PARTICIPANT GUIDE

Sun Cluster 3.0 Training

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A.

Part No.: TBD
Course No.: TBD
Revision 1.1, September 2000

Contents

CHAPTER 1

Sun Cluster 3.0 Overview

Objectives

Purpose:

Provide an overview of the features and components of Sun Cluster 3.0.

Prerequisites:

Solaris system and network administration experience

Knowledge of Sun server hardware

Knowledge of Sun storage hardware

Objectives:

After completing this chapter, the participant will be able to:

- Define the concept of a cluster
- Describe the features and benefits of Sun Cluster 3.0
- Describe the components of Sun Cluster 3.0
- Describe basic cluster concepts, such as:
    - Cluster Topologies
    - Quorum Devices and Failure Fencing
    - Logical Hosts

Chapter Objectives

- Introduce Sun Cluster 3.0 features
- Describe the architecture of Sun Cluster 3.0
- Describe the Sun Cluster 3.0 components
- Explain some common clustering concepts

What is a Cluster?

Before we jump into the world of Sun Clusters, perhaps we should start out with defining a cluster in general. Just what is a cluster anyway?

A cluster is a group of completely separate systems (or nodes) that are connected together to work towards a common goal. Each node is, in fact, a fully functional stand-alone server system. However, in a clustered configuration, the nodes work together as a single entity to cooperatively provide applications, system resources and data to the user community.

So why are clusters such a hot item these days? Simply put, clusters offer the following advantages:

- **Increased Availability** - a cluster will also provide increased availability, should one node fail, other cluster nodes can continue to provide data services, where the failed node’s workload can be spread across the remaining nodes.
- **Increased Performance** - a cluster is scalable across nodes, providing a highly scalable growth path: when an additional node is added, it brings with it not only increased processing power, but also increased memory, storage and I/O bandwidth.

In addition, these benefits can be realized without highly specialized hardware (that is, without high cost) -- the nodes are off-the-shelf servers that, in turn, are connected to off-the-shelf storage platforms. Some additional hardware to accommodate the inter-node communications infrastructure along with specialized software are all that’s needed to convert this off-the-shelf hardware into a working cluster.

What is a Cluster?

- A connected set of standalone server systems (called "nodes") working as a single entity towards a common goal
- Provides increased performance (scalability)
- Provides increased availability
- Utilizes "off-the-shelf" components

Some Definitions

Clustering is a complex topic. Just like any other computing paradigm, it has its own set of terms and definitions. Before we move on, let's try to define some of these terms:

- High Availability
- Failover
- Scalability
- Membership

Some Definitions

- High Availability
- Failover
- Scalability
- Membership

High Availability (HA)

High Availability (HA)

One very popular use of clusters is to provide *high availability*. What exactly is high availability?

A very basic definition may be the following: the computer system (this means hardware, software and data) is *available* whenever it's needed.

Obviously, there is a lot of ambiguity in this definition, especially with the "whenever it's needed" part! But the truth of the matter is, this is the expectation for a lot of the "mission-critical" applications out there -- basically: never ever go down, unless it's a planned outage, and if it does happen to go down unexpectedly, don't stay down for very long.

Unfortunately, with the complexity of today's hardware and software, things **will** break unexpectedly. This is where clusters can help us maintain a higher level of availability than single servers: **Cluster systems can survive a single failure (and sometimes even multiple failures), thus making them more *highly available* then single servers**.

Thus, for our purposes, the definition of high availability is the ability of a cluster system to keep a data service up and running, even through a failure that would normally take down a single server system.

In addition, the chances are, a cluster will keep the data service (or services) running while whatever caused the failure is being fixed (as long as another failure is not experienced before the first failure is fixed).

High Availability (HA)

- ➤ In today’s business environment, computer systems (both hardware and software) are expected to be available whenever they are needed
- ➤ Unfortunately, things *will* break ...
- ➤ Clusters provide high availability by keeping an application running, even during a failure
- ➤ In addition, the cluster can keep the data service running while a failure condition is corrected

Failover

Failover

Failover is the process by which the framework of the cluster provides high availability. It is essentially the series of steps that the cluster performs to migrate a data service from one node to another.

Most HA clusters are designed to run a particular application on one of the nodes of the cluster. If a failure condition causes this node to fail, all the relevant resources which are required by the data service are migrated to a designated backup node.

A typical failover process may look like this (assuming a node failure has been detected by the cluster framework):

1. Determine the resources required by the data service (usually pre-defined to the cluster framework when the cluster is configured)

2. Migrate any system resources (e.g. disk and/or network resources) hosted by the failed node to an appropriate backup node(s)

3. Start up the data service (daemons, etc.) and perform crash recovery, if required, on the backup node

After the failover is complete, client systems will be able to reconnect to the data service. Note that there may be an interruption of service associated with the failover -- the client system will view the failover as a crash of the server followed by a reboot. Processes that were “in progress” when the failure occurred may need to be restarted by the client systems, however the fact that the server is now a *different physical machine* should not make any difference what so ever.

Switchover is the administrator-controlled process of migrating a data service from one node in a cluster to another node. The important difference between switchover and failover is that a switchover is a manual procedure whereas a failover occurs because a node fails.

Failover

- Processes used by the cluster framework to migrate a data service from a failed node to a backup node
- Typical failover processes (after a failure is detected):
    - Determine resources required by the data service
    - Migrate any system resources (e.g. disk and/or network resources) hosted by the failed node to an appropriate backup node(s).
    - Start up the data service on a backup node
- A failover may cause an interruption of service
- Client systems view the failover as a crash followed by a reboot of the server -- client processes in progress when the crash occurred may need to be retried or restarted
- Switchover is a manual procedure to migrate a data service from one node of a cluster to another node

Scalability

Scalability

Another important property of a cluster is *scalability*. While high availability utilizes the cluster framework to provide redundancy, scalability is concerned with utilizing the cluster framework to *provide constant response time or throughput without regard to load*.

The scalability features of a cluster are designed to leverage the multiple computing platforms (the nodes) to concurrently run an application, thus providing increased performance.

This can be accomplished in several ways:

- Create special, parallel versions of the applications that are designed to utilize the cluster framework to allow the multiple nodes to work concurrently
- Utilize the cluster framework to provide “load-balancing” capabilities to off-the-shelf applications. Incoming requests for data are distributed to the nodes of the cluster for service, thus allowing multiple nodes in the cluster to concurrently provide data services to client systems.

The concepts of scalability and high availability are not mutually exclusive. An application designed to be scalable on a cluster will also be able to take advantage of the high availability features of the cluster -- if one of the nodes fails, the cluster framework will simply remove it from the set of nodes that are providing the scalable service, allowing the remaining nodes to continue.

Scalability

- The scalability features of the cluster are designed to utilize the multiple nodes of the cluster to *concurrently* run a data service, thus providing a constant response time or throughput, regardless of the load.
- Provides increased performance, each node can simultaneously be processing and serving data
- Two ways of accomplishing this:
  - Create specialized versions of software to take advantage of the cluster framework to be able to process and serve data in parallel. Examples of this are parallel database products
  - Provide a load balancing framework in the cluster to automatically distribute incoming data service requests to the different, eligible nodes in the cluster for service
- Scalable clusters can also provide high availability features

Membership

Membership

Membership is one of the central concepts of a cluster -- basically it's a determination of which nodes are part of the cluster and which nodes are not. This sounds simple enough, but determining membership and more importantly, what to do when the membership changes, is one of the more complex tasks performed by the cluster framework.

Cluster membership brings up a couple of tricky issues:

1. What happens when the membership changes (for example, when a node joins the cluster or leaves the cluster)?
2. How do we make sure that "bad" nodes leave the cluster?
3. How do we ensure that "bad" nodes stay out of the cluster until they are fixed?

Determination of membership is usually done with a heartbeat mechanism through some type of *cluster interconnect*. Each node will periodically send out an "I'm alive" message, as well as monitor the "I'm alive" messages from the other nodes.

Loss of a heartbeat, or gaining of a new heartbeat message from a node, will cause the situation in item 1 -- a change in membership. The cluster framework will need to determine the new membership and redistribute any affected resources accordingly.

Another interesting problem related to cluster membership is ensuring that nodes with bad hardware (specifically a bad cluster interconnect) are removed from the cluster and not allowed to join the cluster until the problem is fixed. This is a process called *failure fencing*, which ensures that a cluster cannot partition itself into separate independent clusters. This situation would be extremely dangerous, possibly leading to data corruption on the cluster's storage devices

We will be covering these topics in more detail (as they relate to Sun Cluster 3.0) later in this chapter.

Membership

- Basically - “Who’s part of the cluster and who’s not?”
- Usually done with a heartbeat mechanism - each node sends out periodic “I’m alive” messages while listening for the equivalent messages from other nodes in the cluster
- Changes in membership will result in a reconfiguration of the cluster resources - redistributing the workload to match the new membership
- Determination of valid cluster members is very important, rogue members (nodes that, due to a hardware/software failure, cannot work in cooperation with the other nodes of the cluster) must be excluded from the working cluster and must also be prevented from forming their own competing cluster
- The process of excluding “bad” nodes is called *failure fencing*, and is a very important topic, since competing cluster partitions can cause data corruption

What is a *Sun* Cluster?

