify -p <vPar> -d mem::<amount>:f
  ```

  Alternatively,

  ```
  # hpvmmodify -P <vPar_name> -a cpu::<cores>
  # hpvmmodify -P <vPar_name> -d mem::<amount>:f
  ```

- However, for a live partition, base memory add and floating memory modify operations can be performed in the same command; provided that the floating memory modify operation resulted in addition of floating memory.

  ```
  # vparmodify -p <vPar> -a mem::<amount>:b -m mem::<amount>:f....
  ```

  Alternatively, you can use `hpvmmodify` command:

  ```
  # hpvmmodify -P <vPar_name> -a mem::<amount>:b -m mem::<amount>:f
  ```

- A cancel operation is supported only for the last pending memory OL* operation.

  ```
  # vparmodify -p <vPar> -C
  ```

  Alternatively,

  ```
  # hpvmmodify -P <vPar_name> -R
  ```

- When upgrading vPars from earlier product versions, the total memory of the vPar would be marked as base memory in the new configuration file.
- If a VM guest is transformed into a vPar, then the total memory of the VM guest will be associated as base memory in the vPar configuration.
- If a vPar with floating memory is transformed to a VM guest (using hpvmmodify ‘vm_type’ option), the total memory (base + floating) will be associated with the VM guest. It will continue to operate with original base and floating memory configuration if reverted to a vPar again. However, if any memory operation was performed on the VM guest using hpvmmodify –r option, the total memory will be treated as base memory when the guest is transformed to a vPar.
- Base and floating memory of a partition is updated according to the following rules when `hpvmmodify –r` option is used to modify the total partition memory.

```
# hpvmmodify -P <vPar_name> -r <amount>
```

- If the specified amount of memory is greater than the current total memory, then, floating memory is incremented.
- If the specified amount of memory is less than the current total memory, then, floating memory is decremented first and if required based memory is also decremented.
- For a live partition, if the modify operation results in the decrement of base memory, online memory modification is not performed.

A very large increase in total memory using the `hvpmmodify -r` option makes floating memory value much larger than base memory. This can sometimes result in vPar panic during boot time. The `-r` option of the `hpvmmodify(1M)` command is deprecated for modification of vPar memory configuration. You can use `-a mem|-d mem|-m mem` options of the `hpvmmodify(1M)` command to modify memory of a vPar with the recommended base and floating memory values.

NOTE: There are some scenarios where online memory migration cannot be initiated. In such failure cases, the `hpvmmodify` command saves the new memory changes in the “next” configuration file, which is applied during the next boot of the vPar. On the contrary, the `vparmodify` command does not save any memory changes that cannot be dynamically applied. This is the existing behavior of the `hpvmmodify` and `vparmodify` commands.

An illustration of vPar online memory migration

This section describes the usage of command through an example of a vPar memory migration.

The memory migration operation is as follows:

1. Describe the experimental setup.
2. Describe memory usage on `vpar1` that has 2 GB of base memory.
3. Describe memory usage on `vpar1` after online addition of 4 GB of base memory and 4 GB of floating memory.
4. Describe memory usage on `vpar1` after online deletion of 4 GB of floating memory.

At each step, appropriate commands are executed to examine the memory usage and monitor the progress of the operation. Only the relevant output from the command is shown.

The setup used for this experiment is a system with 1 vPar, configured with 2 GB base memory. Following is the output of the `vparstatus` command with the memory distribution.

```
# vparstatus
[Virtual Partition Resource Summary]
Virtual  Partition     CPU        Num      Num     Total MB      Floating MB
Num      Name         Min/Max     CPUs     IO      Memory        Memory
======   ===========  =========== ======   =====   ==========    ============
1        vpar1         1/512       1         2      2048           0
# vparstatus -p 1 -v
[Virtual Partition Details]
Number:       1
Name:         vpar1
RunState:     DOWN
State:        Inactive
.......
[Memory Details]
Total Memory (MB):      2048
Floating Memory (MB):   0
.......
```

The overall memory available in the guest pool for memory allocation can be viewed by the following `vparstatus` command:

```
# vparstatus   -A
........
[Available Memory]:   411968 Mbytes
........
```

Now, the `vpar1` guest is booted.

```
# vparboot   -p 1
(C) Copyright 2000 - 2012 Hewlett-Packard Development Company, L.P.
Mapping vPar/VM memory:  2048MB
......
vparboot: Successful start initiation of vPar or VM 'vpar1'
```

At this point, you will notice that the overall memory available in the guest pool is reduced as some memory is used for booting the vpar1 guest.

```
# vparstatus   -A
......
[Available Memory]: 409792 Mbytes
.......
```

The following shows the `vparmodify` command that is used to add 4 GB of base memory and 4 GB floating memory to the vpar1 guest online.

```
# vparmodify -p 1 -a mem::4G -a mem::4G:f
vparmodify: A Memory OLAD operation has been initiated for this vPar.
Please check vparstatus output or syslog for completion status.
```

You can verify the Memory OL* completion status as follows:

```
# vparmodify -p 1 -v
[Virtual Partition Details]
Number:       1
Name:         vpar1
RunState:     UP
State:        Active
.......
[Memory Details]
Total Memory (MB):     10240
Floating Memory (MB):  4096
......
[Memory OL* Details]
Operation:               MEM change
Base Memory (MB):        4096
Floating Memory (MB):    4096
Status:                  PASS
........
```

You can also find information about the completion status in the guest log file:

```
# tail  /var/opt/hpvm/guests/vpar1/log
......
Trying to add Base:   4096 MB, Float: 4096 MB
Added Base 4096 MB, Float:  4096 MB
```

You can verify the new size of the `vpar1` guest by using the `vparstatus` command.

```
# vparstatus
........
[Virtual Partition Resource Summary]
Virtual  Partition     CPU        Num      Num     Total MB     Floating MB
Num      Name         Min/Max     CPUs     IO      Memory       Memory
======   ===========  =========== ======   =====   ==========   ============
1        vpar1         1/512       1         2      10240       4096
```

At this point, you will notice that the overall memory available in the guest pool is further reduced as some of the memory is added online to the `vpar1` guest.

```
# vparstatus   -A
......
[Available Memory]: 401600 Mbytes
.......
```

Now, 4 GB of floating memory is removed from the same guest using the `vparmodify` command.

```
# vparmodify -p 1 -d mem::4G:f
vparmodify: A Memory OLAD operation has been initiated for this vPar.
Please check vparstatus output or syslog for completion status.
```

```
# vparstatus   -p 1 -v
[Virtual Partition Details]
Number:        1
Name:          vpar1
RunState:      UP
State:         Active
.......
[Memory Details]
Total Memory (MB):      6144
Floating Memory (MB):   0
.......
[Memory OL* Details]
Operation:              MEM change
Base Memory (MB):       0
Floating Memory (MB):   4096
Status:                 PASS
.......
```

```
# vparstatus
........
[Virtual Partition Resource Summary]
Virtual  Partition     CPU         Num      Num     Total MB     Floating MB
Num      Name         Min/Max      CPUs     IO      Memory       Memory
======   ===========  =========== ======   =====   ==========   ============
1        vpar1         1/512        1        2       6144        0
```

At this point, you will notice that the overall memory available in the guest pool is increased from earlier, as some of the memory was deleted online from the `vpar1` guest.

```
# vparstatus  -A
........
[Available Memory]:  405696 Mbytes
..........
```

For more information about the online memory migration, see *Reconfiguring vPars v6 memory with zero downtime* at **http://www.hpe.com/info/hpux-hpvm-docs**.

Online CPU migration for vPar

Online CPU migration is supported for vPars. An illustration is as follows.

NOTE: At each step, appropriate commands are executed to verify the CPU count on the vPar and monitor the progress of the operation. Only the relevant output from each command is shown.

In this example, the VSP with several vPars and VM guests is configured as follows:

```
# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar0004             4     VP    HPUX    On (OS)   1       2     1    2048 MB
SHVM0005             5     SH    HPUX    On (OS)   1       1     2       2 GB
SHVM0007             7     SH    HPUX    On (OS)   1       1     1       2 GB
vPar0003             3     VP    HPUX    On (OS)   1       1     1    2048 MB
vPar0001             1     VP    HPUX    On (OS)   1       1     1    2048 MB
```

```
vPar0002              2      VP     HPUX    On (OS)   1      1      1    2048 MB
SHVM0006              6      SH     HPUX    On (OS)   1      1      1       2 GB
```

```
# vparstatus
```

```
[Virtual Partition]
Num Name                           RunState      State
=== ========================== ============  =========
4   vPar0004                       UP            Active
3   vPar0003                       UP            Active
1   vPar0001                       UP            Active
2   vPar0002                       UP            Active

[Virtual Partition Resource Summary]
Virtual Partition     CPU     Num    Num    Total MB    Floating MB
Num     Name        Min/Max   CPUs   IO     Memory      Memory
======= =========   =======   ====   ====   ========    ===========
4       vPar0004     1/512     1      3     2048        0
3       vPar0003     1/512     1      2     2048        0
1       vPar0001     1/512     1      2     2048        0
2       vPar0002     1/512     1      2     2048        0
```

As seen in the output, `vPar0001` is currently running with a single CPU.

You can look at the number of CPUs that are available for vPars.

```
# vparstatus -A
```

```
[Available CPUs]:  8

[Available Memory]:  42048 Mbytes
…
```

Now, you can add 4 CPUs to vPar0001 using the `vparmodify` command.

```
# vparmodify -p 1 -a cpu::4
```

```
vparmodify: A CPU OLAD operation has been initiated for this vPar.
Please check vparstatus output or syslog for completion status.
```

```
# vparstatus -p 1 -v
```

```
[Virtual Partition Details]
Number:    1
Name:      vPar0001
RunState:  UP
State:     Active
…
[CPU OL* Details]
Operation : CPU change
CPU Count: 5
Status: PASS
…
```

As seen in the output, the `vparstatus` command shows that a total of 5 CPUs are now configured in the live vPar.

NOTE: You can run the `evmget/evmshow` command from within the vPar being modified to verify the progress of the operation.

The `vparstatus -A` command reflects the reduction of 4 CPUs from the free pool.

```
vparstatus -A
```

```
[Available CPUs]:  4

[Available Memory]:  42048 Mbytes
…
```

Now, we can use the `vparmodify` command again to delete 4 CPUs from the online vPar.

```
# vparmodify -p 1 -d cpu::4
```

```
vparmodify: A CPU OLAD operation has been initiated for this vPar.
Please check vparstatus output or syslog for completion status.
```

```
# vparstatus -p 1 -v
```

```
[Virtual Partition Details]
Number:    1
Name:      vPar0001
RunState:  UP
State:     Active
…
[CPU OL* Details]
Operation : CPU change
CPU Count: 1
Status: PASS
…
```

As seen in the output, the vPar `vPar0001` is running with one CPU.

Dynamic I/O for vPars and Integrity VM guests

All I/O devices supported within vPars and Integrity VM guests may be dynamically added or deleted from running instances of vPars and Integrity VM guests; this capability is called as Dynamic I/O.Dynamic addition of I/O devices was first made available with HP-UX vPars and Integrity VM v6.3. The dynamic deletion is available from v6.3.5.

NOTE: Dynamic I/O must not be confused with the Direct I/O functionality using which Ethernet network adapters are directly presented to vPar and Integrity VM guests.

Dynamic I/O functionality is conceptually similar to online addition or removal of I/O devices on physical servers, using *olrad*(1M). While *olrad*(1M) functionality is available only on platforms with OLARD capability, dynamic I/O, is not dependent on capabilities of the VSP. It may be used on vPars and Integrity VM guests running on any HPE Integrity system which may be used as a VSP. All types of I/O devices supported for usage within vPar and Integrity VM guests can be added or deleted dynamically.

Operational details

Dynamic addition and deletion are asynchronous operations. The operation is initiated using *hpvmmodify*(1M) or *vparmodify*(1M); the command performs basic validation, informs the virtualization layers to proceed with the operation and returns immediately. The functional operation proceeds to completion in the background. The `-V` option to *hpvmstatus*(1M) is used to display the status of the last dynamic I/O operation.

Devices are added or deleted from vPars and Integrity VM guests using *hpvmmodify*(1M) or *vparmodify*(1M); the `-a` (for addition) or `-d` (for deletion) option is used along with appropriate resource specification as given in *hpvmresources*(5).

Devices may be added at a desired specific location within the PCI bus hierarchy by specifying a free bus and slot location in the resource specification. The `-V` option for *hpvmstatus*(1M) displays all used and reserved PCI I/O device slots for the specified vPar or Integrity VM guest. This information may be used to select a free slot where an I/O device must be dynamically added.

NOTE: The I/O slot at PCI bus 0, device 3 is a core I/O slot for vPars and Integrity VM guests. Dynamic I/O operations are not permitted on this slot.

For each vPar or Integrity VM guest, only one dynamic I/O operation may be operational at any given time; multiple operations or devices cannot be combined together into one command. Further, no other modification operation may be combined with a dynamic I/O operation.

Dynamic I/O operations can be run only when the guest configuration is in stable state. It cannot be run while

- Previous dynamic I/O operations or PCI OLR operations are in progress.
- The target of the operation is an Integrity VM guest that is being or has been suspended

  or
- The target of the operation is an Integrity VM guest being migrated.

Errors and failure logs

Errors in command parameters or conditions which prevent the operation from being initiated are reported immediately in the output of *hpvmmodify*(1M) and *vparmodify*(1M). Operational failures are logged in the guest log file at `/var/opt/hpvm/guests/<Guest Name>/log`. Errors within the vPar or Integrity VM guest are logged in `/var/adm/syslog/syslog.log` of the vPar or Integrity VM guest.

vPar or VM log files

Each vPar or VM guest has a log file named `/var/opt/hpvm/guests/`*`<Guest Name>`*`/log` on each VSP.

The VSP log files are stored as `/var/opt/hpvm/common/command.log` and `hpvm_mon_log`.

The `command.log` file contains the entries in the following formats:

`mm/dd/yy hh:mm:ss|process_id|message_type|owner|user|Message`

process_id
: field captures the process id of the program which logs the message

Message_type
: indicates the nature of message, such as "ERROR", "WARNING", SUMMARY, NOTE, CHANGE

Owner
: indicates whether it is running on behalf of host or guest. This field usually logs the guestname when hpvm* command is executed on particularguest

user
: indicates the user-name of the process which has logged the message

---

NOTE: A `Failed API access to local running guest.` message in the `command.log` is a notification that a communication attempt with the `hpvmapp` process has failed. This message is not an indication of a problem and can be ignored.

Managing the device database

A vPar or VM guest cannot detect all potential backing store conflicts, and does not always prevent misconfigured vPars or VM guests from booting. Conflicts can arise from the following:

- Specifying the same backing store for more than one virtual device.

  If you add `disk:avio_stor::disk:/dev/rdisk/disk2` for guest A, do not add the same device to another guest or to the list of VSP restricted devices.
- Specifying multiple backing store parameters that lead to the same physical storage.

  If the VSP has multiple paths to a storage device, such as `/dev/rdisk/disk0` and `/dev/rdisk/disk4`, only one path must be specified for a `disk:avio_stor` or `dvd:avio_stor` in guest A. The other path must not be used as a backing store by guest A or by any other guest or the VSP.

- Overlapping physical storage allocated for different backing store types.

  If a guest uses a logical volume (for example, `rlvol1`) as a backing store device, the disks used by the volume group on which the logical volume is made (for example, `/dev/vg01`) cannot be used as backing stores.

You can use the `ioscan` command to detect these conflicts. If you force guests configured with these conflicts to start, the data might get corrupted.

Do not use Veritas VxVM DMP device files (files under `/dev/vx/rdmp`) used as a backing store for a guest root disk, on the VSP. If this is done, then explicitly run `insf -e` so that partitions on DMP node gets reflected on the physical disk as well. If you do not run `insf -e` there is no way for VxVM to communicate the partition information on the DMP nodes to HPUX I/O tree.

NOTE: If DMP naming scheme changes, then you have to update guest configuration file using the `hpvmmodify` command.

SCSI information will be displayed only for DMP devices presented through NPIV.

On the VSP, do not extend a logical volume (LVM or VxVM) used as a backing store for a guest root disk. If you do this, the guest panics on its next reboot with the following error:

```
System panic: all VFS_MOUNTROOTs failed: Need DRIVERS.
```

The guest must be able to boot if the logical volume is reverted (using `lvreduce` in case of LVM) to its original size. If this fails, the guest root device is corrupted, and the guest operating system must be reinstalled.

An AVIO logical volume backing store not used as a root disk can be extended while the guest is online. For HP-UX 11i v3 guests using AVIO, the guest is notified of the increased size of the backing store for logical volumes and raw disks, and the guest can take appropriate actions to use the larger size.

After you extend the logical volume, use operating system commands on the guest to extend its file system.

NOTE: When you create a file system using the `sam` command on an HP-UX guest, do not initialize the disk. It returns an error and the file system is not created.

VM or vPars device database file

The vPar or VM guest device management stores vPar or VM guest device mapping information in the device database file (`/var/opt/hpvm/common/hpvm_mgmtdb`). This file is divided into three sections:

- The header, which states that the file cannot be hand edited.
- The restricted device section, which contains a list of host devices that guests are not allowed to access.
- The guest devices section, which contains devices, both storage and network, that guests are configured to use.

Do not edit the `hpvm_mgmtdb` file directly unless you are specifically advised to do so. Always use supported Integrity VM commands (such as `hpvmmodify` or `hpvmdevmgmt`) to modify virtual devices.

Using the `hpvmdevmgmt` command

To view and modify the devices used by the VSP and the vPar or VM guests, use the `hpvmdevmgmt` command.

Table 39 (page 272) lists the options that can be used with the `hpvmdevmgmt` command.

Table 39 Options to the `hpvmdevmgmt` command

Option	Description
`-l {server|rdev|gdev}:`*`entry_name`*`:attr:`*`attr_name=attr_value`*	Lists an entry. To list all entries, enter the following command: `# ` **`hpvmdevmgmt -l all`**
`-v`	Displays the version number of the `hpvmdevmgmt` output format. The version number is followed by the display specified by other options.
`-V`	Increases the amount of information displayed (verbose mode).
`-S` *`size filename`*	Creates a file for use as a virtual device. The size argument must end in either M for megabyte or G for gigabyte.
`-I`	Creates passthrough device files (for example, `/dev/rscsi`). Passthrough devices are used by attached devices, such as tape devices, media changers, and CD or DVD burners.
`-m {server|rdev|gdev}:`*`entry_name`*`[:attr:`*`attr_name=attr_value`*`]`	Modifies an existing attribute or adds the attribute if it does not already exist.
`-a {server|rdev|gdev}:`*`entry_name`*`[:attr:`*`attr_name=attr_value`*`]`	Adds an entry.
`-d {server|rdev|gdev}:`*`entry_name`*`[:`*`param`*`:`*`arg`*`]`	Deletes an entry.
`-d gdev_alias:/dev/rdisk/disk`*`nn`*	Deletes one alias if a device has multiple aliases defined.
`-n gdev:`*`oldentry_name`*`:`*`newentry_name0`*`[,`*`newentry_name1`*`]`	Replaces a device.
`-r`	Generates a report script that can be used after inspection to fix various device database problems.

For example, to view a list of the restricted devices, enter the following command:

```
# hpvmdevmgmt -l rdev
/dev/rdisk/disk4:CONFIG=rdev,EXIST=YES,DEVTYPE=DISK,
SHARE=NO::6005-08b4-0001-15d0-0001-2000-003a-0000
```

To make a device shareable among guests, enter the following command:

```
# hpvmdevmgmt -m gdev:/data/file.iso:attr:SHARE=YES
```

NOTE: Whenever you add a device that is going to be used in guest configurations to an Integrity VSP, run the `hpvmdevmgmt -I` command after adding the device to the host.

Sharing devices

With Integrity VM, you can allow devices to be specified as either shared or not shared. By default, vswitches are configured to be shared, and storage devices are configured to not be shared. As administrator, you can configure a storage device to be shared by multiple guests.

The `SHARE` attribute is checked only when booting a guest. If one guest is running with a nonshared device and another guest attempts to boot using that same device, the latter guest is blocked. If multiple guests must share devices, then the `SHARE` attribute for those devices must be changed to `SHARE=YES` using the modify option (`-m`) with the `hpvmdevmgmt` command.

For example, to make the HP-UX iso.* images shareable so that two VMs (`host1` and `host2`) can use them to install at the same time, enter the following commands:

```
# hpvmdevmgmt -m gdev:/var/opt/hpvm/ISO-images/hpux/:attr:SHARE=YES
# hpvmmodify -P host1 -a dvd:avio_stor::null:/var/opt/hpvm/ISO-images/hpux/
# hpvmmodify -P host2 -a dvd:svio_stor::null:/var/opt/hpvm/ISO-images/hpux/
```

Virtual DVDs and virtual network devices can be shared. DVDs are not shareable unless you specify otherwise. Sharing of virtual devices or hardware backing stores must be carefully planned in order to prevent the data getting corrupted.

To restrict the vswitch named `myswitch` so that it is no longer shareable, enter the following command:

```
# hpvmdevmgmt -m gdev:myswitch:attr:SHARE=NO
```

This command restricts the vswitch called `myswitch` to be used by one guest only.

Replacing devices

If a backing storage device malfunctions, replace it by using the `hpvmdevmgmt -n` option. The `-n` option works for only guest devices. It replaces the existing device entry with the new device entry while keeping all the current guest dependents. Thus, each guest dependent is modified to replace the old device with the new one. If the device being replaced is a pNIC, use the `hpvmnet` command to halt and remove the current vswitches using that pNIC, and recreate the same named vswitches using the new pNIC. This method allows guests to use the new pNIC through the old vswitch names without modifying the guests.

Deleting devices

A device entry can be deleted only if it has no dependents. If a device has dependents, those dependents must be removed before you delete the device. The `hpvmmodify` command that removes a device removes that guest as a dependent on that device.

If the guest cannot be modified, you can use the `hpvmdevmgmt -d` command to delete a dependent from a device. However, this command does not modify the guest that is dependent on the device. Use this method only if you can use the `hpvmmodify` command on the guests that are dependent on the device. The following example shows how to remove a guest as a dependent:

```
# hpvmdevmgmt -d gdev:entry_name:depend:depend_name
```

Restricting VSP devices

You must set up restricted devices to ensure that no guest uses devices that are reserved for use by the VSP, including the storage devices that the VSP uses to boot and run. This can also include a network LAN device to which the host requires exclusive access.

If a volume manager is used for host-specific file systems, then the restricted devices must include both, the volume devices and the underlying special device files to protect both from guest access. For more information about storage devices, see “Reserved resources and resource over-commitment” (page 54).

You can also allow guests to access certain files while restricting them from accessing the device files that contain those files. You can add or delete restricted device entries to the Integrity VM device database.

For example, to add `/dev/rdisk/disk0` as a restricted device, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk0
```

To delete the restricted device `/dev/rdisk/disk0`, enter the following command:

```
# hpvmdevmgmt -d rdev:/dev/rdisk/disk0
```

To add network `lan0` as a restricted device, enter the following command:

```
# hpvmdevmgmt -a rdev:lan0
```

If the configuration file of the guest contains restricted devices, the guest does not start.

Inspecting and editing the repair script

The `hpvmdevmgmt -r` report and repair-script function might identify one or more new pathnames for disks whose old pathnames no longer exist. The repair-script performs that reassignment using the `hpvmdevmgmt -n` command.

In general, you must inspect and edit the script before running it for the following reasons:

- All replace commands, `hpvmdevmgmt —n`, in the script are commented out. You must delete only the comment characters before only one of the `hpvmdevmgmt —n` commands for a particular device. Otherwise, subsequent `hpvmdevmgmt -n` commands for the same device fails.
- If a legacy device name is replaced with another legacy device name, both, the legacy device name and the agile device name are added. However, if the agile device name is used to replace a legacy device name, only the agile device name is used.

Attributes that can be changed dynamically

A dynamic change does not require a reboot of the virtual environment in question. Table 40 (page 274) lists the attributes that can be changed dynamically.

Table 40 Attributes changed dynamically

Attribute	vPars	VMs
CPU **1.** Changing vPar or VM vCPU entitlement. The default is uncapped mode. In uncapped mode, this is also automatic based on overall “free” entitlement. **2.** Enabling or disabling vCPUs from within a vPar/VM. **3.** Adding or removing CPUs to and from a vPar/VM from the VSP.	**1.** N.A. **2.** No **3.** Yes	**1.** Yes **2.** Yes **3.** No
Memory **1.** Adding or removing the memory in use by a vPar/VM. **2.** Making it automatic with AMR (Automatic Memory Reallocation) based on overall “free” memory.	**1.** Yes (Floating memory. Base memory cannot be deleted from a live vPar) **2.** No	**1.** Yes (Adding to a VM is limited to the maximum size it booted with) **2.** Yes
Network: **1.** Adding or removing virtual switches (vswitches) on the VSP. **2.** Removing vswitches on the VSP if the ports of the vswitch are not assigned to a guest. **3.** Adding and Deleting ports of a vswitch to an online guest.	**1.** Yes **2.** Yes **3.** Yes	**1.** Yes **2.** Yes **3.** Yes

Table 40 Attributes changed dynamically *(continued)*

Attribute	vPars	VMs
Storage • Adding or removing storage to or from a vPar/VM. **NOTE:** Depending on the type of storage being used, there may be additional steps required. See Section (page 70)	Yes	Yes
Migration **1.** Migrating online. **2.** Migrating offline.	**1.** Yes **2.** Yes	**1.** Yes **2.** Yes

NOTE: Before you add or remove memory, networking, or storage from a vPar or a VM, ensure you know if further action is required on the vPar or VM.

HPE AVIO Stor EFI Driver enumeration policy

The default enumeration policy of the “HPE AVIO Stor EFI Driver” is to enumerate boot LUNs. Use the `drvcfg` EFI utility to change the enumeration policy to do the following:

- Enumerate boot LUNs only. (Default policy)
- Enumerate all LUNs.

The enumeration policy can be set separately for SCSI (non-NPIV) LUNs and FC (NPIV) LUNs. Setting the policy to enumerate all LUNs (especially FC LUNs) might result in long guest boot time in configurations with a large number of LUNs. The delay might be noticed in the following cases:

- The EFI Boot Manager menu screen takes a long time to present itself.
- When entering the EFI shell, a long delay might occur before the device mappings are displayed and the EFI shell prompt is presented.

The following example shows the policy configuration dialog. In this example, the policy is unchanged from the default policy.

```
Shell> drvcfg -s
HP AVIO Stor Driver Configuration
==================================
Warning: enumerating all SCSI or FC LUNs increases initialization times.

Enumerate all SCSI LUNs (Y/N)? [current setting: N]: N

Enumerate all FC LUNs (Y/N)? [current setting: N]: N


  Drv[2F]  Ctrl[ALL]  Lang[eng] - Options set. Action Required is None
None
None

Shell>

Reset the guest for the change to take effect

   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[g1] vMP> CM

        (Use Ctrl-B to return to vMP main menu.)

[g1] vMP:CM> RS
```

```
At next boot only boot LUN will be enumerated

    Use ^ and v to change option(s). Use Enter to select an option
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.62]onsole - - - - - - - - - - - -
Device mapping table
  fs0  : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part1,SigBEC59C34-E6C8-11DB-8002-D6217B60E588)
  fs1  : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part3,SigBEC59C70-E6C8-11DB-8004-D6217B60E588)
  blk0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)
  blk1 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part1,SigBEC59C34-E6C8-11DB-8002-D6217B60E588)
  blk2 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part2,SigBEC59C52-E6C8-11DB-8003-D6217B60E588)
  blk3 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part3,SigBEC59C70-E6C8-11DB-8004-D6217B60E588)
startup.nsh> echo -off

  setting hpux path(\EFI\HPUX)...
  type 'fs[x]:' where x is your bootdisk (0, 1, 2...)
  type 'hpux' to start hpux bootloader
```

Changing the policy to enumerate all AVIO storage devices might result in longer guest boot times (to the EFI level), depending on the guest's configuration.

If you must boot from a tape device attached to an NPIV (such as performing tape-based Ignite-UX recovery), change the enumeration policy to "Enumerate all FC LUNs". As mentioned previously, enumerating all FC LUNs can result in a long guest boot time. To minimize this delay, you can temporarily reduce the number of NPIV HBAs for the VM to only the one containing the tape boot device.

16 Managing vPars and VMs using GUI

There are multiple user friendly GUI tools to manage vPars and VMs. This chapter describes how you can manage vPars or VM guests using GUI tools such as VSMgr and HPE Matrix OE.

Managing VMs with VSMgr

HP Integrity Virtual Server Manager is the GUI that you can use from your browser to manage Integrity VM resources. Integrity Virtual Server Manager allows you to create, configure, and manage VMs or vPars, and to monitor and evaluate data and resources at the level of the VSP. You can view all VMs and vPars of a VSP and also the resources assigned to the VSP or to a specific VM, vPar or virtual switch. For example, Integrity Virtual Server Manager provides graphical views of virtual-to-physical network and storage devices so that you can view I/O data, including resource utilization information. Integrity Virtual Server Manager obtains information about Integrity VM resources through Web-Based Enterprise Management (WBEM) providers installed on the VSP and on VMs or vPars (guest operating systems).

To start the VSMgr tool from SMH, open the SMH on the VSP (you can type the link in a browser as http://<VSP IP address/hostname>:2301. The VSMgr is available under the tools link in SMH. A VSMgr icon is displayed in the Matrix OE visualization page of HP System Insight Manager CMS. For more information about VSMgr, see the documents at **http://www.hpe.com/info/matrixoe/docs** and VSMgr Online Help

NOTE: For more information about Dynamic memory restrictions, see “Dynamic memory restrictions” (page 259).

Managing vPars and VM guests with HPE Matrix OE

HPE Matrix OE is an integrated command center that enables you to analyze and optimize your cloud and converged infrastructure. It builds on the HPE infrastructure management portfolio, including HP SIM and HP Insight Management.

Matrix OE provides an integrated graphical environment for managing physical servers, logical servers, VMs, server blades, nPartitions, vPars, applications, and workloads.

You can dynamically resize virtual servers and migrate resources where they are needed, based on service-level objectives and business requirements.

CAUTION: All vPars and Integrity VM versions are not compatible with all Matrix OE versions. For more information about compatibility between Matrix OE and vPars and Integrity VM, see *Insight Management Support Matrix* available at **http://www.hpe.com/info/matrixoe/docs**.

Managing VMs with HPE Matrix Infrastructure Orchestration

HPE Matrix Infrastructure Orchestration extends HPE Matrix OE to provide rapid provisioning and repurposing of infrastructure services from shared compute resource pools using a Self Service Portal. Matrix infrastructure orchestration delivers advanced template-driven design, provisioning, and ongoing operations for multi-node, multi-tier infrastructure services built around the following Hewlett Packard Enterprise platforms:

- HP Insight Control, including Insight Control virtual machine management
- HPE Virtual Connect Enterprise Manager
- HPE Matrix Operating Environment

The following types of backing stores are supported for use with Matrix Infrastructure Orchestration V7.2 onwards:

- NPIV LUNs
- SLVM-based logical volumes (LVs)

For more information about Infrastructure Orchestration and CloudSystem Matrix for HP-UX, see **http://www.hpe.com/info/cloudsystem**.

Managing vPars and Integrity VMs from HPE Matrix Operating Environment Logical Server Management

A logical server is a set of configuration information that you create, activate, and move across physical servers and VMs. It contains the logical server definition and description, including the server compute resources (for example, number of CPU cores and amount of memory), and the server connections to storage fabric and networks.

Most of the logical server operations (Create, Import, Move, Copy, and so on) are now supported for Integrity VMs and vPars.

The following types of backing stores are supported for use with HPE Matrix OE Logical Server Management:

- Whole disk backing stores consisting of SAN LUNs
- NPIV LUNs
- SLVM-based logical volumes (LVs)

For more information about the supported set of operations for vPars and Integrity VM and for information about storage and networking configurations, see the latest *Matrix Operating Environment Logical Server Management User Guide* at **http://www.hpe.com/info/matrixoe/docs**.