Sun Cluster 3.0 is the newest version of Sun’s clustering solution. It’s an upgrade to Sun Cluster 2.2. A Sun Cluster can consist of up to 8 Ultra Enterprise servers coupled with a variety of Sun storage platforms.

Sun Cluster 3.0 is a general purpose cluster, in that it can be configured for maximum *scalability* (by configuring scalable data services), maximum *availability* (by configuring failover data services) or a mixture of *both*.

In addition to the base cluster framework, Sun Cluster 3.0 also can be configured to work with various off the shelf Solaris applications, such as relational database systems, internet servers and other data service applications.

What is a *Sun* Cluster?

- Sun Cluster 3.0 is an upgrade to Sun Cluster 2.2
- General Purpose Cluster - supports both scalability *and* availability
- Can be configured to work with a variety of off the shelf software applications

Sun Cluster 3.0 Features

Some features of Sun Cluster 3.0 are the following:

- ➤ **Up to 8 Nodes**

  Sun Cluster 3.0 will support a cluster containing up to eight Ultra Enterprise server nodes.

- ➤ **Cluster File System**

  The Cluster File System will allow mounting of cluster-wide UFS or HSFS file systems, allowing concurrent, continuous access to the filesystem from any node in the cluster. (VXFS file system will be supported after GA.)

- ➤ **Global Device Access**

  Disk devices, tape drives and CD-ROM drives on a node can be accessed from any node in the cluster.

- ➤ **Cluster Networking**

  While each node retains its own, publicly accessible IP address, a global IP address can be configured for the applications on the cluster, where the data service requests received through the global address may be distributed to different nodes in the cluster based on a selected load balancing policy.

- ➤ **Failover and Scalable application support**

  Sun Cluster 3.0 will support a number of off-the-shelf Solaris applications, making these applications either highly available, scalable or both.

- ➤ **Rich Set of Cluster APIs**

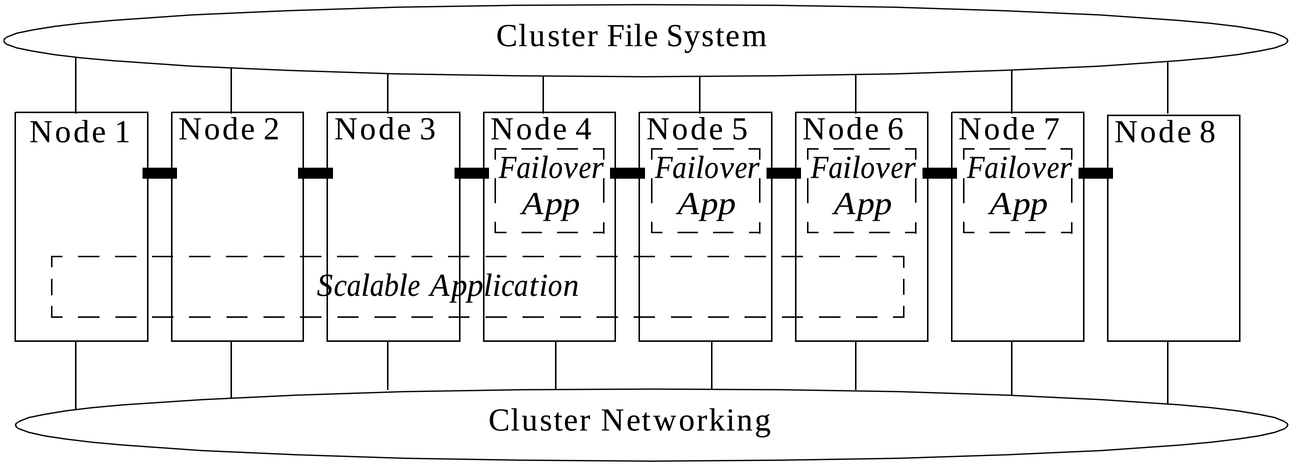
  A set of cluster APIs will be included to integrate applications with the cluster framework.

- ➤ **Sun Management Center-based monitoring**

  Sun Cluster 3.0 can be monitored using Sun’s Sun Management Center system management tool.

Sun Cluster 3.0 Features

- ➤ Up to eight nodes
- ➤ Cluster File System
- ➤ Global Device Access
- ➤ Cluster Networking
- ➤ Failover and Scalable Application Support
- ➤ Rich Set of Cluster APIs
- ➤ Sun Management Center-based Monitoring



Case Studies

Now that we know the features of Sun Cluster 3.0, let's see how we can use SC 3.0 to provide solutions to some "real world" business problems:

- ❑ Increasing the availability of data services
- ❑ Increasing the scalability of a web server
- ❑ Increasing both the availability and scalability of data services

Using Sun Cluster 3.0 to provide High Availability

Sun Cluster 3.0 can be used to increase the availability of supported data service applications. Let's examine the following scenario:

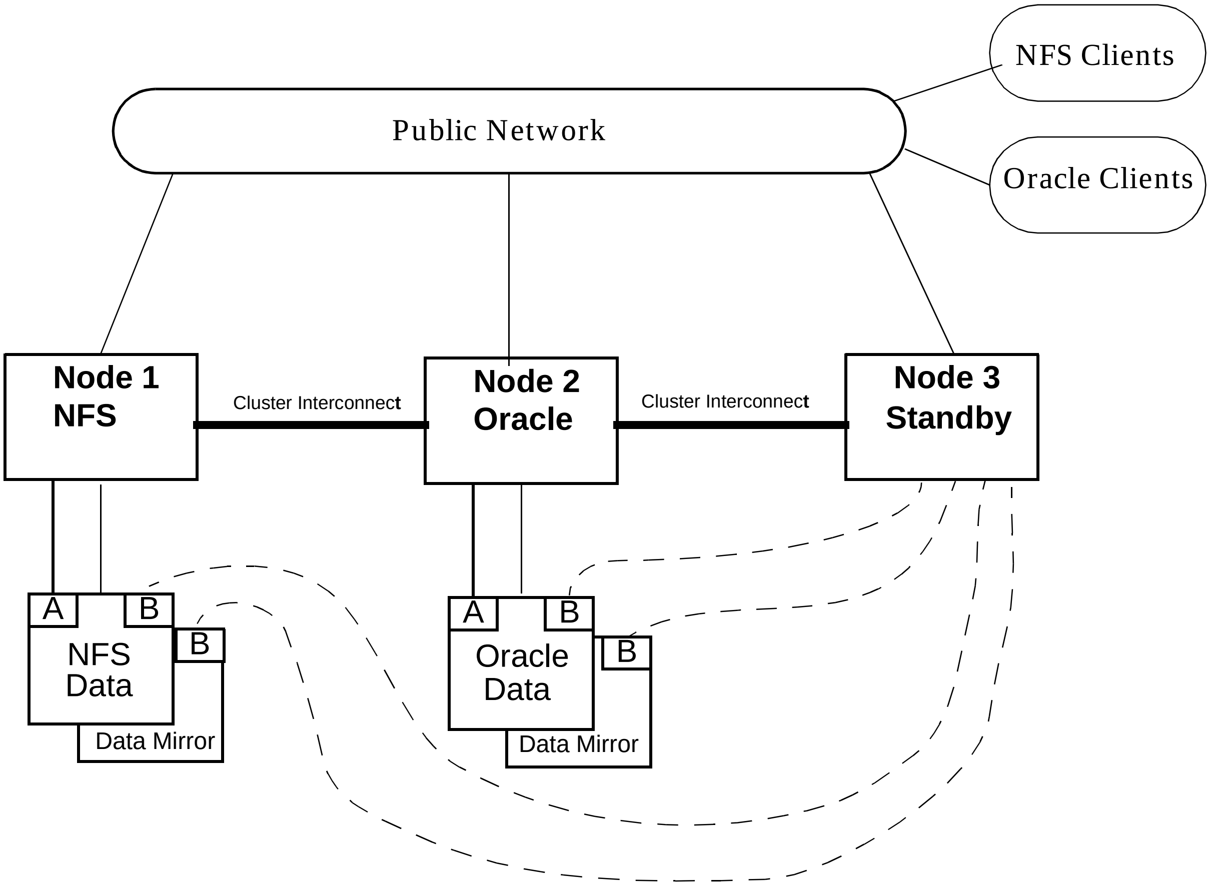
- ➤ We have a need to maintain a large NFS server, holding the shared files for our engineering workstations
- ➤ We also have a large Oracle database server for our mission critical ERP application

By installing these data services onto a three node cluster, we can significantly increase their availability. We can configure the cluster such that each application has its own server, with a remaining standby server, ready to automatically take over the NFS or Oracle data service (or both) should a failure occur. The data services in the cluster will stay available, even if:

- ❑ The node housing the NFS server suffers a catastrophic hardware or software failure
- ❑ The node housing the Oracle database suffers a catastrophic hardware or software failure
- ❑ An I/O path fails between the NFS or Oracle server and its associated disk resources
- ❑ A network interface fails on one of the nodes
- ❑ One of the disk arrays fails

Case Study - High Availability Cluster

- ➤ Data Services: NFS server and Oracle database server
- ➤ Cluster Solution: 3 Node cluster - a dedicated server for each data service and a shared standby server
- ➤ Can survive failures of either (or even both) online servers as well as network interface failures, disk I/O data path failures or disk array failures



Case Studies

Using Sun Cluster 3.0 to Increase Scalability

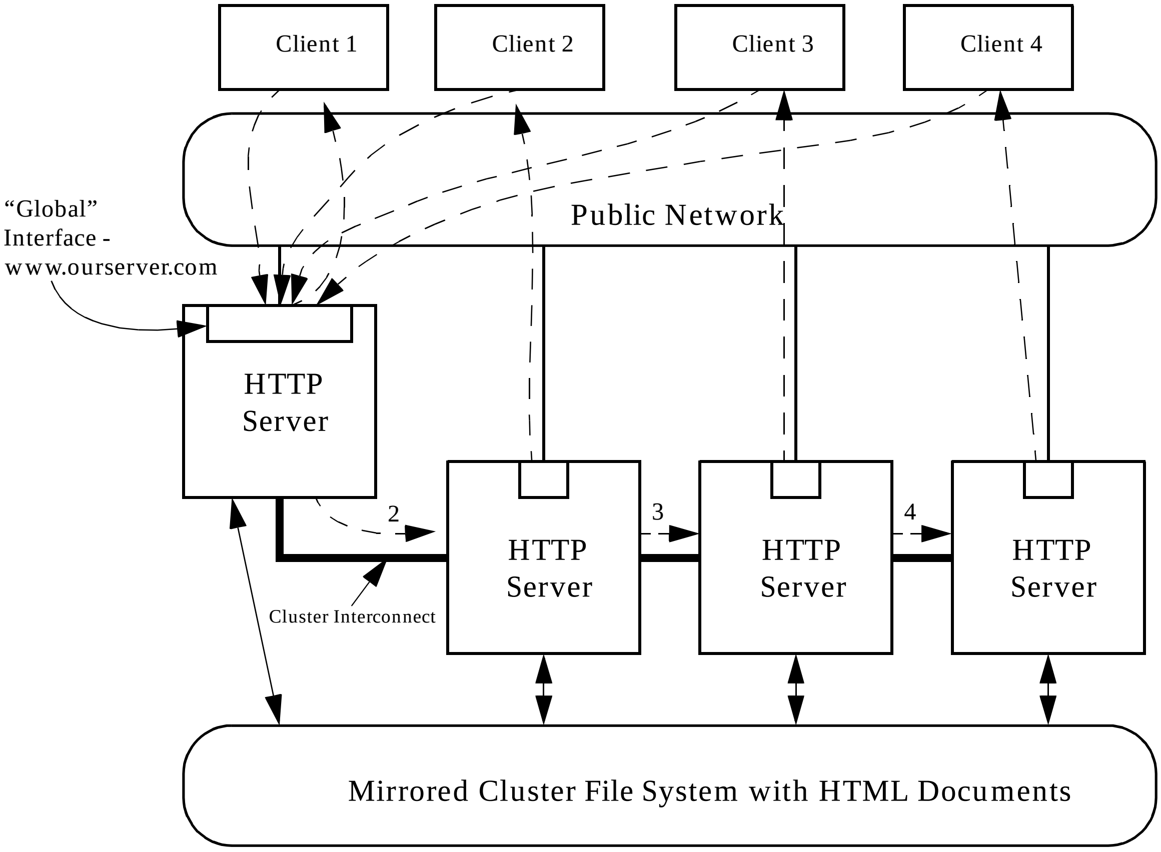
Sun Cluster 3.0 can be used to increase the scalability of certain data service applications such as web servers. When the cluster is used to increase scalability, an application is “spread” across several nodes of the cluster, making the cluster appear as a single, large computing entity to the client systems.

One scenario where Sun Cluster 3.0 can increase scalability of an application is with a company’s internet web server. Being able to handle large amounts of traffic without significant increases in response time is the goal of almost all web servers. By setting up the web server as an SC 3.0 scalable data service, the following benefits are realized:

- Multiple nodes in the cluster are used to answer incoming requests, thus allowing *simultaneous* processing and fulfilling of web page requests
- A weighted load-balancing algorithm is used to distribute load among nodes
- If a node crashes, the cluster framework will keep the data service running with the remaining nodes, thus making the web service highly available as well
- A single copy of the site’s HTML documents can be placed on a globally accessible cluster file system, thus making it unnecessary to maintain multiple copies of the site’s documents
- Critical data is mirrored across multiple disk arrays, protecting against data loss due to disk or array failures
- Data backup can be done from any node, to a globally accessible tape drive

Case Study - Scalable Cluster

- Data Service: Web Server
- Cluster Solution: Four node cluster, each node able to answer web page requests
- Web page requests are distributed to the nodes in the cluster using a weighted algorithm; each node answers a client request directly through the public network interface
- All nodes have access to a single copy of the HTML pages through the cluster file system



Case Studies

Using Sun Cluster 3.0 to Provide Both Scalability and Availability

Sun Cluster 3.0 allows the combination of HA data services together with scalable data services in the same cluster. A scenario where a cluster can be used to provide both scalability and availability is that of an e-commerce server. E-commerce servers are usually made up of at least two components: a web server to serve the pages to the buying public and a database server to hold transaction and product information.

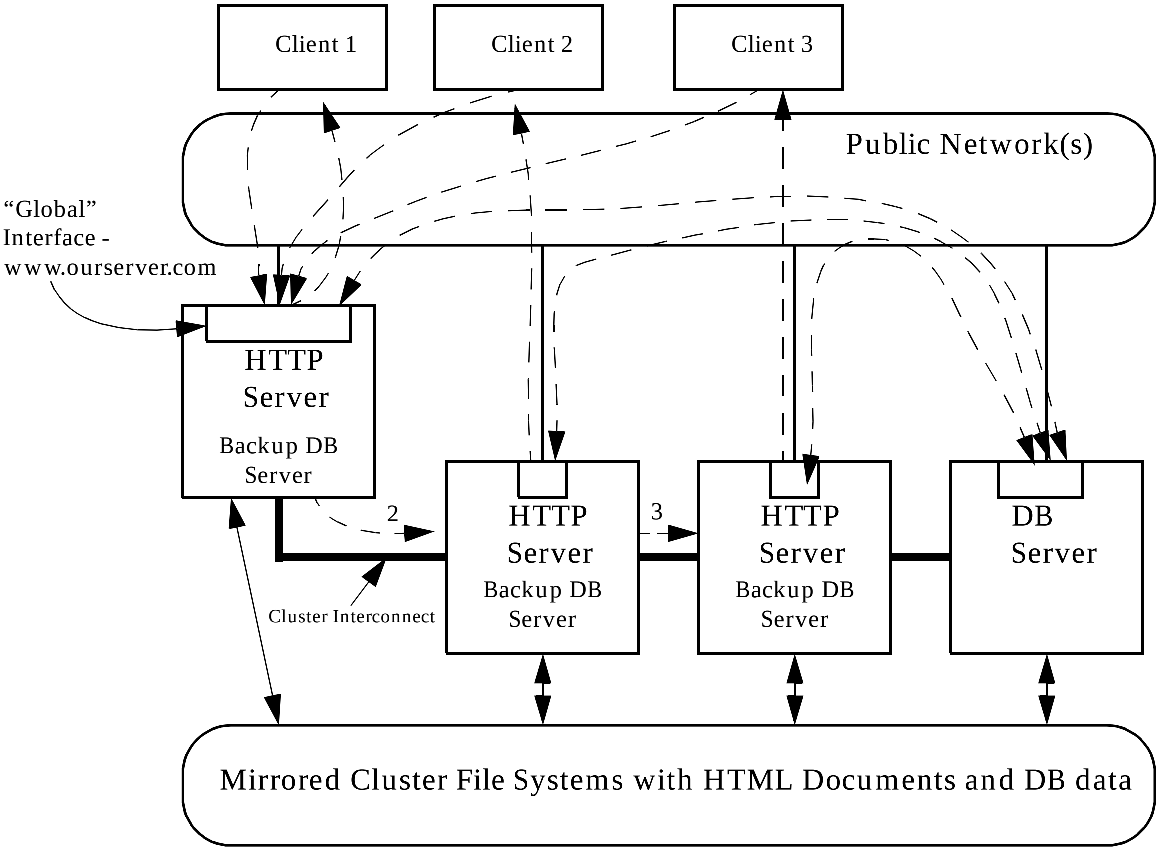
A 4 node cluster can be configured to provide both scalable web service and HA database service. The first three nodes of the cluster can be configured to provide the scalable web service while the forth node can be configured as the primary database server, with the first three nodes as backup nodes. This will allow nodes 1-3 to devote 100% of their resources towards web service and node 4 to devote all of its resources to providing the back-end database service. If node 4 happens to fail, one of the remaining nodes will be used for both database and web services (the other 2 nodes will still be able to devote 100% of their resources to web service).