NOTE: For more information about Dynamic memory restrictions, see Section (page 259).

Configuring guest backing storage with HPE Matrix OE

This section describes how to configure the following backing stores with HPE Matrix OE:

- NPIV LUNs

  Matrix OE version 7.2 onwards supports NPIV based backing stores. This is the preferred backing store for managing Integrity vPars and VM guests as it offers various advantages as described in Section (page 99).

  All LSM and IO operations are supported with NPIV backing store from Matrix OE 7.2 onwards.

  To verify if your VSP can support NPIV, use the `fcmsutil` command as specified in “Verifying whether VSP can support NPIV” (page 100).

- Whole disk backing stores consisting of SAN LUNs

  The supported operations for this type of backing store in LSM are Import, Move, Power On, Power Off, and Unmanage.

  NOTE: This type of backing store is not supported with Matrix Infrastructure Orchestration.

- SLVM-based logical volumes (LVs)

To use SLVM-based logical volumes (LVs):

1. Create an appropriate sized SLVM volume group for the device management database using LVM Version 2.2. For example:

   Create the volume group using LVM Version 2.2:

   ```
   # vgcreate -V 2.2 -s 4m -S 100g /dev/slvm_v21 /dev/disk/disk61
   ```

   For information about creating SLVM volume groups, see *SLVM Online Volume Reconfiguration* at **http://www.hpe.com/info/hpux-LVM-VxVM-docs**.

2. Add SLVM volume groups into the device database using the `hpvmdevmgmt` command. For each SLVM volume group you add to the device management database, set the device attribute *`VIRTPTYPE`* to *`container_volume_SLVM`*, with the `PRESERVE=YES` attribute setting. For example:

   ```
   # hpvmdevmgmt -a gdev:/dev/slvm_v22:attr:VIRTPTYPE=container_volume_SLVM,PRESERVE=YES
   ```

3. Run the `hpvmhostrdev —u` command to add the underlying disks of the (just created) SLVM volume groups into the device database as restricted devices.

   NOTE: The SLVM volume groups must be in the activated mode before running the `hpvmhostrdev` script. For information about deactivated volume groups, see "Storage for deactivated volume groups not protected by VM storage management" (page 279).

4. Run the `hpvmhostgdev -a` command to ensure that all the devices are populated in the `gdev` database. The `hpvmhostgdev` command analyzes `disklist` and `lvlist` output and adds unused `gdevs` to the device database.

   NOTE: If you add new devices in the future, run the `hpvmhostgdev -a` script again. If you want to select the guest devices instead of adding all of them to the `gdev` database, create a list of unused disks and logical volumes with the `-l` option and pipe them to a file. Use the specified device-list file to add devices for guest use with the `-f` option.

   ```
   # hpvmhostgdev -l > devicelist
   # hpvmhostgdev -f devicelist
   ```

   For information about the `hpvmhostgdev` script, see *hpvmhostgdev*(1M).

5. Managing VMs does not require them to be in a VM as a Serviceguard Package. However, if you plan to use clustered VMs, ensure that the VSP is properly configured with Serviceguard (11.19 or 11.20) and SLVM.

   NOTE: For information about configuring Serviceguard and SLVM, see *Managing HP Serviceguard A.12.00.00 for Linux* manual.

   If you already have your VMs clustered in a VM as a Serviceguard Package, but prefer not to manage them this way, you can use the `cmdeployvpkg` Serviceguard command to deconfigure (delete) the package. For information about the `cmdeployvpkg` command, see the *Serviceguard Toolkit for Integrity Virtual Servers User Guide* at **http://www.hpe.com/info/hpux-serviceguard-docs**.

Storage for deactivated volume groups not protected by VM storage management

When an LVM volume group is deactivated, the physical volumes used by that storage are designated as unused by HP-UX system administration tools such as SMH. This is also true for Integrity VM storage management. As a result, these physical volumes are not automatically protected from use by VMs as virtual disks.

You can resolve this problem in one of two ways:

- If the volume group is to remain deactivated, the VSP administrator can manually add the physical volume as a restricted device using the `hpvmdevmgmt` command.
- If the VSP storage management database is to be updated, run the `hpvmhostrdev` command, after activating the volume group.

An HP-UX system administrator can deactivate a volume group using the `vgchange` command. It can also be deactivated if it is an SLVM volume group, whenever the associated Serviceguard cluster is reconfigured, or the VSP system is rebooted. You must ensure that all SLVM volume groups are activated after a VSP reboot or Serviceguard cluster reconfiguration.

Matrix OE troubleshooting

This section lists some common CLI commands that helps when troubleshooting the issues when using vPars and Integrity VM with Matrix OE.

Adding and removing devices

Most of the VSP devices get added into vPars and Integrity VM device database automatically. You can add devices that are not automatically added by using the `hpvmdevmgmt gdev PRESERVE` attribute. The following device types require manual addition:

- File backed disks
- File backed DVDs
- VxVM volumes

The following examples show how to add various device types to the storage pool:

- File:

  ```
  # hpvmdevmgmt -a gdev:/var/opt/hpmv/ISO-images/hpux/112350GOLD.ISO:attr:PRESERVE=YES
  ```

- VxVM volume:

  ```
  # hpvmdevmgmt -a gdev:/dev/vx/rdisk/guestdg/vxvm_g2:attr:PRESERVE=YES
  ```

To remove a device from the storage pool, run the following command:

```
# hpvmdevmgmt -d gdev:/dev/rdisk/disk23
```

NOTE: Adding devices to the storage pool does not prevent them from being used by the HP-UX operating system or other Integrity VM commands.

The storage pool does not fully support lunpaths or directories. In addition, Virtual Machine Management (VMM), a layer between Integrity VM and LSM, has no way to insert or eject a DVD, because this is done from the virtual console.

Registering and unregistering a VM

The following vPars and Integrity VM properties are set when a guest is registered in Matrix OE.

- `runnable_status=enabled`
- `modify_status=enabled`
- `visible_status=enabled`

Matrix OE ensures that a VM is registered (and, therefore, runnable) on only one VSP at a time.

When a VM is unregistered in Matrix OE, the following attributes are set:

- `runnable_status=disabled`
- `modify_status=disabled`
- `visible_status=disabled`

When Matrix OE queries the `register_status`, the value of `visible_status` is returned. If the VM is not visible, you cannot visualize it with the graphical tools, and therefore; you cannot modify it or run it.

You can set the `register_status`, `modify_status`, `visible_status` and `runnable_status` of a VM to enabled or disabled with the `hpvmmodify -P vmname -x` command. The following are the examples:

```
# hpvmmodify -P vmname -x runnable_status={enabled|disabled}
# hpvmmodify -P vmname -x modify_status={enabled|disabled}
# hpvmmodify -P vmname -x visible_status={enabled|disabled}
# hpvmmodify -P vmname -x register_status={enabled|disabled}
```

CAUTION: Hewlett Packard Enterprise does not recommend using any of the earlier options except with extreme caution. Integrity VM commands ensure that the VM is registered only on one VSP at a time. Registering a VM on more than one VSP can lead to accidentally booting the VM on more than one VSP and can cause inconsistencies with the display of graphical tools. Any modification made to the configuration of the VM will be lost when it is migrated back to this VSP.

However, if you find that VM is not registered on any other VSP, you can manually use the earlier commands.

Cannot distinguish between JBOD and Remote SAN with device check

If your VSP has local JBOD disks configured, they appear as disks that are SAN-resident in the Virtualization Provider making them available for guests. If your guest configurations require only SAN-resident disks, the JBOD disks, set them as restricted disks in the Integrity VM device database.

The following example sets the device `/dev/rdisk/disk100` as a restricted device:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk100
```

Unpresenting SAN devices to Integrity VSPs

Unpresenting SAN devices that were configured to be used by guests causes the guest to fail to start. If SAN devices must be unpresented, guests configured to use those devices must be reconfigured to no longer require them. After unpresenting a device special file, remove it from the Integrity VSP using the following command:

```
# rmsf  -a  device_special_file
```

The device special file can be derived from the wwid_string, obtained from the SAN appliance, as follows:

```
# scsimgr -p get_attr -a wwid -a device_file current all_lun | grep wwid_string
```

17 Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  www.hpe.com/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates, go to either of the following:
  - Hewlett Packard Enterprise Support Center **Get connected with updates** page:

    www.hpe.com/support/e-updates

  - Software Depot website:

    www.hpe.com/support/softwaredepot

- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Websites

Website	Link
Hewlett Packard Enterprise Information Library	**http://www.hpe.com/info/enterprise/docs**
Hewlett Packard Enterprise Support Center	**http://www.hpe.com/support/hpesc**

Website	Link
Contact Hewlett Packard Enterprise Worldwide	**http://www.hpe.com/assistance**
Subscription Service/Support Alerts	**http://www.hpe.com/support/e-updates**
Software Depot	**http://www.hpe.com/support/softwaredepot**
Customer Self Repair	**http://www.hpe.com/support/selfrepair**
Insight Remote Support	**http://www.hpe.com/info/insightremotesupport/docs**
Serviceguard Solutions for HP-UX	**http://www.hpe.com/info/hpux-serviceguard-docs**
Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix	**http://www.hpe.com/storage/spock**
Storage white papers and analyst reports	**http://www.hpe.com/storage/whitepapers**

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

For more information and device support details, go to the following website:

www.hpe.com/info/insightremotesupport/docs

Related information

The following documents [and websites] provide related information:

Table 41 Documentation and its location

<table>
<tr><th>Documents</th><th>Website</th></tr>
<tr><td>HP-UX GUID Manager Administrator Guide</td><td>http://www.hpe.com/info/hpux-vpars-docs and http://www.hpe.com/info/insightdynamics-manuals</td></tr>
<tr><td>Integrity Virtual Server Manager User Guide
Integrity Virtual Server Manager Release Notes
NOTE: The HP Integrity Virtual Server Manager is the graphical user interface version of the command-line interface HP-UX vPars and Integrity VM.</td><td>http://www.hpe.com/info/matrixoe/docs and http://www.hpe.com/info/insightdynamics-manuals</td></tr>
<tr><td>BladeSystem onboard administrator command line interface user guide version 4.50
BladeSystem Onboard Administrator User Guide version 4.50
BladeSystem c3000 nclosure (whitepaper)
BladeSystem c7000 enclosure technologies (whitepaper)</td><td>http://www.hpe.com/info/blades_enclosures-docs
In the main page, click HP BladeSystem c-Class Enclosures→HP BladeSystem c3000 Enclosures or HP BladeSystem c7000 Enclosures.</td></tr>
<tr><td>Virtual Partitions documentation</td><td>http://www.hpe.com/info/hpux-vpars-docs</td></tr>
<tr><td>HP-UX 11i v3 documentation</td><td>http://www.hpe.com/info/hpux-core-docs
In the main page, click HP-UX 11i v3.</td></tr>
<tr><td>Run Oracle OLTP workloads in HP-UX vPars and Integrity VM v6.2 - Technical white paper</td><td>http://www.hpe.com/info/hpux-hpvm-docs</td></tr>
</table>

18 Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

Support policy for HP-UX

For more information about support policy for HP-UX, see **HP-UX support policy**.

A Troubleshooting

Online vPar Migration

Online vPar migration is not supported for guest

If the following error is observed, online vPar migration may fail:

```
hpvminfo: Running on an HPVM host.

# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar1                    1 VP   HPUX    On (OS)        6     1     3 8192 MB

# hpvmmigrate -p 1 -o -h host2
hpvmmigrate: Connected to target VSP using 'host2'
hpvmmigrate: ERROR (vpar1): vpar1 OE does not support vpar online
migration.
hpvmmigrate: ERROR (vpar1): Online vpar migration is not supported for
guest (vpar1).
```

If the vParOGMEnh bundle is not installed in the vPar, following error is displayed:

```
# hpvminfo
hpvminfo: Running inside an HPVM vPar.

# swlist vParOGMEnh
# Initializing...
# Contacting target "vpar1"...
ERROR: Software "vParOGMEnh" was not found on host "vpar1:/".
```

The vParOGMEnh bundle needs to be installed in the vPar for enabling the online vPar migration feature.

NOTE: For more information on configuring vPar for enabling online vPar migration feature, see *HP-UX vPars and Integrity VM v6.4 Release Notes*.

vPar or VM is not fully running

If the following error appears, online vPar migration may fail:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar1                    1 VP   HPUX    On (OS)        6     1     3 8192 MB

# hpvmmigrate -p 1 -o -h host1 -q
hpvmmigrate: ERROR (vpar1): The vPar or VM is not fully running.
```

Even though the vPar is shown in On (OS) state, online vPar migration fails with the message:

```
The vPar or VM is not fully running
```

It indicates the guest user space daemon, `hpvmgud`, is not yet running inside the vPar. The guest user space daemon is required to run the pre- and/or post migration scripts inside the vPar. Once the rc (1M) scripts launch the guest user space daemon, migration proceeds.

Online vPar migration aborts if free vPar memory is less than 30%

If the amount of free memory available in a vPar is less than 30%, then online vPar migration may abort and display the following message:

```
# hpvminfo && hostname
hpvminfo: Running on an HPVM host.
host2

# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar1                    1 VP   HPUX    On (OS)        6     1     3 8192 MB

# hpvmmigrate -p 1 -o -h host1 -q
hpvmmigrate: Starting vPar/VM 'vpar1' on target VSP host 'host1'
hpvmmigrate: Init phase (step 30) - progress 80%
hpvmmigrate: ERROR (vpar1): Remote message: Target vPar or VM exited.
Status 2.
hpvmmigrate: ERROR (vpar1): Remote message: Unable to start vPar/VM
on target.
hpvmmigrate: ERROR (vpar1): Migration was aborted by vPar. Please refer
vPar's syslog.
```

The `hpvmmigrate` command indicates that the vPar aborted migration due to some reason. It also advises the user to check the file `/var/adm/syslog/syslog.log` of vPar. The vPar syslog file shows the following information pertaining to migration:

```
vi /var/adm/syslog/syslog.log
...
vpar1 vmunix: vm_mig_validate_ken_handler:Free memory left
at source is less than 30 percent, can not migrate
vpar1 vmunix: vm_mig_validate_ken_handler: Backout initiated
```

You can use `glance(1)` to see remaining memory before initiating online vPar migration. If the amount of free memory is less than 30%, then you can add more memory to the vPar to resolve the issue.

NOTE: Online migration of memory is not supported on an online migrated vPar. You must configure sufficient memory to avoid the requirement of having 30% free memory, and ensure a successful online vPar migration.

Online vPar migration aborts due to insufficient contiguous memory

Due to insufficient contiguous memory, online vPar migration may abort and display the following message:

```
# hpvminfo && hostname
hpvminfo: Running on an HPVM host.
host2

# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vpar1                    1 VP   HPUX On (OS)           6     1     3 8192 MB

# hpvmmigrate -p 1 -o -h host1 -q
hpvmmigrate: Starting vPar/VM 'vpar1' on target VSP host 'host1'
hpvmmigrate: Init phase (step 30) - progress 80%
hpvmmigrate: ERROR (vpar1): Remote message: Target vPar or VM exited.
Status 2.
hpvmmigrate: ERROR (vpar1): Remote message: Unable to start vPar/VM
on target.
hpvmmigrate: ERROR (vpar1): Migration was aborted by vPar. Please refer
vPar's syslog.
```

The `hpvmmigrate` command indicates that the vPar aborted migration due to some reason. It also advises the user to check the file `/var/adm/syslog/syslog.log` of vPar. The vPar syslog file shows the following information pertaining to migration:

```
# vi /var/adm/syslog/syslog.log
...
Jun 7 11:28:33 vpar1 vmunix: Not enough contiguous free memory (base)
to continue Migration
Jun 7 11:28:33 vpar1 vmunix: Migration Aborted. Required 9 granules
(64MB each) of memory,
Jun 7 11:28:33 vpar1 vmunix: Obtained 8 granules only. Please shutdown
some applications
Jun 7 11:28:33 vpar1 vmunix: and try again
Jun 7 11:28:33 vpar1 vmunix: vm_mig_memcopy_handler: Back-out initiated
Jun 7 11:28:33 vpar1 vmunix: vm_mig_memcopy_handler: Backout initiated
```

Unable to get source vPar topology on target VSP

Online vPar migration may fail with the following message:

```
# hpvmmigrate -p 1 -o -h host1
hpvmmigrate: Connected to target VSP using 'host1'
hpvmmigrate: Starting vPar/VM 'vpar1' on target VSP host 'host1'
(C) Copyright 2000 - 2016 Hewlett-Packard Development Company, L.P.
addguest failed 'Not enough cpu, can't fit 2 pct entitlement'
Allocation of resources failed
Unable to get source vPar topology on target host.
To force migration, try with force_vpar_migration=enabled using
hpvmmodify
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/2f8dc300-5147-11e5-8b03
1cc1de40a7e8/vmm_config.next): Unable
to allocate vPar or VM resources
hpvmmigrate: Init phase (step 4) - progress 0%
hpvmmigrate: ERROR (vpar1): Remote message: Target vPar or VM exited.
Status 2.
hpvmmigrate: ERROR (vpar1): Remote message: Unable to start vPar/VM
on target.
hpvmmigrate: ERROR (vpar1): Migration was aborted because of connection
failure 5 error 0.
```

In this case, online vPar migration failed because the resource agent cannot allocate identical resources to vPar on the target VSP as it had allocated to it on the source `hpvmmodify (1M)` VSP. The `hpvmmigrate` advises the user to retry the migration by enabling the `force_vpar_migration` option using the `hpvmmodify (1M)` command. This option `force_vpar_migration` is disabled by default. It can be enabled as follows:

```
# hpvmmodify –p 1 –x force_vpar_migration=enabled
```

This option can be changed dynamically and once enabled, migration proceeds successfully.

Migration was aborted by timeout in frozen phase

If a vPar is running load, online vPar migration may fail with the following message:

```
# hpvmmigrate -p 1 -o -h targethost –q
hpvmmigrate: Connected to target VSP using 'targethost'
hpvmmigrate: Init phase completed successfully.
hpvmmigrate: Copy phase completed successfully.
hpvmmigrate: I/O quiesce phase completed successfully.
hpvmmigrate: Frozen phase (step 2) - progress 0%
    Target: transfer aborted by source
    Specific Core OLD thread being terminated
    Received SHUTDOWN for this VPAR guest.
hpvmmigrate: ERROR (vpar1): Remote message: Target vPar or VM exited.
Status 2.
hpvmmigrate: ERROR (vpar1): Remote message: Unable to start vPar/VM on
target.
hpvmmigrate: ERROR (vpar1): Migration was aborted by timeout in Frozen
phase step 2.
```

The source vPar's log file can be inspected to find out the reason for the timeout. Following is the relevant part of the log file:

```
# vi /var/opt/hpvm/guests/vpar1/log
...
Source: frozen phase timeout abort - insufficient time available to transfer memory
Source: estimated 210.364 seconds required - 200 available
Source: need to transfer 34240 MB - Copy phase rate 180.298 MB/s,
      Freeze zero copy =365494270 ticks (457 ms)
      Checksum(2) validation estimation =5593000000 ticks (7000 ms)
      Resume time estimation =10387000000 ticks (13000 ms)
```

The default timeout for frozen phase of online vPar migration is 200 seconds. The log file indicates that 200 seconds is insufficient for this particular migration. To resolve the issue, the timeout value of the frozen phase can be increased appropriately with the following command:

```
# hpvmmodify -p 1 -x migrate_frozen_phase_timeout=300
```

In this example, the timeout value of the frozen phase is increased to 300 seconds. The new timeout value must be specified in seconds.

Another operation in progress, please retry the operation

When a vPar is running load, online vPar migration might fail with the following message:

```
# hpvmmigrate -p 1 -o -h host2
hpvmmigrate: Connected to target VSP using 'host2'
hpvmmigrate: ERROR (vpar1): Another operation in progress,
please retry the operation.
```

The previous message is displayed if an online addition or deletion operation is in progress. It can also occur if some migration cleanup activity is pending due to a previous online vPar migration termination attempt.

It is resolved by retrying the migration after some time.

vpar_guest_ogm_enable is not set for vpar1

Migration may fail with the following error message:

```
# hpvmmigrate -p 1 -o -h host1
hpvmmigrate: Connected to target VSP using 'host1'
hpvmmigrate: ERROR (vpar1): vpar_guest_ogm_enable tunable is not set for vpar1.
hpvmmigrate: ERROR (vpar1): Online vpar migration is not supported for guest (vpar1).
```

In this scenario, an online vPar migration operation is failed because the `vpar_guest_ogm_enable` tunable was not set. But inside the vPar, the `vpar_guest_ogm_enable` tunable is already enabled.

```
# hpvminfo && hostname
hpvminfo: Running inside an HPVM vPar.
vpar1

# kctune vpar_guest_ogm_enable
Tunable Value Expression
vpar_guest_ogm_enable 1 1
```

This problem occurs because the vPar is disabled for online migration. To resolve the issue, enable online migration by issuing the following command:

```
# hpvmmodify -p 1 -x online_migration=enabled
```

Now, restart the vPar to enable vPar for online migration.

Online addition or deletion of a resource may fail on a rebooted guest

Online addition or deletion of a resource (CPU, memory or IO device) may fail on a vPar that is rebooted immediately after a successful online vPar migration operation.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
```

```
vpar1                    1 VP   HPUX    On (OS)        2     1     4 4096 MB

# vparmodify -p 1 -a cpu::1
vparmodify: Modification of vPar or VM is disabled.
vparmodify: Unable to modify the vPar.

# vparmodify -p 1 -a mem::1G
vparmodify: Modification of vPar or VM is disabled.
vparmodify: Unable to modify the vPar.

# hpvmmodify -p 1 -a network:avio_lan::vswitch:localnet
hpvmmodify: Modification of vPar or VM is disabled.
hpvmmodify: Unable to modify the vPar or VM.

# hpvmstatus -p 1 -V
...
Runnable status : Not runnable
Not runnable set by : Migrate
Not runnable reason : vPar/VM is being migrated to this VSP.
Modify status : Not modify
Not modify set by : Migrate
Not modify reason : vPar/VM is being migrated to this VSP.
Visible status : Not visible
Not visible set by : Migrate
Not visible reason : vPar/VM is being migrated to this VSP.
```

This problem happens because the vPar was rebooted immediately after an online vPar migration operation was completed successfully. During an online vPar migration operation, once the vPar is marked to `On(OS)` state on the target VSP, post migration scripts are run in the vPar. The post migration scripts restart diagnostic daemons, and also issue an ioscan (1M) operation in the vPar. If many IO devices are configured in the vPar, then ioscan (1M) may take more time to complete. If the vPar is rebooted before the ioscan (1M) operation completes, then this problem is observed.

A vPar may be marked as Off (NR) if it is shut-down immediately after a successful online migration

A vPar may be marked in Off (NR) state if it was shut down immediately after a successful vPar migration. During an online vPar migration operation, once the vPar is marked to On (OS) state on the target VSP, post migration scripts are run in the vPar. The post migration scripts restart diagnostic daemons and also issue an ioscan (1M) operation in the vPar. If many IO devices are configured in the vPar, then ioscan (1M) may take some time to complete. If the vPar is shut down before the ioscan (1M) operation completes, then the vPar will be marked as Off (NR). Consequently, the vPar is marked as Off (NR) on both the source and destination VSPs.

To resolve the issue, execute the following commands to mark the vPar as modifiable, visible, and runnable on either the source or target VSP:

```
# hpvmmodify -p 1 -x register_status=enabled

# hpvmstatus -p 1 -V | grep -e "Runnable" -e "Modify" -e "Visible"
Runnable status : Runnable
Modify status : Modify
Visible status : Visible
```

When an online migration operation is aborted, then guest state may not revert back to On (OS) state from the previous On (MGS) state

When an online guest migration is aborted in frozen phase, then the state of the guest may not revert to On (OS) state from the previous On (MGS) state. This happens under rare circumstances when the VSP controller daemon (hpvmctrld) fails to update the guest status.

To resolve this issue, the VSP controller daemon can be restarted on the source VSP.

```
# hpvmctrld –r
#
```

vPar/VM has pending modifications and cannot be migrated

Under rare scenarios, an online migration operation may fail with the following messages:

```
# hpvmmigrate -p1 -o -h host1
hpvmmigrate: Connected to target VSP using 'host1'
hpvmmigrate: ERROR (vpar1): vPar/VM 'vpar1' has pending modifications
and cannot be migrated online.

# hpvmstatus -p1 -A
Changed items
        Current = Runnable status : Runnable
   Next Start = Runnable status : Not runnable

        Current = Modify status : Modify
   Next Start = Not runnable set by : Migrate

        Current = Visible status : Visible
   Next Start = Not runnable reason : vPar/VM is being migrated to this VSP.

Items only in the next start configuration
      Modify status : Not modify
      Not modify set by : Migrate
      Not modify reason : vPar/VM is being migrated to this VSP.
      Visible status : Not visible
      Not visible set by : Migrate
      Not visible reason : vPar/VM is being migrated to this VSP.
```

This problem occurs because the previous online migration operation has left a stale copy of the `vmm_conifg.next configuration file` configuration file.

To resolve this issue, the stale copy of the `vmm_config.next` file needs to be removed. Issue the following command to remove the stale copy of the `vmm_config.next` file:

```
# hpvmmodiy –p 1 –U
#
```

In addition, the HPVM APIs communicate with the local VSP controller daemon (hpvmctrld) through IPC message queues and shared memory. The shared memory region needs to be refreshed by restarting the VSP controller daemon. To restart the VSP controller daemon, issue the following command:

```
# hpvmctrld –r
#
```

POST/REVERT migration operation failed

During online vPar migration, the following messages may be displayed on vPar log file:

```
# vi /var/opt/hpvm/guests/<guest_name>log
(4) POST migration operation failed(10).Please refer Admin guide
for corrective actions.

OR

# vi /var/opt/hpvm/guests/<guest_name>log
...
(4) REVERT migration operation failed(10).Please refer Admin guide
for corrective actions.
```

The above messages indicate that migration scripts have failed to execute for some reason. The messages advise the user to take corrective action. If POST migration operation is failed, then run the following commands in the vPar:

```
vpar1# /opt/hpvm/bin/migrate.d/M000hpvmguest_run_ioscan post_migrate
vpar1# /opt/hpvm/bin/migrate.d/M101hpvmguest_diags post_migrate
```

If REVERT migration operation is failed, then run the following commands in the vPar:

```
vpar1# /opt/hpvm/bin/migrate.d/M000hpvmguest_run_ioscan revert_migrate
vpar1# /opt/hpvm/bin/migrate.d/M101hpvmguest_diags revert_migrate
```

Creating VMs

Configuration error on starting the VM

When you start the VM, the following message is displayed:

```
Configuration error: Device does not show up in guest.
```

If this is observed:

- Verify that the path name to the file-backing store is correct and that the physical storage device is mounted.
- Verify that the size of the physical storage device is divisible by 512 bytes (for a disk device) or 2048 (for a DVD device).
- Modify the VM to use correct file-backing store path name and size, using the `hpvmmodify` command.

Storage

Attachable storage devices

Storage devices are not seen in guest

Use the `ioscan` command and check that the devices are connected and claimed by VSP. Install any device special files for new devices, if required.

The following is an example of a claimed tape device:

```
# ioscan -m lun /dev/rtape/tape1_BEST
Class I  Lun H/W Path Driver     S/W State   H/W Type     Health Description
==============================================================================

Tape     1  64000/0xfa00/0x0        estape       CLAIMED DEVICE online    STK T9940B
0/1/1/1.0x500104f00048b29d.0x0
0/7/1/1.0x500104f00048b29e.0x0
/dev/rtape/tape1_BEST/dev/rtape/tape1_BESTn
/dev/rtape/tape1_BESTb/dev/rtape/tape1_BESTnb
```

The following is an example of an unclaimed media changer device:

```
# ioscan —fk

Class    I   H/W Path    Driver  S/W State   H/W Type  Description
======= ===  ========    ======  ==========  ========= ==============
ext_bus  6   0/2/1/0     c8xx    CLAIMED     INTERFACE SCSI C1010 Ultra0 Wide LVD A6828-60101
target  35   0/2/1/0.0   tgt     CLAIMED     DEVICE
unknown -1   0/2/1/0.0.0         UNCLAIMED   UNKNOWN   HP ThinStor AutoLdr
```

If the device is not seen, there is a hardware problem or AVIO ID conflict. See the documentation for the particular device to resolve this issue before proceeding.

If the device is seen but not claimed, this is a result of missing drivers in the VSP. Integrity VM does not require the drivers to be loaded on the VSP for the devices to be attached. The HP-UX tape (`stape` and `estape`) and changer (`schgr` and `eschgr`) drivers are not loaded by default unless those devices are connected at install time. To load the drivers, use the `kcmodule` command to statically load the drivers. To complete the installation, the VSP must be rebooted. Any guests that are running must be shut down before loading these drivers.

The following is an example of installing the tape driver:

```
# kcmodule stape=static
```

The following is an example of installing the media changer driver:

```
# kcmodule schgr=static
```

If you are loading the VSP drivers, the devices must show up in `ioscan` with device files, after the VSP reboot.

Commands that operate on attachable storage devices appear to hang

Accessing some attachable devices involve multiple system calls which altogether consume observable time before completing. Commands such as `hpvmcreate(1M)` and `hpvmmodify(1M)` that operate on such devices may appear to hang; such commands usually complete in about a minute. The following are the examples of such usage:

```
# hpvmcreate -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmcreate -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
```

Access errors on storage devices

The following are the access errors and suggestions for resolving the errors that are reported by storage devices on both VSP and within guests.

- VSP error messages

  The VSP's attempt to access a shared tape is denied when it is in use by any guests; a busy error is returned. The following example describes the behavior of `diskinfo` on a tape which is being used by a guest:

  ```
  # diskinfo /dev/rtape/tape1_BEST
  diskinfo: can't open /dev/rtape/tape1_BEST: Device busy
  ```

- Guest error messages
  - 11i v3 guest — access error on a shared attached device

    The attempt of a guest to access a shared tape is denied when it is in use by the VSP or other guests. Applications receive a busy error in such cases. The following example describes the behavior of `diskinfo` on a tape that is being used by another guest.

    ```
    # diskinfo /dev/rtape/tape1_BEST
    diskinfo: can't open /dev/rtape/tape1_BEST: Device busy
    ```

  - 11i v2 guest — access error on a shared attached device

    The attempt of a guest to access a shared tape is denied when it is in use by the VSP or other guests. Applications receive a no-device error in such cases. The following example describes the behavior of `diskinfo` on a tape that is being used by another guest.

    ```
    # diskinfo /dev/rmt/c7t0d0BEST
    diskinfo: can't open /dev/rmt/c7t0d0BEST: No such device or address
    ```

NPIV storage devices

NPIV devices are not visible from guest EFI shell after being successfully added to guest

The EFI functionality to enumerate FC (NPIV) LUNs is switched off by default. For instructions to turn it on, see “HPE AVIO Stor EFI Driver enumeration policy” (page 275).

Guests with large number of NPIV HBAs take a long time to boot

The EFI setting that enumerates the FC devices might have been switched ON to obtain a list of all NPIV devices at EFI shell. If the guests are configured with a large number of FC (NPIV) LUNs, enumeration of these devices at the EFI shell might require a substantial time. The option to enumerate FC LUNs at the EFI shell must be enabled only if required; it must (preferably) be disabled after the purpose is met.

Online migration of guests configured with NPIV HBAs fails; error messages indicate “data put failure” and “invalid target”

Online migration of a guest configured with NPIV HBAs fails with the following message:

```
Target: dynamic IO data put failure - status 4 tag 0 length 0 depth 0
```

And, the target VSP syslog contains an error message from the host virtual storage driver similar to the following:

```
HVSD: HPVM online migration error: invalid target id 0x207000c0ffda4ee1 under hba port 0x5001438002a30063 for
VM instance 1
```

This can be an indication that a target port that was visible from the source VSP is no longer visible from the target VSP. This might occur if zoning configuration on the FC fabric shared by the source and target VSP is incorrect. To ensure successful migration of guests with NPIV devices, Hewlett Packard Enterprise recommends that the SAN administrator uses WWN based zoning instead of Port based zoning.

This error can mean that a target port that was visible from the source VSP has failed and gone offline. To be able to migrate the guest online, the failed or unavailable target must be cleaned up from the guest prior to attempting a migration. This can be done by running a `rmsf -H` against the target path in the guest.

For the `rmsf` command to clean up all the stale target information, the FC drivers or FCoC drivers in the host must be March 2013 version or later. For more information about the list of dependencies, see *HP-UX vPars and Integrity VM Release Notes*.

NPIV LUNs not shown by default invocation of ioscan

By default, the `ioscan(1M)` command displays only devices that use legacy style device file format. NPIV LUNs use the agile device file format. The `-N` option to ioscan must be specified in order to display NPIV LUNs.

SCSI queue depth on legacy AVIO and NPIV devices

During high I/O load, tools like glance, when run inside a vPar or VM shows a very large value against the Qlen field. Qlen is an indication of number of I/Os that in queue waiting to be processed by the device. One way to reduce this is to tune the SCSI Queue Depth on the guest devices. This value is the maximum number of concurrent I/O requests that could be outstanding for a device and it must be based on the capability of the actual physical device to which the guest device is mapped to on the VSP. The SCSI queue depth can be set or viewed on a vPar or VM device using the `scsimgr` command, just the way it is set or viewed on a physical server.

For more information about tuning the SCSI queue depth for AVIO devices, see *Integrity VM Accelerated Virtual I/O Overview* at **http://www.hpe.com/info/hpux-hpvm-docs**.

AVIO storage devices

When vPar or VM guest is created with VxVM as the root volume manager and AVIO storage device(s) are presented to this guest, the online deletion (OLD) of such legacy AVIO backing stores may fail with the following error even if the devices are not in use within the vPar or the VM guest.