The cluster configuration can provide the following benefits:

- Ability to survive the loss of a node, network interface or disk I/O path
- Load balancing algorithm will evenly distribute the traffic among the configured web server nodes
- Data for both the web server and the database server is mirrored across multiple arrays, protecting against disk or array failure
- Data backup can be performed from any node of the cluster to a globally accessible tape drive
- A single copy of the HTML documents can be placed on a globally accessible cluster file system, simplifying web site maintenance

Case Study - HA and Scalability

- ➤ Data Service: E-Commerce Server -- Web server and DB server
- ➤ Cluster Solution: 4 Node cluster providing both scalable web service and HA database service
- ➤ Nodes can be allocated to provide the scalable web services as well as serve as backup nodes for the database server
- ➤ Advantages of both of the previous case studies can be realized

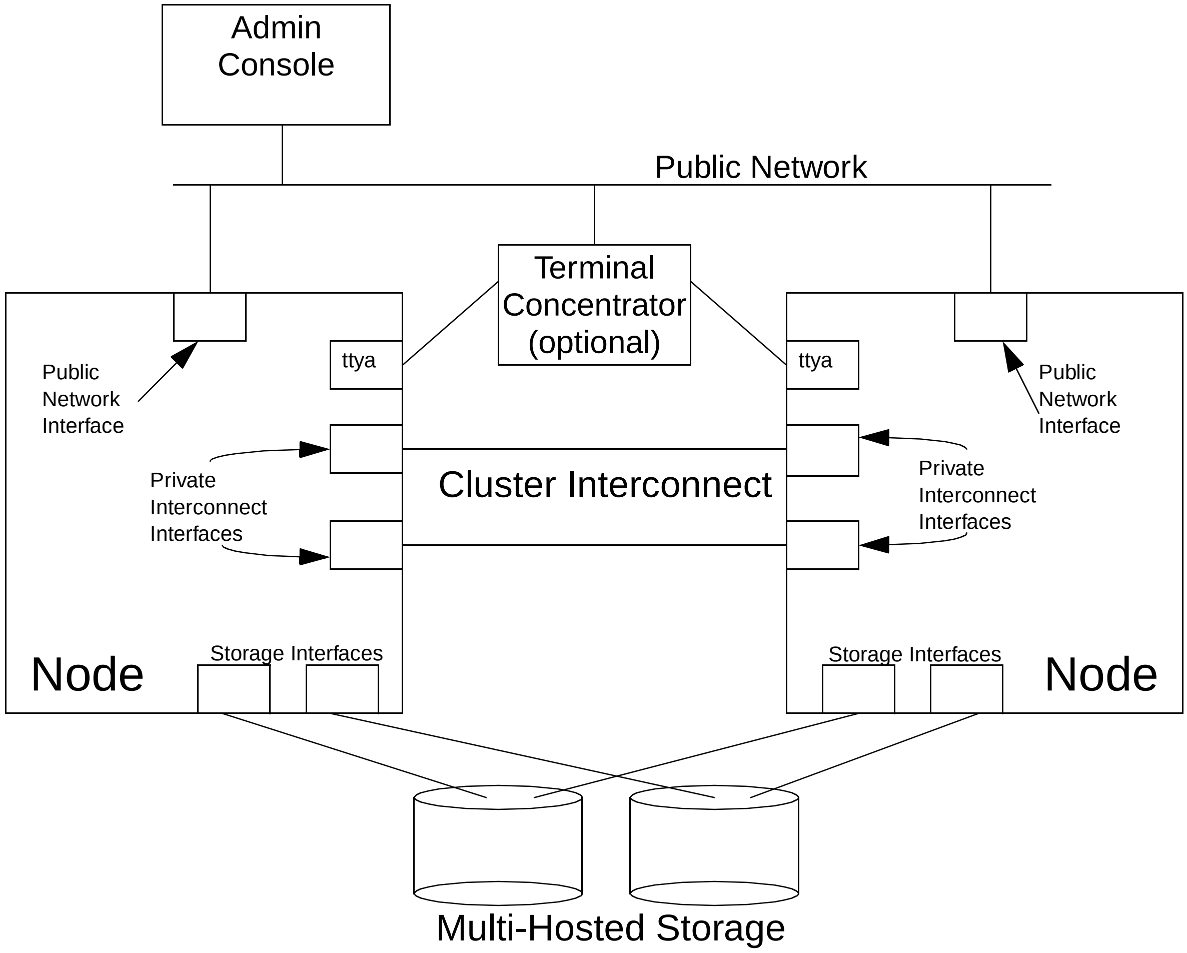


Sun Cluster 3.0 Architecture - Hardware Components

There are a number of hardware components that make up a cluster. These include:

- **Nodes** - the main computing platforms of the cluster
- **Cluster Interconnect** - Provides a channel for inter-node communication
- **Public Network Interfaces** - the network interfaces used by client systems to access data services housed on the cluster
- **Multi-Hosted Storage** - Disk drive arrays that are able to be accessed by multiple hosts
- **Terminal Concentrator** (optional) - Supplies console level access to each node in the cluster through the public network.
- **Administrative Console** - A separate workstation used to administer the console

Sun Cluster 3.0 Hardware Components



- Nodes
- Cluster Interconnects
- Public Network Interfaces
- Multi-Hosted Storage
- Terminal Concentrator
- Administrative Console

Nodes

Nodes

Nodes are the compute platforms of the cluster. Sun Cluster 3.0 supports a maximum of 8 nodes. Sun’s entire Ultra Enterprise server line is supported as a cluster node[1].

A Sun Cluster may consist of heterogeneous nodes, however, nodes should be of similar processing, memory and I/O capability to allow for failovers to occur without significant degradation in performance.

The following platforms are supported as cluster nodes:

- Ultra 2
- Ultra Enterprise 250
- Ultra Enterprise 450
- Ultra Enterprise 3x00, 4x00, 5x00, 6x00
- Ultra Enterprise 10000

[1] Actual platforms supported TBD

Nodes

- Off-the-shelf stand alone server
- Nodes of different types are supported; however, it is highly recommended that nodes have similar I/O and CPU capabilities
- The following platforms are supported:
    - Ultra 2
    - Ultra Enterprise 250
    - Ultra Enterprise 450
    - Ultra Enterprise 3x00, 4x00, 5x00, 6x00
    - Ultra Enterprise 10000

Cluster Interconnect

Cluster Interconnect

One of the basic features of a cluster is a cluster interconnect which allows the nodes to communicate with each other. This cluster interconnect is used to coordinate cluster activities such as membership management, cluster reconfiguration and/or distributed resource management. This interconnect is designed to carry only cluster related traffic and is not designed to carry general application traffic.

Sun Cluster requires that there be at least 2 paths in the cluster interconnect, such that if one fails, the other one can takeover.

An interconnect path consist of 3 main components:

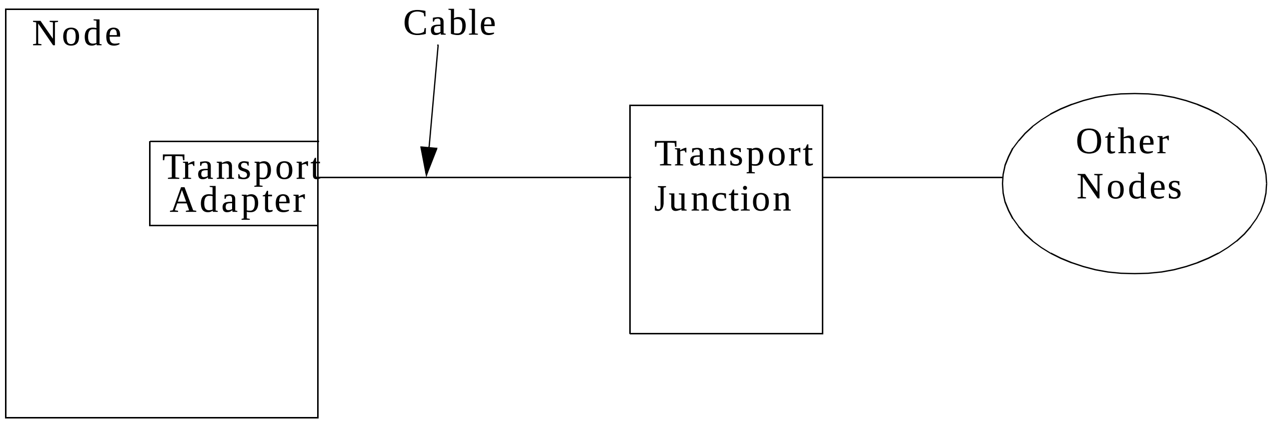
- A transport adapter in the node, such as an Sbus or PCI fast Ethernet card
- A transport junction, such as an Ethernet switch
- A cable connecting the adapter to the junction

The following types of interconnect paths are supported:

- Fast Ethernet
- Gigabit Ethernet

Cluster Interconnect

- The cluster interconnect provides the main communication paths between the nodes in the cluster
- Only cluster-related traffic is allowed to use the cluster interconnects, not general application traffic
- The cluster interconnect must consist of at least 2 paths to protect against a single path failure
- A path consists of three main parts:
  - A node’s transport adapter
  - A transport junction, such as an Ethernet switch (optional on 2 node clusters)
  - A cable connecting the host adapter to a transport junction
- The paths can be Fast Ethernet or Gigabit Ethernet



Public Network Interfaces

Public Network Interfaces

The public network interfaces are the network interfaces utilized by the client systems to access the data and applications housed on the cluster. Sun Cluster supports multiple network interfaces (multi-homed hosts) as well as network adapter failover.