```
[Dynamic I/O Interface Details]
IO OLAD operation in progress     : none
IO OLAD current operation argument : none
IO OLAD last operation completed  : LUN Delete
IO OLAD last operation argument   :
Device type            : disk
Adapter type           : avio_stor
Ioscan format          : 0/0/0/1/0.0.0

Bus                    : 0
Device                 : 1
Function               : 0
```

```
Target                  : 0
Lun                     : 0
Physical Device         : /dev/rdisk/disk123
IO OLAD last operation status     : failed_guest
IO OLAD last operation error      : Device busy
```

The online delete operation fails with “Device busy” error as VxVM (running in the vPar or VM guest) would continuously access the device to monitor its status. VxVM must stop accessing the device before it can be safely removed from the guest.

Run the following command (in guest) inside the vPar or VM guest to stop VxVM from accessing the device prior to the device OLD operation.

```
# hpvmdevinfo
Virtual Machine Name    Device Type     Bus,Device,Target       Backing Store Type      Host Device Name
Virtual Machine Device Name
TestGuest           disk        [0,1,0]          disk           /dev/rdisk/disk123      /dev/rdisk/disk1
# vxdisk rm <disk_name>
For example: vxdisk rm disk1
```

NPIV devices with bandwidth entitlement

When vPar or VM guest has NPIV HBA configured with bandwidth entitlement on 16Gb port, and this card is OLR’ed and replaced with other card with older Firmware (less than v8.1.80), then NPIV HBAs with bandwidth entitlement will remain in disabled state, as the firmware on the card is old.

To get NPIV HBA back online, replace the card with the firmware (>= v8.1.80) that supports the bandwidth entitlement. Alternatively, upgrade the firmware on the card, this requires all the vFCs on the card to be deleted. After the upgrade, the vFCs must be added back again.

Networking

AVIO networking

Do not kill hpvmnetd

Do not use the `kill` command to remove the `hpvmnetd` process. The following error message indicates that the `hpvmnet` daemon has been killed:

```
hpvmnetd: Switch 0000564d4c414e31 already exists
```

If the `hpvmnetd` process is removed, vswitches do not work properly. To recover from this, run `hpvmnet -b` which restarts the vswiches.

AVIO LAN devices not claimed by guest with DOWN vswitch at boot time

In addition to running `ioscan`, it is necessary to re-run network startup scripts so that IP addresses can be configured on network interface cards (NICs). For example:

```
/sbin/rc2.d/S340net start
/sbin/rc2.d/S340net-ipv6 start
```

Redefining pNICs for HP-UX guests

Changing the hardware address of a vswitch has the same effect as moving a network adapter from one hardware slot to another on an Integrity system. Similar to other HP-UX systems, the guest file `/etc/rc.config.d/netconf` must be modified so that `INTERFACE_NAME[0]` reflects the new LAN PPA assigned by the HP-UX network driver on the first guest reboot after modification. At the first reboot, the LAN interfaces configuration fails, as follows:

```
Configure LAN interfaces ............................ FAIL*
```

When the guest is running, you can use the `nwmgr` command to identify the new LAN PPA and `netconf` command to modify the new LAN PPA. For example:

```
# nwmgr

Name/        Interface   Station     Sub-    Interface  Related
ClassInstance   State    Address     system   Type    Interface
```

```
=============  ========   ============   ========  =========  =========
lan3           UP         0x02636c6E3030  iexgbe    10GBASE-KR
```

In the preceding example, before the modification, the LAN PPA was `0`. The new LAN PPA on the first boot after the modification is `3`. To resolve this, you must bring the guest network down, then you must change the `INTERFACE_NAME[0]` from `lan0` to `lan3`. You can then use `/sbin/rc2.d/S340net` to restart the guest network. For example:

```
# /sbin/rc2.d/S340net stop
# ch_rc -a -p "INTERFACE_NAME[0] = "lan3"
# /sbin/rc2.d/S340net start
```

The guest network begins to function.

Problems with VLANs

When VLANs are configured on the vswitch, the partitioned LAN must have its own set of network servers to service requests on the VLAN.

If guests start slowly or hang during starting, determine whether the guest network interface is on a VLAN, and whether the appropriate network services (such as DNS) are set up and available on the VLAN. You might need to disable some of these network services on the guest before booting up the guest on a VLAN.

When VLANs are configured on the vswitch and the guests are required to communicate over a VLAN with a remote node outside the VSP, you might need to set up the physical network appropriately for the VLAN. For information about configuring VLANs on the switches, see the product documentation for the physical network adapters.

If TCP/UDP applications have trouble communicating between a guest and the local VSP over a VLAN, it is possible that the host interface for the vswitch is checksum-offload capable. To resolve the problem, identify the interface used by the vswitch and run the following command on the VSP to disable the CKO feature, where 4 is the VSP interface as shown in the `hpvmnet` command output.

```
# nwmgr -s -A tx_cko=off -c lan4
lan4 current values:
Transmit Checksum Offload=Off
```

`Checksum offloading (CKO)` is not supported. On most of the physical interfaces that are not of 10 Gigabyte type, CKO is turned off by default. Consult your interface card documentation for details.

Turning on CKO can cause host-to-guest connections as well as guest-to-host communication over a VLAN to fail. If you are receiving failures with host-to-guest connections or guest-to-host communication using a VLAN, ensure that the CKO is turned off in the host interface driver. If that does not fix the problem, reboot the vswitch.

To turn off the CKO on the VSP, identify the PPA of the network interface for the vswitch using the `hpvmnet` command and use `nwmgr` command with -A tx_Cko on the PPA. For example:

```
# hpvmnet

Name       Number State   Mode      PPA     MAC Address    IP Address
========   =====  ======= ========= ======= ============   ==========
localnet   21     Up      Shared    N/A     N/A
vmlan0     22     Up      Shared    lan0    0x00306ea72c0d 15.13.114.205
vmlan4     23     Up      Shared    lan4    0x00127942fce3 192.1.2.205
vmlan900   24     Up      Shared    lan900  0x00306e39815a 192.1.4.205
```

VLAN-Backed vswitches

To enable the VLAN-backed vswitch (VBVsw) feature, PHNE_40215 or a superseding patch is required to be installed on the VSP. This patch is available as an individual patch or as part of "FEATURE11i" bundle. To verify that the patch is installed, enter the following:

```
# swlist -l product |  grep PHNE_40215
PHNE_40215            1.0            LAN cumulative patch
```

The `dlpi_max_ub_promisc` kernel tunable must be set to when using a VBVsw type vswitch. Otherwise, attempting to boot the vswitch fails with the following error message from the `hpvmnet` command:

```
# hpvmnet -b -S vs5000
hpvmnetd: setup_downlink: promisc failed, recv_ack:
promisc_phys: UNIX error - Device busy, errno 5
```

To set the kernel tunable, enter the following:

```
# kctune dlpi_max_ub_promisc=16
```

Miscellaneous AVIO Networking problems

The following are the other AVIO networking problems:

- If you modify the MAC address of an interface in the guest, the `hpvmstatus` command in the VSP does not display the current MAC address correctly. There is no fix or workaround for this problem at this time.
- Just as with physical devices on a network, for communication to occur uninterrupted between all stations on a LAN segment, the MTUs of all the systems on the LAN segment or VLAN must match, whether they are physical systems or guests. The VSP does not check for MTU mismatches for its guests.
- The `lanadmin` card specific options that are supported on igssn on the guest are:
  - `-x:speed,fctrl,cko,type,card_info,stats drv,vmtu,and drv_pr.`
  - `-X:drv_pr_on,drv_pr_off,and stats clear`
- Inconsistent CKO/TSO settings

  Modifying the CKO/TSO settings of an interface on the VSP must be performed with caution. The CKO and TSO settings of backing interface on the card must be either enabled or disabled, having CKO enabled and TSO disabled state can cause network traffic of guest to stall.

  When CKO and TSO are disabled on NIC, the vswitch associated with NIC must be started with the following syntax:

  ```
  hpvmnet –x disable_ckotso=1 -b – S <switchname>
  ```
- MAC Address changes during Online Guest Migration must be avoided.

  As explained in Section (page 128), the vswitch must be restarted when there is a change in MAC address of the backing interface, this ensures successful operation of the Online Guest Migration.
- Co-locating the Ignite Server on a VSP server, is not a suggested configuration.

  Attempting to install the Guest Operating system from such an Ignite Server can result in the guest not booting. This behavior is observed with Checksum offload (CKO) enabled cards. Checksum Offload (CKO) is generally enabled by default on 10G backing NIC interfaces to enable driving higher LAN throughput. However, this causes EFI AVIO-LAN driver in the guest to fail to load the guest image.

  This problem can manifest with two different symptoms:

  Symptom A: When lanboot’s dbprofile feature is used and booting is not successful, the following methods can be used to workaround the issue.

  Method 1: Disable the CKO/TSO on the vswitch.

  ```
  hpvmnet -x disable_ckotso=1 -h -S <switchname>
  hpvmnet –x disable_ckotso=1 -b -S <switchname>
  ```

  This method disables the CKO/TSO on the backing physical NIC. The setting must be reverted back after the installation is over.

  Method 2: On VSP the tftpd daemon must be started with block size of 512.

```
tftpd -r 512
```

Symptom B: PXE-E18: Timeout. Server did not respond.

Method 1: Use Dbprofile and disable CKO/TSO on the vswitch

Method 2: Running instl_bootd instead of bootpd on the VSP. The `/etc/inetd.conf` entry for bootps must be modified to "`bootps dgram udp wait root /opt/ignite/lbin/instl_bootd instl_bootd`"

Troubleshooting DIO

If you are unable to add a DIO function or device to the DIO pool that is not in use by the VSP or already in the DIO pool, check the CRA log file `/var/adm/cra.log`. When `hpvmhwmgmt -p dio –a` … is executed, a Critical Resources Analysis (CRA) Report is generated and might provide clues as to why the function or device cannot be added to the pool. For example, Serviceguard might own the interface:

```
DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
0/0/0/4/0/0/0 (lan2)
0/0/0/4/0/0/1 (lan3)


DATA CRITICAL RESULTS:
Interface lan2: COMMAND cmnetd  PID 2907
Interface lan2: COMMAND cmnetd  PID 2907
Service-Guard(SG) Usage:
The interfaces listed below are being used by SG:
 lan2
```

Use the `hpvmdevinfo` command to display the hardware device mapping between vPar or VM and the VSP. You can run this command on the VSP or the vPar or VM:

VSP:

```
# hpvmdevinfo

Virtual       Device  Bus,Device,Target   Backing Store  Host Device Name  Virtual Machine
Machine Name  Type                         Type           Name                Device Name
============  ======  ===================  =============  ================    ================
vpar1        disk    [0,0,0]              disk          /dev/rdisk/disk13  /dev/rdisk/disk3
vpar1        disk    [0,0,2]              disk          /dev/rdisk/disk21  /dev/rdisk/disk5
vpar1        hba     [0,5]                npiv          /dev/fcd0          /dev/gvsd2
vpar1        lan     [0,6,0x7E06F5393261]  hwpath        0/0/0/4/0/0/0      0/0/0/6/0 (lan3)
```

vPar or VM:

```
# hpvmdevinfo

Device
Type    Bus,Device,Target  Backing Store Type   Host Device Name     Virtual Machine Device Name
=====   =================  =================    ================     ===========================
disk    [0,0,0]            disk                 /dev/rdisk/disk13    /dev/rdisk/disk3
disk    [0,0,2]            disk                 /dev/rdisk/disk21    /dev/rdisk/disk5
hba     [0,5]              npiv                 /dev/fcd0            /dev/gvsd2
lan     [0,6]              hwpath               0/0/0/4/0/0/0        0/0/0/6/0 (lan3)
```

VSP (Virtualization Services Platform)

CPU or memory info in `machinfo` output on VSP could be confusing

The `machinfo` command displays system information from the HP-UX view of the system configuration. The `machinfo` command might show different values based on when the command is executed. If executed on the VSP after installing HP-UX and before installing the Integrity VM software, `machinfo` shows all the sockets and logical processors. The logical processor count represents cores if the kernel tunable `lcpu_attr` value is 0 and threads when `lcpu_attr` value is 1. You can obtain the value of `lcpu_attr` by using the `kctune` command.

Note that `lcpu_attr` is set to zero in the VSP by default for optimal VSP performance, and so, the logical processor count is always the CPU core count.

After the Integrity VM software is installed, the logical processor count of `machinfo` represents the number of VSP logical processors and the logical processors in the vPar/VM pool, but not yet activated in a vPar.

When a vPar is started, the logical processors in the vPar/VM pool assigned to the vPar are deallocated from the VSP and the `machinfo` output in the VSP will reflect that reduction in logical processor count.

When a vPar is stopped, the processor count shown in the VSP `machinfo` output will increase by the number of CPUs assigned to the vPar.

The memory value displayed by the `machinfo` command shows the amount of memory that was available to HP-UX when booted on the VSP. This memory value includes memory that is allocated to the vPars and the memory used by the VSP. Unlike the logical processor count, the memory amount does not change with the installation of the Integrity VM software.

As workaround, use the `vparhwmgmt —p cpu —l` command to view the number of processor cores that are allocated to the VSP and to the vPar pool.

Performance

CPU intensive applications may not be responsive when the VSP is servicing high I/O load for guests

Applications like SMH (which needs significant CPU bandwidth) are not likely to be very responsive when the VSP cores are already under heavy load servicing vPar or VM I/O requests. Hewlett Packard Enterprise recommends that you increase the number of VSP CPU-cores under such circumstances.

Integrity VM and vPar CLI commands experience poor performance when there are numerous devices on the VSP

The commands like `vparmodify`, `hpvmmodify`, `hpvmcreate`, and `hpvmclone`, (commands used to modify the vPar or VM configuration), experience slow performance when there are numerous devices available on the VSP, or configured in the vPar and/or VM configurations. When you have a large number of devices, it is more than likely that the majority of those devices are storage devices. If storage devices are being exposed to the VSP from a SAN and then individually mapped to vPar/VM configurations, alternatively, you can map SAN-based LUNs directly to the VMs or vPars using NPIV. Replacing individually mapped SAN-based LUNs with one or more virtual HBAs using NPIV ports, reduces the number of devices that need to be managed, and thus improves the CLI performance.

I/Os take long to complete under heavy I/O conditions on vPars or VMs with large NPIV LUN configuration

In a large LUN configuration, spread NPIV HBAs across multiple physical HBA ports at the VSP level.

CLI

`hpvmhwmgmt(1M)` reports DIO resources are in use by VSP

If `hpvmhwmgmt -p dio -a hwpath` fails due to a resource being in use by the VSP, check the `/var/adm/cra.log` file for additional information on the resources in use by the VSP. The following example shows the type of error you might see in this case:

```
system# hpvmhwmgmt -p dio -a 0/0/0/4/0/0/0
hpvmhwmgmt: ERROR - Resources for: '0/0/0/4/0/0/0'
are in use by your host. hpvmhwmgmt: ERROR - could not reserve hwpath: '0/0/0/4/0/0/0'
```

```
for DIO. hpvmhwmgmt: ERROR - failed to reserve and store hw path: '0/0/0/4/0/0/0'
for DIO. hpvmhwmgmt: Unable to manage dio pool resource.
```

`hpvmmodify(1M)` may fail with the message “intent failed Can’t get the resource maxima”

The `hpvmodify` command invoked on a running VM might fail when it should succeed. When failing, the following error message is displayed:

```
resource
intent failed 'Can't get the resource maxima.' vPar/VM vm_name configuration problems:     Error 1: Internal
error -1 when attempting to use the ragent 'intent' interface
hpvmmodify: Unable to modify the vPar or VM.
```

This failure might happen only when the same processor is used by several VMs.

Miscellaneous

While booting a vPar or VM guest the message “WARNING: VCPU0 not scheduled” is displayed

In v6.2, messages similar to the following are occasionally seen during the initial boot of a vPar or VM:

```
WARNING: VCPU0 not scheduled for NNNNN ms" messages in hpvm_mon_log
```

They can occur during the early portion of booting the vPar/VM before HP-UX is launched into the vPar/VM, when either the EFI layer or the boot loader is running.

You can safely ignore these messages.

When a vPar is terminated by a `TC` command from its console, a corresponding vm.core is not always generated on the VSP

When a vPar is started up, it begins execution as a special application program. A `TC` command issued during early stage of starting up produces a vm.core on the VSP.

After early initialization, control is passed to boot stage and the vPar takes responsibility for its resources. After this stage, a `TC` command will not produce a vm.core on the VSP. Relevant state information is captured in the crash dump generated by the HP-UX OS in the vPar, as part of handling the `TC` command.

Note that HP-UX crash dump configuration must be done on the vPar to ensure that the dump is captured.

B Reporting problems with vPars and Integrity VM

You can report vPars and Integrity VM defects through your support channel. Follow these instructions to collect data to submit with your problem report.

1. Run the `hpvmcollect` command on the VSP to gather information about the guest before modifying any guest. Preserve the state of the VSP and the vPar and VM guest to best match the environment when the VSP failed.

   If multiple guests are running, run the `hpvmcollect` command for guest that was running at the time.
2. After the `hpvmcollect` archive is stored on the VSP, reboot the vPar and VM guest that caused the VSP to crash.
3. Run the `hpvmcollect` command on the guest again. Include this information in the `hpvmcollect` archive from the VSP.
4. Report the information through your support channel.

This chapter describes how to use the `hpvmcollect` command and how to investigate vPars and Integrity VM log files for information, including the following topics:

- "Collecting vPars and Integrity VM data"
- "Managing the size of the VMM driver log file"

Collecting vPars and Integrity VM data

You can use the `hpvmcollect` command on the VSP or on the vPar and VM to collect information that is useful in analyzing system problems. The options available for the `hpvmcollect` command on the VSP are different from those available on vPars/VMs. For information about using the `hpvmcollect` command, see one of the following sections:

- Using the `hpvmcollect` command the VSP, see "Using the hpvmcollect command on the VSP" (page 301).
- Using the `hpvmcollect` command on vPars/VMs, see "Using the `hpvmcollect` command on vPars or VMs" (page 304).

Using the hpvmcollect command on the VSP

Table 42 (page 301) describes the options to the `hpvmcollect` command on the VSP:

Table 42 Options to the `hpvmcollect` command on the VSP

Option	Description
`-P` *`vm-name`*	Specifies the vPar and VM guest name, where *`vm-name`* is the name of the vPar or VM.
`-p` *`vm-number`*	Specifies the vPar and VM guest number, where *`vm-number`* is the number of the vPar orVM.
`-s` *`host`*	Specifies a VSP name to receive the archive, which is copied using the `scp` command. Verify that you can log in to the VSP without a password.
`-n` *`crash-dump`*	Specifies the number of crash dumps to copy to the archive. By default, the `hpvmcollect` command copies the latest crash dump directory (based on the bounds file). This option can be used only with the `-c` option.
`-d` *`dir`*	Specifies a target directory in which to create the `hpvmcollect_archive` directory.
`-b` *`report-number`*	Specifies the archive name with the specified label. If an archive with the same name exists, it is renamed by appending a time stamp to the original name before the new archive is created.

Table 42 Options to the `hpvmcollect` command on the VSP *(continued)*

Option	Description
`-c`	Includes the latest crash dump directory in the archive. This option is used if the guest or the VSP fails or hangs.
`-f`	Forces an archive to be overwritten, if it exists, rather than renamed with an appended time stamp.
`-h`	Displays the help message for the `hpvmcollect` command.
`-l`	Leaves the collected information in a directory rather than in an archive file. The directory name follows the same naming convention as the archive name.
`-g`	Deletes old guest memory dump data as part of data collection.
`-a`	Selects all vPars/VMs on the VSP for inclusion in the collection. Valid only on the VSP.
`-r` *`directory`*	Specifies a remote target directory in which to store the collected archive, overriding the default of/crashes.Valid on both the VSP and the vPar and VM guest. The `-r` option is valid only with the `-s` option.

If the VSP hangs, generate a crash dump using the `TC` command on the VSP console. When the VSP crashes, it tries to dump a predefined set of memory pages into the crash dump area, including those that belong to Integrity VM. This is crucial to collecting a successful crash dump to analyze vPars and Integrity VM problems.

The `hpvmcollect` command is a shell script that can be run on either the VSP or the vPar and VM guest to gather system information, log files, Integrity VM logs, and configuration files for later analysis.

Because the `hpvmcollect` command collects generic vPars and Integrity VM and HP-UX operating system and system information, it might not collect all the information needed to analyze the source of the problem. Make sure that all the relevant information is included in the collection. For example, if the vPar and VM guest is running an Oracle® application, include the Oracle application log files and configuration.

By default, the `hpvmcollect` command creates a directory called `hpvmcollect_archive` in your current directory, and copies and collects all the vPars and Integrity VM and VSP information. For example, to gather information for a VM named `host1` on the VSP, enter the following command:

```
# hpvmcollect -P host1
```

This command creates a directory called `hpvmcollect_archive` in your current directory (if it does not already exist) and then collects information about the VSP crash dump. The information is then put into a `tar` file format (if there is a crash dump) or `tar.gz` file format (if there is no crash dump). Do not modify the guest configuration before running the `hpvmcollect` command.

If you do not want to archive the collection into `tar.gz` but simply want to examine the contents of the collection, use the `-l` option to leave the contents as they are.

If the VSP failed, use the `-c` option to collect crash dump files as well. Because the `-c` option collects the latest crash dump, use the `-n` option to specify a crash dump number.

Use the `-d` option to specify a different directory in which to store the `hpvmcollect_archive`.

For example, to collect information about `host1`, enter the following command:

```
# hpvmcollect -c -n 21 -d /tmp/hpvm_collect_archive -P host1
```

This command collects information about the guest called `host1` using crash dump number 21. The final archive is under `/tmp/hpvm_collect_archive` directory. The following is an example of `hpvmcollect` output on the VSP:

```
# hpvmcollect -P host1

  HPVSP crash/log collection tool version B.06.10.05
```

```
Gathering info for post-mortem analysis of guest 'host1' on host

Collecting I/O configuration info .................................. OK
Collecting filesystem info ......................................... OK
Collecting system info ............................................. OK
Collecting lan info ................................................ OK
Running lanshow .................................................... NO
Collecting installed sw info ....................................... OK
Collecting command logs ............................................ OK
Collecting messages from vmm ....................................... OK
Collecting lv info ................................................. N/A
Collecting vgdisplay info .......................................... OK
Collecting vxprint info ............................................ OK
Collecting disk info ............................................... N/A
Collecting passthru disk info ...................................... N/A
Collecting file backing store info ................................. N/A
Copying guest's log file ........................................... OK
Copying guest's tombstone file ..................................... N/A
Copying guest's console log file ................................... OK
Copying hpvm configuration ......................................... OK
Copying hpvm control script ........................................ OK
Copying guest's config file ........................................ OK
Getting status of the guest ........................................ OK
Getting detailed status of the guest ............................... OK
Getting guest's entitlement ........................................ OK
Copying guest's config file change log ............................. OK
Copying guest VM crash image ....................................... OK
Copying host vmunix image .......................................... OK
Copying host hpvmmkimage image ..................................... N/A
Copying VMM image .................................................. OK
Copying hpvmdvr image .............................................. OK
Copying hpvmntdvr image ............................................ OK
Copying NVRAM image ................................................ OK
Collecting IPMI logs ............................................... OK
Collecting crash dump .............................................. NO
Running crashinfo .................................................. NO
Collecting tombstone ............................................... NO
Collecting system message buffer ................................... OK
Collecting system syslogs .......................................... OK
Collecting measureware logs .......................................... OK

Finished with the collection

Tar archiving and compressing ...................................... TGZ
Remote copying the archive  ........................................ NO

The collection is
"/tmp/host1/hpvmcollect/hpvmcollect_archive/test_Jan.28.12_095249EDT.tar.gz"
```

If the command results in an error message like the following, you are out of disk space in the current directory or in the directory you specified with the `-d` option:

```
msgcnt 10 vxfs: mesg 001: vx_nospace - /dev/vg00/lvol5 file system full(1 block extent)
Tar: end of tape
Tar: to continue, enter device/file name when ready or null string to quit.
```

Use a file system with enough free space for the archive, especially when you use the `-c` option.

Additional data collected by the `hpvmcollect` command includes log files (guest, Integrity VM, and VSP) and VSP system information, including output from the `ioscan`, `lanscan`, and `swlist` commands. The `hpvmcollect` command also collects information about devices used by the guest. Output from the `crashinfo` and `lanshow` commands are included, if available.

The `hpvmcollect` command records device information in the following files:

```
config/
     host.diskinfo
     host.fsinfo
     host.ioscan
     host.laninfo
     host.sysinfo
```

Using the `hpvmcollect` command on vPars or VMs

To use the `hpvmcollect` command on the vPar and VM guest, you must first install the vPar and VM guest VirtualBase software on the vPar and VM guest (if it is not already installed) as described in "Installing VirtualBase on a vPar or VM Guest" (page 28).

Table 43 (page 304) lists the options that can be used with the `hpvmcollect` command on the guest.

Table 43 Options to the `hpvmcollect` command on guests

Option	Description
`-c`	Includes the latest crash dump directory in the archive. This option is used if the vPar and VM guest or the VSP fails or hangs.
`-f`	Forces an archive to be overwritten, if it exists, rather than renamed with an appended time stamp.
`-g`	Deletes old vPar and VM guest t memory dump data as part of data collection.
`-h`	Displays the help message for the `hpvmcollect` command.
`-l`	Leaves the collected information in a directory rather than in an archive file. The directory name follows the same naming convention as the archive name.
`-b` *report-number*	Specifies the archive name with the specified label. If an archive with the same name exists, it is renamed by appending a time stamp to the original name before the new archive is created.
`-d` *dir*	Specifies a target directory in which to create the `hpvmcollect_archive` directory.
`-n` *crash-dump*	Specifies the number of crash dumps to copy to the archive. By default, the `hpvmcollect` command copies the latest crash dump directory (based on the bounds file). This option can be used only with the `-c` option.
`-s` *host*	Specifies a VSP name to receive the archive, which is copied using the `scp` command. Verify that you can log in to the VSP without a password.

When you use the `hpvmcollect` command on the vPar and VM guest, do not specify the vPar and VM guest name. By default, the vPar and VM guest name is used as an archive directory name. You can use the `-d` option to specify the archive name. The following is an example of the `hpvmcollect` when it is run on the VM`host1`:

```
host1# hpvmcollect -c
HPVM guest crash/log collection tool version B.06.10.05
Gathering info for post-mortem analysis on guest (hostname 'host1')

Collecting I/O configuration info ................................... OK
Collecting filesystem info .......................................... OK
Collecting system info .............................................. OK
Collecting lan info ................................................. OK
Running lanshow ..................................................... NO
Collecting installed sw info ........................................ OK
Collecting crash dump 1  ............................................ OK
Running crashinfo ................................................... NO
Collecting tombstone ................................................ N/A
Collecting system message buffer .................................... OK
Collecting system syslogs ........................................... OK
Collecting measureware log .......................................... N/A

Finished with the collection

Tar archiving and compressing ....................................... TAR
Remote copying the archive  ......................................... NO
```

```
The collection is
"//hpvmcollect_archive/host1_Sep.29.05_122453PST.tar"
```

Recommendations for using `hpvmcollect` command

Hewlett Packard Enterprise recommends that `hpvmcollect` command should be always used with the options `-a` and `-c` together. If required, the `-n` option may be used to include multiple crash-dumps. Using these options will ensure that all system data is collected along with the related crash-dumps.

Managing the size of the VMM driver log file

The monitor log file (`/var/opt/hpvm/common/hpvm_mon_log`) is limited in size to 1024 KB. When the log file grows larger than this, it is copied to a new file (`hpvm_mon_log.$time`), and an empty one is created for the new log. To allow this log file to increase to 102400 KB, include the following line in the `/etc/rc.config.d/hpvmconf` file:

```
VMMLOGSIZE=102400
```

After you make this change to the `hpvmconf` file, enter the following commands to determine the PID for the monitor log daemon and to kill it:

```
# cat /var/run/hpvmmonlogd.pid5052# kill -HUP 5052
```

Using live dump

If a vPar crashes during online vPar migration, the VSP live dump patch will capture the vPar crash dump on the corresponding VSP. Capturing crash dumps are useful for analyzing system problems.

The VSP live dump patch can be installed from the same software distribution depot which was used to install online vPar migration software.

```
# swlist -s my.server.example.com:/host_depot/path
HostPatches.PHKL_44487
...
# HostPatches                      1.0 HostPatches Bundle
# HostPatches.PHKL_44487           1.0 livedump cumulative patch
  HostPatches.PHKL_44487.CORE2-KRN 1.0 OS-Core.CORE2-KRN
  HostPatches.PHKL_44487.CORE2-KRN 1.0 OS-Core.CORE2-KRN
```

You can install the live dump patch on both the source and target VSPs as shown below:

```
# swinstall -x autoreboot=true -s my.server.example.com:/depot/path
HostPatches.PHKL_44487
```

The following command can be used to verify if the live dump patch is installed on the VSP:

```
# swlist PHKL_44487
...
# PHKL_44487         1.0 livedump cumulative patch
PHKL_44487.CORE2-KRN 1.0 OS-Core.CORE2-KRN
```

You can remove live dump patch from the VSP by executing the following command:

```
# swremove -x autoreboot=true PHKL_44487
```

NOTE: Hewlett Packard Enterprise recommends installing the live dump patch on both source and target VSP's so that full vPar crash dumps are captured on their respective VSPs.

Use a file system with enough free space for capturing the full vPar crash dump.

In the following example, the `hpvmmigrate` command informs that vPar has crashed on the target VSP during online vPar migration. The vPar crash dump is collected on the target VSP. An online vPar migration is aborted due to a vPar panic. Following message advises the user to look in the crash dump directory of the target VSP for the live dump of vPar.