Public Network Interfaces

- ➤ Interfaces used by client systems to access data services housed on the cluster
- ➤ Each node in cluster may have multiple public interfaces (that is, multi-homed hosts)
- ➤ All public networks on the cluster are monitored by the Public Network Management (PNM) subsystem of the cluster framework
- ➤ Redundant interfaces per node may be configured on the same subnet for the Network Adapter Failover (NAFO) feature

Multi-Hosted Storage

Multi-Hosted Storage

A basic feature of a cluster is a highly available cluster-wide file system and device access. The cluster file system is made highly available by utilizing multi-hosted storage. By connecting to multiple hosts, there are multiple paths available to access the data, if one fails, another one is available.

Sun Cluster supports a variety of Sun storage platforms, including the StorEdge A5x00 and StorEdge Multipacks. Each of these (depending on the model) can be physically connected to from two to four hosts.

The GA release supports use of the StorEdge Multipack, the StorEdge D1000 and the StorEdge A5x00 as the cluster's multi-hosted storage.

This release also supports Oracle Parallel Server (OPS) with the A3500 hardware RAID array as storage. No software volume manager is required for this configuration.

OPS relies on fully-connected storage actively used by all cluster nodes at the same time. The model of device primaries used for cluster global devices, Solstice DiskSuite, VxVM, and the cluster file system is not compatible with this OPS requirement. Instead, the local path to fully connected storage is used. To specify the local path, use the `/dev/did/rdisk` device paths as raw devices for OPS.

For more information on installing and configuring OPS, see the Sun Cluster 3.0 Data Services Installation and Configuration Guide.

Multi-Hosted Storage

- ➤ Multi-hosted storage can be physically connected to multiple nodes in the cluster, thus providing highly available access to data
- ➤ Each node physically connected to a storage device provides a path to the storage devices for the rest of the cluster, if one path fails, another one is available
- ➤ Variety of storage platforms supported from the Multipack to the A5x00
- ➤ Only one path to each disk is active at any given time, except when using OPS with A3500 hardware RAID.
- ➤ OPS uses fully-connected storage actively used by all cluster nodes at the same time.

Terminal Concentrator

Terminal Concentrator

The Terminal Concentrator provides an administrative interface for the nodes in the cluster. It basically allows access to ttya on each node in the cluster through a TCP/IP network. This will allow for console level access to each of the cluster nodes from a remote workstation.

The terminal concentrator is an optional component in Sun Cluster 3.0.

Terminal Concentrator

- Provides administrative access to each node in the cluster
- Allows remote access (through TCP/IP) to the console ports (ttya) on each node
- Is optional and is not used for quorum or fencing

Administrative Console

Administrative Console

The administrative console is a remote workstation which can be used to administer the cluster through the Terminal Concentrator. The administrative console can be used to administer and monitor the cluster through the command line interface or through the Sun Management Center console (monitoring only).

Administrative Console

- Workstation used to remotely administrate the cluster through the Terminal Concentrator
- Can be used to run the “Cluster Console” utility and / or the Sun Management Center console

Sun Cluster 3.0 Architecture - Software Components

Now that we have covered the basic components of Sun Cluster 3.0, let's dive into the software architecture of SC 3.0. The software components of Sun Cluster 3.0 are the following:

Application Level Components:

- ➤ Data Services
- ➤ Resource Group Manager (RGM)
- ➤ Cluster API

System Level Components

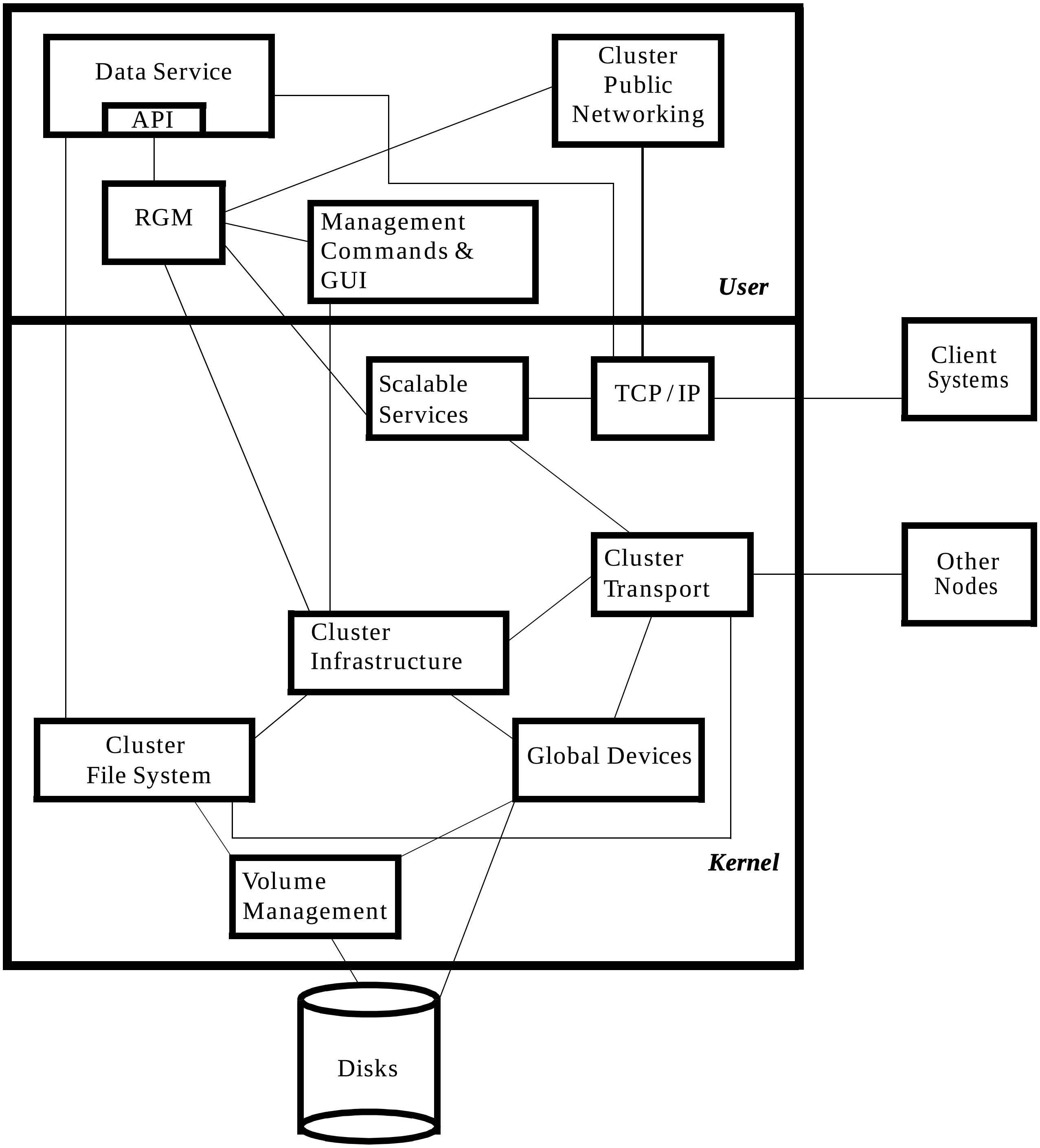
- ➤ Cluster Public Networking
- ➤ Scalable Services
- ➤ Cluster File System
- ➤ Global Devices
- ➤ Cluster Transport
- ➤ Cluster Infrastructure
- ➤ Volume Management

Management Components

- ➤ Cluster Management Commands
- ➤ Cluster Management GUI

The following page illustrates the software architecture of Sun Cluster 3.0.

Sun Cluster 3.0 Software Components



Data Services

Data Services

A data service is simply an application that has been integrated with the cluster. There are two types of data services that can be installed on the cluster: scalable or failover.

A scalable data service is one that can make use of multiple cluster nodes concurrently, while a failover data service is run on a single node, but can be automatically switched to a backup node if the current node fails.

Data Services

- An application that has been integrated with the cluster
- A failover (HA) application, such as a database server or NFS server
- A scalable application, such as a web server or parallel database

Resource Group Manager (RGM)

Resource Group Manager

The Resource Group Manager (RGM) controls the disposition of the failover and scalable data services in the cluster. The RGM is responsible for starting or stopping the data services (or resources) on selected nodes of the cluster in response to cluster membership changes. It is basically the “glue” that takes an “off-the-shelf” data service application and interfaces with the cluster framework.

The RGM works with an entity called a *resource group*. Resource groups are simply a set of *resources*, which are, in turn, instances of *resource types*.