```
# hpvmmigrate -p 1 -o -h host2 -q
hpvmmigrate: Starting vPar/VM 'vpar1' on target VSP host 'host2'
hpvmmigrate: Init phase completed successfully.
hpvmmigrate: Copy phase completed successfully.
hpvmmigrate: I/O quiesce phase completed successfully.
hpvmmigrate: Frozen phase (step 22) - progress 98%
hpvmmigrate: ERROR (vpar1): Remote message: Target vPar or VM exited.
Status 2.
hpvmmigrate: ERROR (vpar1): Remote message: Unable to start vPar/VM on
target.
hpvmmigrate: ERROR (vpar1): Migration was aborted due to vPar panic on
the target VSP.
If livedump is enabled, then please refer the target VSP dump directory.
```

Full vPar crash dump is captured on the target VSP as follows:

```
# hpvminfo
hpvminfo: Running on an HPVM host.

# ls -lrt /var/adm/crash
total 4
-rw------- 1 root root 1 Mar 20 23:51 lbounds
drwx------ 2 root root 1024 Mar 21 00:01 gdump.1

# ls /var/adm/crash/gdump.1
INDEX        cifs        fclp_fcp      fcoc_vbus       gvsd
igelan       image.3.1   image.7.1     itxgbe          pciinfo
satadvd
btlan        ciss        fclp_vbus     fcq             hpvmguestdvr
igssn
image.4.1    image.8.1   lvmp          procsm          sysdev
c8xx         fcd         fcoc          fdd             iether
image.1.1    image.5.1   iocxgbe       mpt             rng          td
cdfs fclp    fcoc_fcp    gelan         iexgbe
image.2.1    image.6.1   iqxgbe        nadv            sasd
```

To get a full vPar crash dump, the vPar kernel executable file must also be copied to the crash dump directory as following:

```
# hpvminfo
hpvminfo: Running on an HPVM host.

# scp root@vpar1:/stand/current/vmunix /var/adm/crash/gdump.1/
Password:
vmunix

# ls /var/adm/crash/gdump.1
INDEX        cifs        fclp_fcp      fcoc_vbus       gvsd
igelan       image.3.1   image.7.1     itxgbe          pciinfo
satadvd
btlan        ciss        fclp_vbus     fcq             hpvmguestdvr
igssn
image.4.1    image.8.1   lvmp          procsm          sysdev
c8xx         fcd         fcoc          fdd             iether
image.1.1    image.5.1   iocxgbe       mpt             rng          td
cdfs fclp    fcoc_fcp    gelan         iexgbe
image.2.1    image.6.1   iqxgbe        nadv            sasd         vmun
ix
```

C Sample script for adding multiple devices

The following example provides a script that enables you to specify multiple storage devices at once for a guest.

```
#!/bin/ksh
# ---------------------------------------------------------------------------------------
# HP Integrity VM example script.
#
# SUMMARY:
#
#  Add disks to an Integrity VM (guest) in &apos;batch mode&apos; with hpvmmodify, using AVIO.
#
# SYNOPSIS
#
# ./thisscript [-a] -P guestname -f disklistfile [-N #] [-n #] [-t #] [-qT] [-F flags]
#    or
# ./thisscript -h | -H
#
# DESCRIPTION
#
#  This is an example script of how to automate adding many disks to an
#  Integrity VM guest using hpvmmodify, adding them as AVIO storage resources,
#  adding them in &apos;batch mode&apos;, that is, adding multiple disks with a single
#  call to hpvmmodify. When adding many disks, adding them in &apos;batch mode&apos;
#  provides a performance improvement over adding them one at a time (one
#  disk added per hpvmmodify call).
#
#  The disks to add are passed in as a filename that contains the list of
#  disks.  An example of how to generate this list is:
#
#   # hpvmhostgdev -u -l | grep /dev/rdisk > disklistfile
#
#  You may add all the disks in the disklistfile to a guest up to the supported
#  limit of 1024, or some lesser number of disks (see -N flag), starting with the
#  first disk in the disklistfile.
#
#  By default, this script adds 10 disks per hpvmmodify command.  You may
#  change the &apos;batch add&apos; number with the -n flag.  The value of -n may be
#  any value between 1 and 1024.
#
#  Also by default, this script does not specify the virtual bus, device,
#  target (b,d,t) triple in the hpvmmodify resource string.  So the default
#  limit of disks that may be added is 945.  [The algorithm used by hpvmmodify
#  default b,d,t assignment imposes this limit.]
#
#  To add 946 to 1024 disks to a guest, hpvmmodify requires that the virtual
#  bus, device, target (b,d,t) triple be specified in the resource string of
#  the additional disks over 945.  This script provides an option, -t, that
#  causes the script to calculate and use explicit b,d,t values for all of
#  the disks.  The valid values for the -t option are 0 and 15-127.  See
#  below for more details on this option.
#
#  This script only adds disks to guests when you specify the -a flag.  If you
#  omit the -a flag, this script will only print the messages that show what the
#  hpvmmodify commands will be.  You may suppress the sample hpvmmodify command
#  messages with the -q flag.
#
#  This script will time the hpvmmodify command with the timex command if
#  you specify the -T command.  [Note: timex output goes to stderr.]
#
#  You may also specify other hpvmmodify command arguments by using the -F
#  option.  The options you chose should be specified as though you were
#  typing them yourself on a commandline, using "-<flag>" or "-<flag> <value>",
#  including the leading hypen (&apos;-&apos;).  You must put the -F option value(s)
#  in double quotes for this script to include them in the hpvmmodify command..
#
#     WARNING: use the -F option at your own risk.  Also, you
#              must use -F option values that would work with
#              hpvmmodify if you were entering the command on
#              the commandline yourself.
#
# OPTIONS
#
#       -a                Add the disks (default is to only display what will be added)
#
#       -F "arg(s)"       Additional hpvmmodify options or flags (double quotes required)
#
#       -f disklistfile   File containing list of disks to add
#
#       -h                Print usage (help)
#       -H                Print usage (Help)
#
#       -N #              Number of devices to add from the disklistfile
#
#       -n #              Number of devices to add at one time (default: 10)
#
#       -P guestname      Name of Integrity VM (guest) to modify
#
#       -q                Quite mode - no display of hpvmmodify command that will run
```

```
#
#       -t targetmax      Max target value to use for -a disk:avio_stor:[b,d,targetmax]...
#                         Valid values:
#                           0       - special case: script will use full 0-127 range
#                          15...127 - script will use specified max
#                           1... 14 - not valid for this script, since 0-14 is
#                                     the normal default range for target values
#                                     if -t is not specified.
#
#       -T                Time the hpvmmodify add command with &apos;timex&apos;
#
# EXAMPLES:
#
#  Add all the disks in file "disklistfile" using defaults
#  # ./thisscript -a -P guest -f disklistfile
#
#  Add all the disks in file "disklistfile" 20 disks at a time
#  # ./thisscript -a -P guest -f disklistfile -n 20
#
#  Add the first 50 disks in the disklistfile, 20 disks at a time
#  NOTE: this will result in 3 calls to hpvmmodify, to
#        add 20 disks, another 20, and then the final 10.
#  # ./thisscript -a -P guest -f disklistfile -N 50 -n 20
#
#  NOTE: all of the above examples do not specify b,d,t values in the
#        hpvmmodify resource string, so the default algorithm is used,
#        to add 15 targets, from 0...14, and then increment to the next
#        virtual adaptor (skipping 0,3).
#
#  The following examples will cause the script to calculate and use
#  explicit values for b,d,t in the hpvmmodify resource string.
#
#  NOTE: Rules for specifying -t in this script:
#        0        Special case, means use 0...127
#        1...14   Invalid in this script, as this is part of the default
#                 range of 0...14
#        15...127 Use specified value as upper limit to target value before
#                 going to next virtual adaptor.
#
#  Add all disks in the file using the full range of target values 0...127:
#  # ./thisscript -a -P guest -f disklistfile -t 0
#
#  Add all disks in the file using a maximum target value of 30
#  # ./thisscript -a -P guest -f disklistfile -t 0
#
#
# ASSUMPTIONS AND LIMITATIONS
#
# - assume that the guest exists and may be modified
# - assume there are no storage devices assigned to the guest
# - assume the disks in the disklistfile are good
# - assume OK to add all disks as avio_stor
# - assume OK to add specified disks to the specified guest
# - limitation: 945 storage devices using default [b,d,t] values
# - limitation: 1024 max avio_stor storage devices
# - limitation: 127 max value for user specified target limit
# ------------------------------------------------------------------------------

#
# Script global variables
#
THISSCRIPT=$0
DFLTDISKLIMIT=945
MAXDISKCNT=1024
XNDEFAULT=10
BDT="" # default [b,d,t] setting

typeset -i BUS
typeset -i DEV
typeset -i TGT
typeset -i TGTMAX
typeset -i USERTGT
BUS=0
DEV=0
TGT=0
BUSMAX=7
DEVMAX=7
DEVSKIP=3
TGTMAX=127
USERTGT=0
WRKTGT=$TGTMAX

#
# function autobdt() - auto generates explicit b,d,t triples
#
function autobdt {
    # echo "autobdt() function not yet implemented"

    # use current BUS,DEV,TGT values

    BDT="$BUS,$DEV,$TGT"

    # setup BUS,DEV,TGT for next call
```

```
    TGT=$TGT+1
    if [ $TGT -gt $WRKTGT ]
    then
        TGT=0
        DEV=$DEV+1
    fi

    # Skip b,d of 0,3
    if [ $BUS -eq 0 ] &amp;&amp; [ $DEV -eq $DEVSKIP ]
    then
        DEV=$DEV+1
    fi

    if [ $DEV -gt $DEVMAX ]
    then
        DEV=0
        BUS=$BUS+1
    fi

    if [ $BUS -gt $BUSMAX ]
    then
        # NOTE: should not be here, but error out just in case.
        echo "ERROR: Max supported bus value exceeded, no more room for another adaptor."
        exit 1
    fi

} # end autobdt()

#
# function usage() - prints help text
#
function usage {

echo "usage: $THISSCRIPT [[-a] [-F flags] -f disklistfile [-N #] [-n #] -P guestname [-q] [-T]] | [-H|-h]"
echo "       -a                Add the disks (default is to only display what will be added)"
echo "       -F \"arg(s)\"       Additional hpvmmodify options or flags (double quotes required)"
echo "       -f disklistfile   File containing list of disks to add"
echo "       -h                Print usage (help)"
echo "       -H                Print usage (Help)"
echo "       -N #              Number of devices to add from the disklistfile"
echo "       -n #              Number of devices to add at one time (default: $XNDEFAULT)"
echo "       -P guestname      Name of Integrity VM (guest) to modify"
echo "       -q                Quite mode - no display of hpvmmodify command that will run"
echo "       -t targetmax      Max target value to use for -a disk:avio_stor:[b,d,targetmax]..."
echo "                         Valid values:"
echo "                           0       - special case: script will use full 0-127 range"
echo "                          15...127 - script will use specified max"
echo "                           1... 14 - not valid for this script, since 0-14 is"
echo "                                     the normal default range for target values"
echo "                                     if -t is not specified."
echo "       -T                Time the hpvmmodify add command with &apos;timex&apos;"

} # end usage()


#
# main() &apos;function&apos;
#

# Command option verification variables
typeset -i a
typeset -i F
typeset -i f
typeset -i N
typeset -i n
typeset -i P
typeset -i q
typeset -i s
typeset -i T
typeset -i t
a=0
F=0
f=0
N=0
n=0
P=0
q=0
s=0
T=0
t=0

# Variables for cmd-line arguments
DISKLISTFILE=""
GUESTNAME=""
TIMECMD=""
FLAGS=""

typeset -i ADDFLAG
typeset -i AUTOBDT
typeset -i QUIET
typeset -i USERTGT
typeset -i USERDISKLIMIT
typeset -i XN
```

```
ADDFLAG=0
AUTOBDT=0
QUIET=0
USERDISKCNT=0
USERTGT=0
XN=$XNDEFAULT


#
# Get cmd line options
#
while getopts :aF:f:HhN:n:P:qTt: option
do
    case $option in
    a) # add flag - do actual call to hpvmmodify
        ADDFLAG=1
        a=$a+1
        ;;
    F) # hpvmmodify flags
        FLAGS=$OPTARG
        F=$F+1
        ;;
    f) # disklist file
        DISKLISTFILE=$OPTARG
        f=$f+1
        ;;
    H) # Help
        usage
        exit 0
        ;;
    h) # help
        usage
        exit 0
        ;;
    N) # number of disks to add from the disklistfile
        USERDISKCNT=$OPTARG
        N=$N+1
        ;;
    n) # number of disks to add at a time
        XN=$OPTARG
        n=$n+1
        ;;
    P) # guest name
        GUESTNAME=$OPTARG
        P=$P+1
        ;;
    q) # quiet mode
        QUIET=1
        q=$q+1
        ;;
    T) # time the add command
        TIMECMD="timex"
        T=$T+1
        ;;
    t) # target max
        USERTGT=$OPTARG
        AUTOBDT=1
        t=$t+1
        ;;
    ?) # error
        echo "ERROR: Error with option: $OPTARG (unknown option, or missing value"
        usage
        exit 1
        ;;
    esac
done

#
# Verify cmd line options
#
if [ $a -gt 1 ] || [ $F -gt 1 ] || [ $f -gt 1 ] || [ $N -gt 1 ] || [ $n -gt 1 ] || \
   [ $P -gt 1 ] || [ $q -gt 1 ] || [ $T -gt 1 ] || [ $t -gt 1 ]
then
    echo "ERROR: Duplicate arguments are not allowed."
    exit 1
fi

if [ $P -eq 0 ]
then
    echo "ERROR: &apos;-P guestname&apos; must be specified."
    exit 1
fi

if [ $f -eq 0 ]
then
    echo "ERROR: &apos;-f disklistfile&apos; must be specified."
    exit 1
fi

if [[ ! -f $DISKLISTFILE ]]
then
    echo "ERROR: Could not find disklist file: $DISKLISTFILE"
    exit 1
```

```
fi

if [ ! -s "$DISKLISTFILE" ]
then
    echo "ERROR: Disklist file: $DISKLISTFILE is a zero-length file."
    exit 1
fi

GUESTSTATUS="`hpvmstatus -P $GUESTNAME -M 2> /dev/null`"
if [ -z "$GUESTSTATUS" ]
then
    echo "ERROR: Could not find guest: $GUESTNAME"
    exit 1
fi

if [ $t -eq 1 ]
then
    if [ $USERTGT -gt 0 ] &amp;&amp; [ $USERTGT -lt 15 ]
    then
        echo "ERROR: User specified target max (-t $USERTGT) must be 0 or in range 15...127."
        exit 1
    fi
    if [ $USERTGT -gt $TGTMAX ]
    then
        echo "ERROR: User specified target (-t $USERTGT) exceeds max value of $TGTMAX"
        exit 1
    fi

    if [ $USERTGT -ne 0 ]
    then
        WRKTGT=$USERTGT
    fi
fi

#
# Get disklist from file
#
DISKLIST="`cat $DISKLISTFILE`"

#
# Setup main loop variables
#
typeset -i DISKCNT
typeset -i FILEDISKCNT
FILEDISKCNT="`ls -1 $DISKLIST | wc -l`"
if [ $USERDISKCNT -eq 0 ]
then
    DISKCNT=$FILEDISKCNT
else
    if [ $USERDISKCNT -gt $FILEDISKCNT ]
    then
        echo "ERROR: -N value ($USERDISKCNT) is greater than number of disks in $DISKLISTFILE ($FILEDISKCNT)."

        exit 1
    else
        DISKCNT=$USERDISKCNT
    fi
fi

if [ $DISKCNT -gt $DFLTDISKLIMIT ] &amp;&amp; [ $AUTOBDT -eq 0 ]
then
    echo "ERROR: Diskcount greater than $DFLTDISKLIMIT requires target max flag (-t) to be set."
    exit 1
fi

if [ $DISKCNT -gt $MAXDISKCNT ]
then
    DISKCNT=$MAXDISKCNT
    echo "INFO: Set diskcount to supported maximum \($MAXDISCOUNT\)."
fi

typeset -i CMDIDX
typeset -i DISKIDX
CMDIDX=0
DISKIDX=0
BASEMODCMD="hpvmmodify -P $GUESTNAME $FLAGS"

#
# Main Loop
#
if [ $ADDFLAG -eq 0 ]
then
    echo "INFO: Add flag (-a) was NOT specified (no disks will be added)."
fi

ADDCMD="$BASEMODCMD"
for DISK in $DISKLIST;
do
    if [ $AUTOBDT -eq 1 ]
    then
        autobdt
    fi
    ADDRSRC="-a disk:avio_stor:$BDT:disk:$DISK"
```

```
    ADDCMD="$ADDCMD $ADDRSRC"
    DISKIDX=$DISKIDX+1
    CMDIDX=$CMDIDX+1

    # Run hpvmmodify if at the add multiplier (-n) or at the last disk
    if [ $CMDIDX -eq $XN ] || [ $DISKIDX -eq $DISKCNT ]
    then
        # Do the hpvmmodify
        if [ $QUIET -eq 0 ]
        then
            echo "Calling: $TIMECMD $ADDCMD"
        fi

        if [ $ADDFLAG -eq 1 ] # check for -a flag
        then
            $TIMECMD $ADDCMD
            RETVAL=$?
            if [ $RETVAL -ne 0 ]
            then
                typeset -i FINALCNT
                FINALCNT=$DISKIDX-$XN
                echo "ERROR - hpvmmodify failed. (total disks added: $FINALCNT)"
                exit 1
            fi
        fi

        # In progress status ...
        echo "Subtotal of disks added: $DISKIDX"

        # Reset hpvmmodify cmd string
        ADDCMD="$BASEMODCMD"
        CMDIDX=0
    fi

    if [ $DISKIDX -eq $DISKCNT ]
    then
        # all done
 break;
    fi
done

if [ $ADDFLAG -eq 1 ]
then
    echo "All done (total disks addded: $DISKCNT)"
else
    echo "All done (Not in add mode: no disks added)"
fi
exit 0
```

D Warranty and regulatory information

For important safety, environmental, and regulatory information, see *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at **http://www.hpe.com/support/Safety-Compliance-EnterpriseProducts**.

Warranty information

HPE ProLiant and x86 Servers and Options

http://www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise Servers

http://www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

http://www.hpe.com/support/Storage-Warranties

HPE Networking Products

http://www.hpe.com/support/Networking-Warranties

Regulatory information

Belarus Kazakhstan Russia marking

Manufacturer and Local Representative Information

Manufacturer information:

- Hewlett Packard Enterprise, 3000 Hanover Street, Palo Alto, CA 94304, U.S.

Local representative information Russian:

- **Russia**:

  ЗАО "Хьюлетт-Паккард А.О.", 125171, Россия, г. Москва, Ленинградское шоссе, 16А, стр.3, тел/факс: +7 (495) 797 35 00, +7 (495) 287 89 05

- **Belarus**:

  ИООО «Хьюлетт-Паккард Бел», 220030, Беларусь, г. Минск, ул. Интернациональная, 36-1, офис 722-723, тел.: +375 (17) 392 28 18, факс: +375 (17) 392 28 21

- **Kazakhstan**:

  ТОО «Хьюлетт-Паккард (К), 050040, Казахстан, г. Алматы, Бостандыкский район, ул. Тимирязева, 28В, 1 этаж, тел./факс: +7 (727) 355 35 50, +7 (727) 355 35 51

Local representative information Kazakh:

- **Kazakhstan**:

  ЖШС «Хьюлетт-Паккард (К)», Қазақстан, Алматы қ., Бостандық ауданы, Тимирязев к-сі, 28В, тел./факс: +7 (727) 355 35 50, +7 (727) 355 35 51

Manufacturing date:

The manufacturing date is defined by the serial number.

CCSYWWZZZZ (serial number format for this product)

Valid date formats include:

- YWW, where Y indicates the year counting from within each new decade, with 2000 as the starting point; for example, 238: 2 for 2002 and 38 for the week of September 9. In addition, 2010 is indicated by 0, 2011 by 1, 2012 by 2, 2013 by 3, and so forth.
- YYWW, where YY indicates the year, using a base year of 2000; for example, 0238: 02 for 2002 and 38 for the week of September 9.

Turkey RoHS material content declaration

Türkiye Cumhuriyeti: EEE Yönetmeliğine Uygundur

Ukraine RoHS material content declaration

Обладнання відповідає вимогам Технічного регламенту щодо обмеження використання деяких небезпечних речовин в електричному та електронному обладнанні, затвердженого постановою Кабінету Міністрів України від 3 грудня 2008 № 1057

Glossary

This glossary defines the terms and abbreviations as they are used in the Integrity VM product documentation.

Accelerated Virtual Input/Output
: See AVIO

adoptive node
: The cluster member where the package starts after it fails over.

APA
: Auto Port Aggregation. An HP-UX software product that creates link aggregates, often called "trunks," which provide a logical grouping of two or more physical ports into a single "fat pipe". This port arrangement provides more data bandwidth and higher reliability than would otherwise be available.

application
: A collection of processes that perform a specific function. In the context of virtual machine clusters, an application is any software running on the guest.

assignable resource
: The resources that you can designate to be assigned to a partition.

asymmetric Serviceguard configuration
: A cluster configuration in which the cluster nodes do not have access to the same physical storage and network devices.

autoboot
: A characteristic of a virtual machine whereby it is set to start whenever Integrity VM starts. Virtual machines can be set to either `auto` or `manual` boot using the `-B` option to the `hpvmcreate`, `hpvmmodify`, `hpvmmigrate`, or `hpvmclone` commands.

available resources
: Processors, memory, and I/O resources that are not assigned to a virtual machine. These resources are available to be used in new partitions or can be added to existing partitions.

AVIO
: Accelerated Virtual Input/Output. An I/O protocol that improves virtual I/O performance for network and storage devices used within the Integrity VM environment. The protocol also enables support for a greater number of virtual I/O devices per guest. Special drivers are required on both the VSP and guests. Participating guests must include a virtual I/O device configured to use the AVIO protocol.

backing store
: The physical device on the VSP that is allocated to guests, such as a disk or file.

Base memory
: This can be used by vPar kernel for critical data structures. You can add the memory but cannot delete from a live vPar.

Blade
: A board that contains CPUs and memory, and slots for C-class mezzanine cards, and onboard NICs. A blade is the equivalent of a cell in terms of being the unit of assignment for defining nPartitions.

BMC
: Baseboard Management Controller. The Management Processor (MP) console for Intel® Itanium systems.

boot virtual machines
: To load a virtual machine's operating system and start it. Once a virtual machine has been configured with an operating system, it is considered a guest, and is started automatically when Integrity VM starts, or manually using the `hpvmstart` command.
See also start virtual machines.

c3000 enclosure
: The HPE BladeSystem c3000 enclosure works well in smaller data centers. A single c3000 enclosure is 6U high and can hold up to eight server, storage, or I/O option blades and up to four interconnect modules.

c7000 enclosure
: The BladeSystem c7000 enclosure is optimized for enterprise data centers. A single c7000 enclosure is 10U high and can hold up to 16 server, storage, or I/O option blades and up to eight interconnect modules.

captive virtual console account
: A special-purpose user account created on the VSP for each guest administrator or operator.

cell local memory
: See CLM

CLM	Non-interleaved memory that can be quickly accessed by processors residing on the same socket as the memory. This is the same concept as SLM.
cluster	Two or more systems configured together to host workloads. Users are unaware that more than one system is hosting the workload.
cluster member	A cluster node that is actively participating in the Serviceguard cluster.
cluster node	A system (VSP or guest) configured to be a part of a Serviceguard cluster.
CRA	Critical Resources Analysis.
Deconfigured	The term used to describe the health of a resource that has been marked as unusable by the Health Repository. Such a resource will be excluded from partition activity.
dedicated device	A pNIC or storage unit that is dedicated to a specific virtual machine. A dedicated device cannot be used by multiple virtual machines.
direct I/O networking	The direct I/O networking feature allows virtual machines to directly control I/O devices.
distributed guests	Guests that has been configured as a Serviceguard package.
DLA	Device Level Assignment
EFI	Extensible Firmware Interface. The boot firmware for all Integrity systems.
enclosure	An BladeSystem c-Class enclosure holds ProLiant and Integrity server blades, storage blades, I/O option blades, interconnect modules (switches, pass-thru modules, and Virtual Connect modules), a NonStop passive signal midplane, a passive power backplane, power supplies, fans, and Onboard Administrator modules.
entitlement	The amount of a system resource (for example, a processor) that is guaranteed to a virtual machine. The actual allocation of resources to the virtual machine can be greater or less than its entitlement, depending on the virtual machine's demand for processor resources and the overall system processor load.
event log	Information about system events. An event log indicates what event has occurred, when and where it happened, and its severity (alert level). Event logs do not rely on normal I/O operation.
extensible firmware interface	*See* EFI.
failover	The operation that takes place when a primary service (network, storage, or CPU) fails, and the application continues operation on a secondary unit. In the case of Serviceguard virtual machines, the virtual machine can fail over to another cluster member. In case of a network failure, on a properly configured system the virtual machine can fail over to another LAN on the same cluster node.
FLA	Function Level Assignment.
Floating memory	This is typically used for user applications. You can either add or delete the memory from a live vPar.
guest	The virtual machine running the guest OS and guest applications.
guest administrator	The administrator of a virtual machine. A guest administrator can operate the virtual machine using the `hpvmconsole` command with action that can affect the specific guest only.
guest application	A software application that runs on a guest.
guest application package	A guest application that has been configured as a Serviceguard package.
guest console	The virtual machine console that is started by the `hpvmconsole` command.
guest management software	Software that is provided with Integrity VM that you install on the guest to ensure the guest is manageable by Integrity VM and other components of the Virtual Server Environment and Integrity Virtual Server Manager.
guest operator	The administrator of the guest OS. This level of privilege gives complete control of the virtual machine but does not allow control of the other guests, the VSP, or the backing stores.
guest OS	Guest operating system.

guest package A Serviceguard package that is an Integrity VM guest.

host
1. A system or partition that is running an instance of an operating system.
2. The physical machine that is the VSP for one or more virtual machines.

host administrator The system administrator. This level of privilege provides control of the VSP system and its resources, as well as creating and managing vPars/VMs.

host name The name of a system or partition that is running an OS instance.

host OS The operating system that is running on the host machine.

HP SIM HP System Insight Manager.

HP SMH System Management Homepage.

HPE Matrix OE HPE Matrix Operating Environment.

Ignite-UX The HP-UX Ignite server product. Used as a core build image to create or reload HP-UX servers.

ILM Interleaved Memory. Is implemented as Partition Memory in HPE Superdome 2, which includes Direct Access Partition Memory and Agent Access Partition Memory

Integrity Virtual Machines The Integrity Virtual Machines product, which allows you to install and run multiple systems (virtual machines) on the same physical host system.

Integrity VM *See* Integrity Virtual Machines..

ISEE HPE Instant Support Enterprise Edition. A secure remote support platform for business servers and storage devices.

LAN Local area network.

localnet A virtual switch created by default when Integrity VM is installed on a VSP. The local network created by this vswitch can be used for communications among guests but not for communication between the VSP and any guest or between any external system and a VM guest.

Machine check abort See MCA.

max. Maximum.

MCA Machine check abort

migration The operation of stopping a Serviceguard package on one cluster member and then starting it on another cluster member. Migrating the package (for example, a virtual machine), can be useful in system management procedures and workload balancing.
See also virtual machine migration..

min. Minimum.

multiserver environment A Serviceguard cluster consisting of VSP systems.

N_Port ID Virtualization See NPIV.

NIC Network Interface Card. Also called “network adapter.”

nPartition A partition that is assigned one or more blades and optionally zero or more I/O bays. An nPartition can run a single OS (either a standalone OS or an HPVM host), or an nPar can be sub-divided into customer-defined vPars. A Superdome 2 nPar works like an nPar on cellular servers.

NPIV N_Port ID Virtualization. A Fibre Channel facility allowing multiple N_Port IDs to share a single physical N_Port.

NSPOF No single point of failure. A configuration imperative that implies the use of redundancy and high availability to ensure that the failure of a single component does not impact the operations of the machine.

online VM migration Enables a running guest and its applications to be moved from one VSP to another without service interruption.

OVMM Online VM migration. See online VM migration.

package configuration script A script that is customized for each virtual machine Serviceguard package and that contains specific variables and parameters, including logical volume definitions, for that virtual machine.

package control script	A script containing parameters that control how Serviceguard operates.
Partition Number	A unique numeric value assigned to a partition.
PMAN	Platform Manager. See VSP.
pNIC	Physical network interface card.
primary node	The cluster member on which a failed-over package was originally running.
redundancy	A method of providing high availability that uses multiple copies of storage or network units to ensure services are always available (for example, disk mirroring).
restricted device	A physical device that can be accessed only by the VSP system. For example, the VSP boot device should be a restricted device.
SAN	Storage Area Network.
Serviceguard	Serviceguard allows you to create high-availability clusters of HP 9000 or Integrity servers. Serviceguard can be used to manage virtual machines as Serviceguard packages. A Serviceguard package groups application services (individual HP-UX processes) together and maintains them on multiple nodes in the cluster, making them available for failover.
Serviceguard node	A Serviceguard node, within the Integrity VM context, is a VSP. See VSP.
SGeRAC	Serviceguard extension for real application clusters.
SGeSAP	Serviceguard extension for SAP.
shared device	A virtual device that can be used by more than one virtual machine.
SLM	Non-interleaved memory that can be quickly accessed by processors residing on the same cell as the memory. This is the same concept as CLM.
SLVM	Shared Logical Volume Manager.
socket local memory	See SLM
SSH	Secure Shell
start virtual machines	To start a virtual machine that has been booted before. *See also* boot virtual machines.
storage unit	A file, DVD, disk, or logical volume on the VSP that is used by the virtual machines running on the VSP.
symmetric Serviceguard configuration	A cluster configuration in which the nodes share access to the same storage and network devices.
TOC	Transfer of control
Transfer of control	See TOC.
VBVsw	VLAN-backed vswitch
virtual console	The virtualized console of a virtual machine that emulates the functionality of the Management Processor interface for Integrity servers. Each virtual machine has its own virtual console from which the virtual machine can be powered on or off and booted or shut down, and from which the guest OS can be selected.
virtual device	An emulation of a physical device. This emulation, used as a device by a virtual machine, effectively maps a virtual device to an entity (for example, a DVD) on the VSP.
virtual machine	Virtual hardware system. Also called VM.
virtual machine application	The executable program on the VSP that manifests the individual virtual machine. The program communicates with the loadable drivers based on information in the guest-specific configuration file, and it instantiates the virtual machine.
virtual machine console	The user-mode application that provides console emulation for virtual machines. Each instance of the virtual machine console represents one console session for its associated virtual machine.

virtual machine host *See* VSP.

Virtual Machine Manager (VMM) The management application responsible for managing and configuring Integrity Virtual Machines.

virtual machine migration Migration of a virtual machine from one VSP system to another by using the Integrity VM command `hpvmmigrate`. Do not use this command for virtual machine packages.

virtual machine package A virtual machine that is configured as a Serviceguard package.

virtual network A LAN that is shared by the virtual machines running on the same VSP or in the same Serviceguard cluster.

virtual switch *See* vswitch.

Virtualization Services Platform *See* VSP.

VM See *Virtual machine*.

vNIC Virtual network interface card (NIC). The network interface that is accessed by guest applications.

vPar Virtual partition. A partition that is created and managed from the VSP. A vPar is assigned CPU cores, and memory.

VSMgr Virtual Server Manager.

VSP Virtualization Services Platform. The management platform for creating and managing virtual partitions. Provides both command-line interface and graphical user interface for configuring and managing vPars.

vswitch Virtual switch. A component in the guest virtual network. By associating the vswitch with a physical working LAN on the VSP, you provide the guest with the capability of communicating outside the localnet.

WBEM Web-Based Enterprise Management. A set of Web-based information services standards developed by the Distributed Management Task Force, Inc.A WBEM provider offers access to a resource. WBEM clients send requests to providers to get information about and access to the registered resources.

workload The collection of processes in a virtual machine.

Index

W