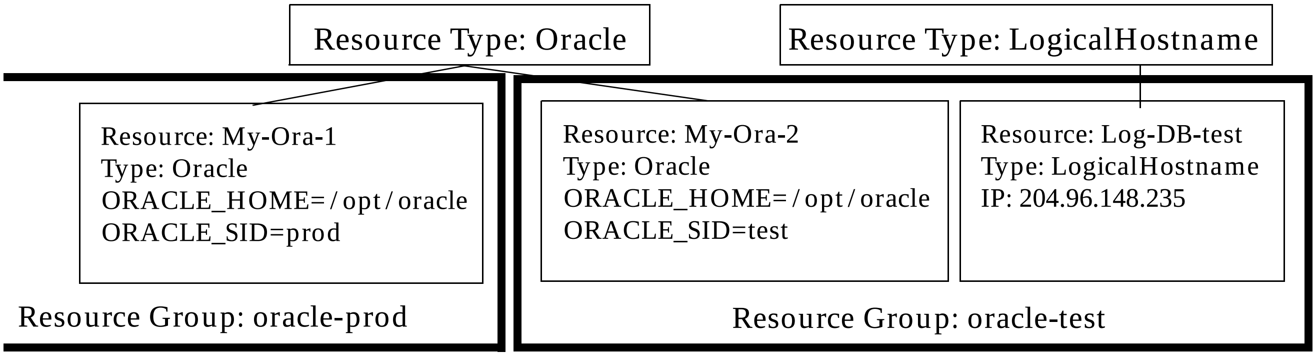
A resource type is simply a collection of elements that describe an application (or data service) to the cluster. It includes information on how the application is to be started and stopped on nodes of the cluster as well as any application specific properties that need to be defined in order to use the application in the cluster. For example, an Oracle resource type may have information on scripts that can be called to startup and shutdown a database instance, as well as declarations of ORACLE_HOME and ORACLE_SID properties. In a resource type, the application specific properties are not assigned values, they are simply declared.

A resource is an instance of a resource type. This will allow multiple instances of an application to be installed on the cluster. When initializing a resource, the application specific properties will be assigned values, and any properties defined on the resource type level are inherited.

Finally, resources are put together into resource groups, which are managed by the RGM. A resource group is simply a set of related or interdependent resources. For example, a resource derived from a LogicalHostname resource type may be placed in the same resource group as a resource derived from an Oracle resource type, thus making the Oracle resource highly available.

Resource Group Manager (RGM)

- Responsible for controlling the starting and stopping of resource groups on selected nodes of the cluster (based on the configuration of each of the resource groups) in response to cluster membership changes.
- A resource group is a collection of related resources, which, in turn, are instances of resource types
- A resource type is a collection of properties that describe a particular application to the cluster framework. Properties may include how to start/stop an application on a node as well as declaration of application specific properties.
- A resource is an instance of a resource type. It has its own set of properties and will inherit any properties set at the resource type level.
- Implemented by the `rgmd` daemon



Cluster API

Cluster API

Sun Cluster 3.0 implements a public API for integrating applications with the cluster framework. The API consists of a set of C libraries and command line utilities (for use in shell scripts) as well as a set of defined *callback methods*, which are executables that are called in a specific order by the cluster framework in response to various cluster events.

Using the API to integrate an application with the cluster framework primarily consists of utilizing the libraries and command line utilities to create the appropriate callback methods. The libraries and command line utilities supply a way for the API programmer to query the state of the cluster or trigger selected cluster events. The set of methods and properties for the application are then registered with the cluster framework as a Resource Type, which can then be used to create Resources for use in Resource Groups.

Cluster API

- A set of C libraries and command line utilities, together with a set of defined callback methods used to integrate an application with the cluster framework
- Callback methods are executables that are called by the cluster framework in a specific order in response to various cluster events
- Integrating an application with the cluster framework consists of creating a set of callback methods for the application (utilizing the API libraries and / or command line utilities to query the state of the cluster and trigger selected cluster events)
- The methods created for an application are registered with the cluster framework as a resource type

Cluster Public Networking

Cluster Public Networking

The cluster public networking component of Sun Cluster provides monitoring and management of the public network interfaces. One feature this subsystem provides is the ability to configure redundant adapters on a node, such that if a network interface fails, network traffic can be switched to a stand-by adapter. This saves the cluster from the expensive and disruptive task of failing over a resource groups and it's associated resources to a backup node if a network adapter fails.

The public networking component of SC 3.0 utilizes an algorithm designed to differentiate between general network failures and interface adapter failures. Cluster public networking is implemented by the pnmd daemon.

Cluster Public Networking

- Monitors the public network interfaces on all nodes of the cluster
- Network interfaces can be configured into Network Adapter Failover (NAFO) groups, allowing a faulty network interface to be failed over to a spare, hot-standby adapter without a full resource groupfailover.
- Detects general network failure versus a failure of a network interface, preventing useless failover
- Implemented by the `pnmd` daemon

Scalable Services

Scalable Services

The scalable services component provides the ability for the *cluster* itself (as opposed to the individual nodes) to have an IP address. Traffic destined for applications registered to work with this cluster-wide (or global) address is distributed to a set of nodes running instances of the application for service. The distribution of traffic is based on a pre-configured load-balancing policy.

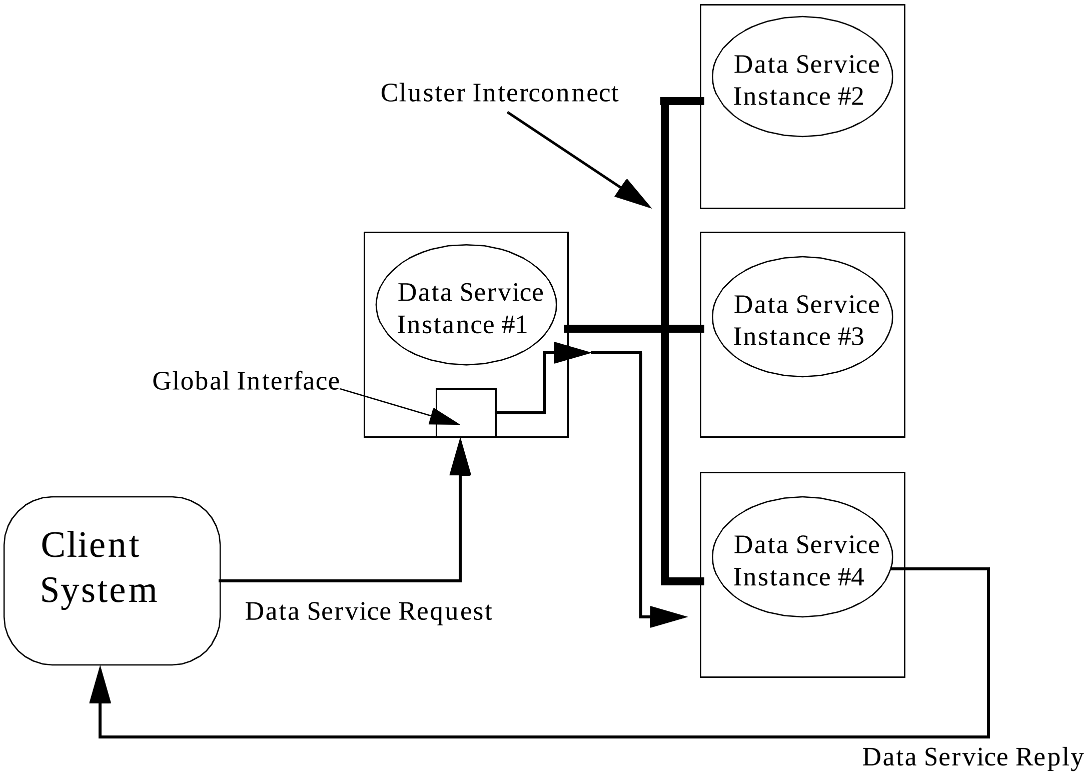
Scalable services works by configuring a shared IP address on a global network interface in one of the nodes of the cluster. All traffic destined for this particular IP is examined by the scalable services subsystem; if the traffic is bound for a data service registered with the global IP address, a distribution table is consulted. Based on the entries in the distribution table, the network traffic is passed to one of the nodes (through the cluster interconnect) in the cluster running an instance of the application that can service the request. Distribution of packets is done in accordance with a configurable load-balancing policy. Sun Cluster 3.0 supports a weighted policy, evenly distributing traffic to all eligible nodes of the cluster.

Not all applications can be made scalable. A true scalable application is one that can be run concurrently (as separate instances) on several nodes of the cluster, with each instance able to independently answer client requests. A scalable data service must meet the following criteria:

- ➤ The service must follow a client/server model -- clients initiate contact with the servers, servers never initiate contact with the client
- ➤ The data service should be stateless, if it is not stateless, the state information must be reliably accessible by all instances of the application. For applications that do maintain locally cached state information (therefore expecting subsequent client connection to arrive at the same server), Sun Cluster can implement a “sticky IP” policy, where packets arriving from a particular IP address will be distributed to the same server.

Scalable Services

- Provides the concept of a global, cluster-wide IP address
- Traffic addressed to this address will be distributed to an eligible node in the cluster based on a load-balancing policy
- Each eligible node will be running an instance of a scalable data service which can process the incoming request
- To be scalable, a data service must be:
    - Client/Server
    - Stateless, or if not stateless, client state information must be sharable
    - Cluster networking is implemented as a resource type, instantiated resources of this type can then be used within resource groups with configured data services



Cluster File System

Cluster File System

The Cluster File System is one of the main features of Sun Cluster 3.0. It provides a cluster-wide, highly available file system with concurrent access by all nodes of the cluster. The cluster file system has the following features:

- ➤ Files can be accessed independent of location, processes on all nodes of the cluster can use the same path to access a particular file. This is very important when configuring failover and scalable data services. A file system is made available to all nodes (even nodes that are not directly attached to the storage devices housing the file system) of the cluster through the cluster transport.
- ➤ When using multi-hosted storage devices, the file system becomes **highly available**, since there are now multiple paths to the data. If a node that is handling a file access request for another node crashes, the request will automatically be retried through another path (another node connected to the storage devices housing the file system). The calling application will not see a failure as long as there is still a path to the data.
- ➤ Coherency protocols are implemented to allow concurrent access from multiple nodes in the cluster
- ➤ The Cluster File System is independent of the underlying file system and volume manager. It will support UFS, VxFS and HSFS file systems as well as Solstice DiskSuite, Sun StorEdge Volume Manager and Cluster Volume Manager (currently, only SDS and VxVM are supported).
- ➤ Can be manipulated using regular Solaris file system semantics - the regular `mount` command is used to mount the filesystem with the options -o global,logging OR -g -o logging. Cluster file systems can also be automatically mounted at boot time.

Cluster File System

- A cluster-wide, highly available filesystem
- Provides transparent access to files regardless of their location - the same path can be used to access a file on every node of the cluster
- When used in conjunction with multi-hosted storage, the cluster file system is highly available. Failures are hidden from the calling application
- Coherency protocols are implemented, allowing concurrent access to a file from all nodes in the cluster
- File system and volume manager independent
- Regular Solaris file system semantics can be used to manipulate the file system (mount / umount)

Global Devices

Global Devices

Global devices access provides cluster-wide device access. Disk, tape and CD-ROM devices attached to a node of the cluster can be accessed from any node of the cluster. The cluster interconnect is used to provide access to devices from a node that is not directly attached to the device.

The cluster automatically assigns unique IDs to each disk, CD-ROM and tape device in the cluster. This allows consistent access to each device from any node in the cluster. The global device namespace is held in the `/dev/global` directory:

Path	Description
`/dev/global/dsk`	Block device files for global disk and CDROM devices
`/dev/global/rdsk`	Raw device files for global disk and CDROM devices
`/dev/global/rmt`	Raw device files for global tape devices

Each node maintains its own set of global device files. The /dev/global directory is actually a symbolic link to `/global/.devices/node@<Node Number>/dev/global`

Disk devices and partitions are represented using a disk ID naming convention of:

```
d<Disk ID Number>s<Slice number>
```

Disks are assigned the unique IDs during a reconfiguration boot of the node.

Global Device Access

- Cluster-wide access to all disks, tape and CD-ROM devices
- Each disk, tape and CD-ROM device is assigned a unique, cluster-wide ID
- Device files are contained in `/dev/global/dsk`, `/dev/global/rdsk` and `/dev/global/rmt`, for block disk devices, raw disk devices and raw tape devices, respectively.
- Disk devices and partitions are represented using a convention of `d<Disk ID number>s<Slice Number>`. For example: `d10s7`
- Disk ID's are assigned during reconfiguration reboots of the nodes of the cluster

Disk ID Driver

Disk ID Driver

The Disk ID (DID) pseudo driver is an integral part of the global device access feature of the cluster. It basically probes all nodes of the cluster and builds a list of unique disk devices, assigning each a unique major and minor number that is consistent on all nodes of the cluster. Access to the global devices is performed utilizing the unique disk ID assigned by the DID driver instead of the traditional `c<x>t<y>d<z>` number.

This approach ensures that any application utilizing the disk devices (such as a volume manager or applications using raw devices) can use a consistent path to access the device. This is especially important for multi-hosted disks, since the local major/minor numbers for each device may vary from node to node, thus changing the `c<x>t<y>d<y>` device naming conventions as well.

The `scdidadm(1M)` command is used to list the DID to local device mappings:

`scdidadm -l`	Lists the mappings of local devices only
`scdidadm -L`	Lists the mappings of devices on **all** nodes of the cluster

Disk ID Driver

- The disk ID (DID) driver will automatically assign unique disk IDs to all disk devices contained in the cluster
- Ensures that each disk device in the cluster has a unique major / minor number that is consistent across all nodes of the cluster. Especially important for multi-hosted disks
    - On node 1, a multi-hosted disk may be device `c1t2d0`
    - On node 2, the same disk may be device `c3t2d0`
    - The DID driver would assign this device a consistent disk ID, e.g. `d10`

Disk ID Driver (Continued)

The following page contains sample output from the `scdidadm (1M)` command.

Disk ID Driver (Continued)

- The scdidadm (1M) command can be used to list the DID to local device mappings:

```
# scdidadm -L
1        venus:/dev/rdsk/c0t0d0         /dev/did/rdsk/d1
2        venus:/dev/rdsk/c1t2d0         /dev/did/rdsk/d2
2        mars:/dev/rdsk/c3t2d0          /dev/did/rdsk/d2
3        venus:/dev/rdsk/c1t3d0         /dev/did/rdsk/d3
3        mars:/dev/rdsk/c3t3d0          /dev/did/rdsk/d3
4        venus:/dev/rdsk/c1t4d0         /dev/did/rdsk/d4
4        mars:/dev/rdsk/c3t4d0          /dev/did/rdsk/d4
5        venus:/dev/rdsk/c1t5d0         /dev/did/rdsk/d5
5        mars:/dev/rdsk/c3t5d0          /dev/did/rdsk/d5
6        venus:/dev/rdsk/c2t2d0         /dev/did/rdsk/d6
6        mars:/dev/rdsk/c4t2d0          /dev/did/rdsk/d6
7        venus:/dev/rdsk/c2t3d0         /dev/did/rdsk/d7
7        mars:/dev/rdsk/c4t3d0          /dev/did/rdsk/d7
8        venus:/dev/rdsk/c2t4d0         /dev/did/rdsk/d8
8        mars:/dev/rdsk/c4t4d0          /dev/did/rdsk/d8
9        venus:/dev/rdsk/c2t5d0         /dev/did/rdsk/d9
9        mars:/dev/rdsk/c4t5d0          /dev/did/rdsk/d9
10       mars:/dev/rdsk/c0t0d0          /dev/did/rdsk/d10
11       mars:/dev/rdsk/c0t6d0          /dev/did/rdsk/d11
# scdidadm -l
1        venus:/dev/rdsk/c0t0d0         /dev/did/rdsk/d1
2        venus:/dev/rdsk/c1t2d0         /dev/did/rdsk/d2
3        venus:/dev/rdsk/c1t3d0         /dev/did/rdsk/d3
4        venus:/dev/rdsk/c1t4d0         /dev/did/rdsk/d4
5        venus:/dev/rdsk/c1t5d0         /dev/did/rdsk/d5
6        venus:/dev/rdsk/c2t2d0         /dev/did/rdsk/d6
7        venus:/dev/rdsk/c2t3d0         /dev/did/rdsk/d7
8        venus:/dev/rdsk/c2t4d0         /dev/did/rdsk/d8
9        venus:/dev/rdsk/c2t5d0         /dev/did/rdsk/d9
```

Cluster Transport

Cluster Transport

The cluster transport is a subsystem that manages the cluster interconnect. It is responsible for configuring and monitoring the state of the interconnect.

The adapters, transport junctions (switches) and cables used by the cluster interconnect must be defined to the cluster transport subsystem. The cluster transport subsystem will configure and monitor the defined adapters, junctions and cables.

Cluster Transport

- Responsible for configuring and monitoring the cluster interconnect hardware
- The transport adapters, transport junctions and cables must be defined to the cluster transport subsystem. This is done during the installation of the cluster software and can be changed after the cluster has been installed
- Provides more than detection of a heartbeat, it also handles “striping” of the interconnect traffic across all enabled paths, as well as other path management tasks.

Volume Management

Volume Management

The volume manager component of Sun Cluster manages the physical disk drives. It provides the following features:

- ➤ Disk Drive Striping and / or Concatenation
- ➤ Disk Mirroring
- ➤ Disk Drive Hot Spares
- ➤ Handling of disk failures and replacement

Disk resources are divided disk sets which are then divided into metadevices. The use of disk mirroring is required for any disk sets used by cluster highly available or scalable data services, this will ensure availability should a disk drive or disk array fail. Metadevices within a disk set can be used as raw devices (for example, for database applications) or as UFS file system devices.

Volume Management

- Provides disk management features:
  - Disk drive striping or concatenation
  - Disk Mirroring
  - Disk Drive Hot Spares
  - Disk failure and replacement handling
- Sun Cluster 3.0 supports both Solstice DiskSuite (SDS) and Veritas Volume Manager (VxVM)
- Disk drives are allocated into disk sets (SDS) or disk groups (VxVM). Within a disk set or disk group, metadevices or volumes are created
- Metadevices or volumes may be used raw or with UFS
- Mirroring is required for the disk resources that will be used by cluster data services

Cluster Infrastructure

Cluster Infrastructure

The cluster infrastructure is the heart of the cluster. Some components of the cluster infrastructure are:

The Cluster Membership Monitor (CMM)

The Cluster Membership Monitor (CMM) monitors the current membership of the cluster. If a membership change is detected, the CMM will drive a synchronized reconfiguration of the cluster, where cluster resources may be redistributed based on the new membership of the cluster.

The Cluster Configuration Repository (CCR)

In order for the nodes to function as a coherently as a cluster, each node must have a consistent view of how the cluster is configured and the current state of the resources managed by the cluster. The Cluster Configuration Repository (CCR) is a private, cluster-wide database which stores information pertaining to the configuration and state of the cluster itself. It provides a consistent (and persistent) database of information that the cluster framework on each node can rely on to provide accurate information about the state of the cluster and its resources.

Cluster Infrastructure

- The Cluster Membership Monitor (CMM)
  - Notified by the transport if a node becomes unresponsive
  - Reaches consensus on changes in cluster membership - if the membership changes the CMM will initiate a reconfiguration sequence, which may result in cluster resources being redistributed among the active nodes in the cluster
- The Cluster Configuration Repository (CCR)
  - A consistent private, cluster-wide database containing cluster configuration and state information
  - A local copy of the CCR is maintained in /etc/cluster/ccr on each node and kept consistent by the cluster framework
  - Ensures that each member node has accurate information about the current state and configuration of the cluster

Management Components

Cluster Management Commands

Primary management of the cluster is performed through the command line. Sun Cluster 3.0 includes a number of commands and utilities that are used to install, configure and administer the cluster:

- ❑ `scinstall`
- ❑ `scsetup`
- ❑ `scconf`
- ❑ `sccheck`
- ❑ `scstat`
- ❑ `scrgadm`
- ❑ `scswitch`
- ❑ `scshutdown`

In addition, a number of Solaris and volume manager (SDS or VxVM) commands are used to administer the cluster nodes and cluster disk resources.

Cluster Management GUI

Sun Cluster 3.0 can be monitored by using Sun Management Center 2.x. Sun Cluster 3.0 provides a SC 3.0 agent module as well as appropriate Sun Management Center server and console add-on packages.

Management Components

- Cluster management commands
  - Sun Cluster 3.0 includes a number of commands and utilities that are used to install, configure and administer the cluster
  - Solaris and volume management (SDS or VxVM) commands and utilities are also used to administer the nodes and disk resources of the cluster.
- Cluster management GUI
  - Sun Management Center 2.x Agent Module and appropriate Sun Management Center server and console add-on packages are provided with Sun Cluster 3.0
  - Cluster console utility, allows simultaneous console (ttya), telnet or rlogin access to all nodes of the cluster. Commands can be typed in once and executed simultaneously on multiple cluster nodes.

Other Cluster Concepts

- **Cluster Topologies**

  The topology of the cluster outlines how the cluster's disk arrays are connected to the cluster nodes

- **Failure Fencing and Quorum**

  The mechanism used to isolate faulty nodes from accessing (and potentially corrupting) data residing on the multi-hosted storage devices

- **Logical Hosts**

  The concept of a logical host is central to configuring highly available data services on the cluster

Other Cluster Concepts

- Cluster Topologies
- Quorum and Failure Fencing

Cluster Topologies

What are topologies?

Topologies are the connection scheme used to connect the cluster nodes to the storage platforms used in the cluster.

Why do we need topologies?

Sun Cluster 3.0 supports the ability to connect up to four servers into a cluster. However, some of Sun's storage platforms support simultaneous connection to a maximum 2 nodes (dual hosting), while others support simultaneous connection to up to 4 nodes. When utilizing 3 or 4 node clusters with storage platforms that can only support dual hosting, a supported topology must be used. Sun supports the following topologies in a cluster:

- Clustered Pairs
- N + 1
- N to N Scalable - See note on following page

Cluster Topologies

- A topology is the way the cluster nodes are physically connected to the storage arrays in the cluster
- Due to differences in the number of hosts a particular storage platform can be connected to versus the number of nodes in a cluster, topologies outline the supported ways a set of storage arrays can be connected to cluster nodes
- The supported topologies are:
  - Clustered Pairs
  - N + 1 Topology
  - N to N Scalable

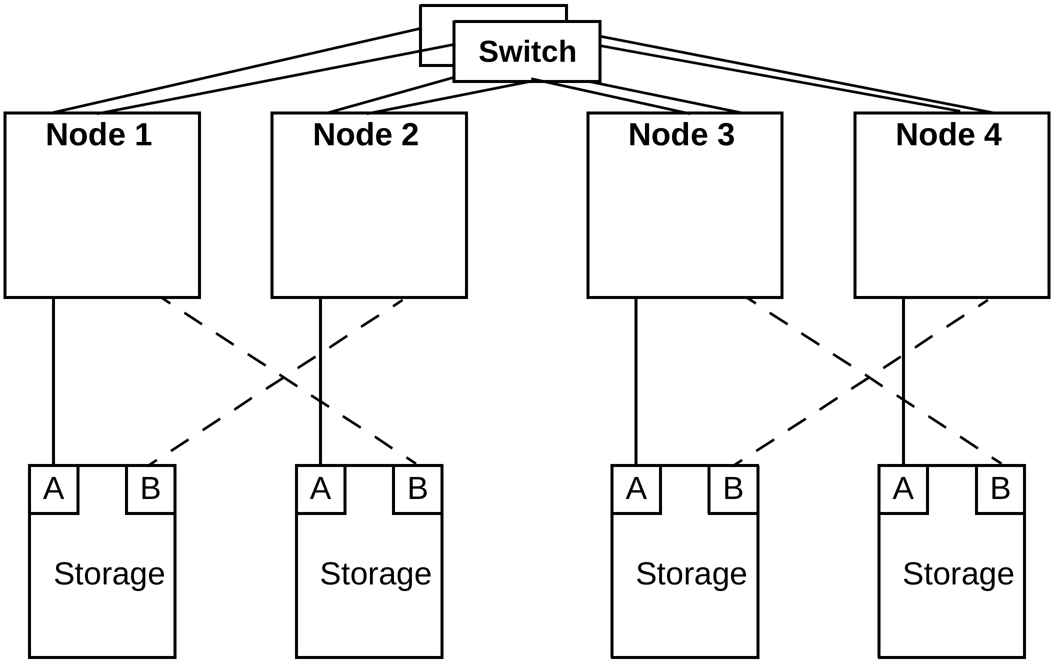
  Note: The N to N Scalable topoplogy is currently NOT supported by Sun Cluster 3.0 due to storage array incompatibility issues (specifically lack of complete SCSI-3 PGR functionality). This topology may be qualified at a later date.
- Topology matters for performance, not functionality

Cluster Topologies

Clustered Pairs

The clustered pair topology is illustrated on the following page.

Clustered Pairs

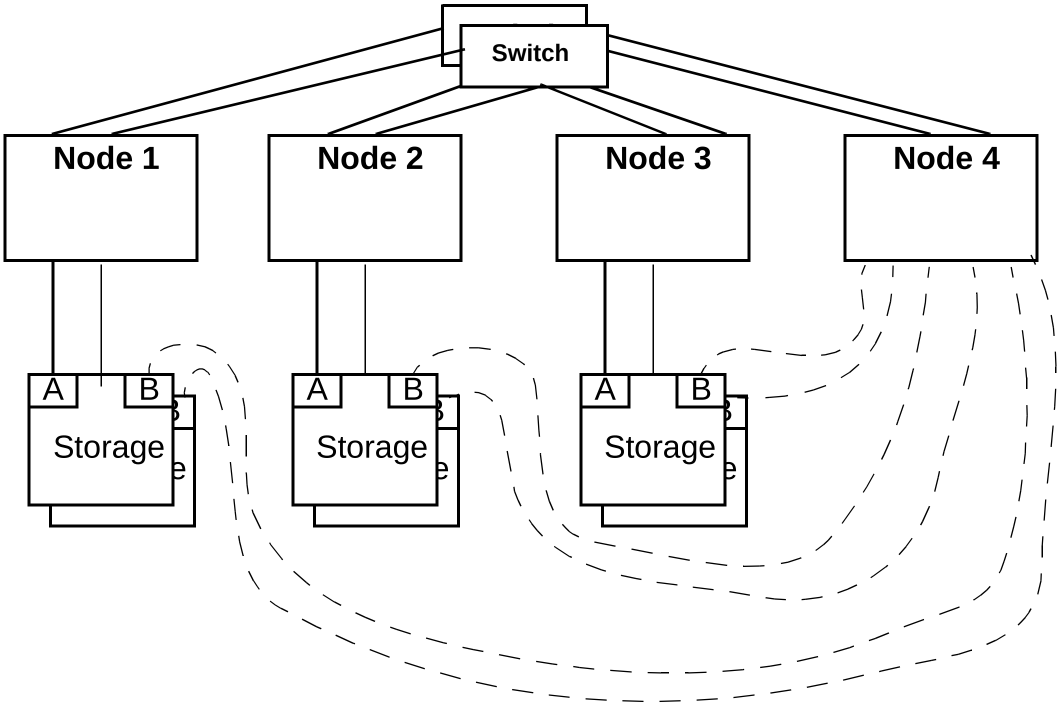


Cluster Topologies

N+1

The N+1 topology is illustrated on the following page.

N + 1 Topology

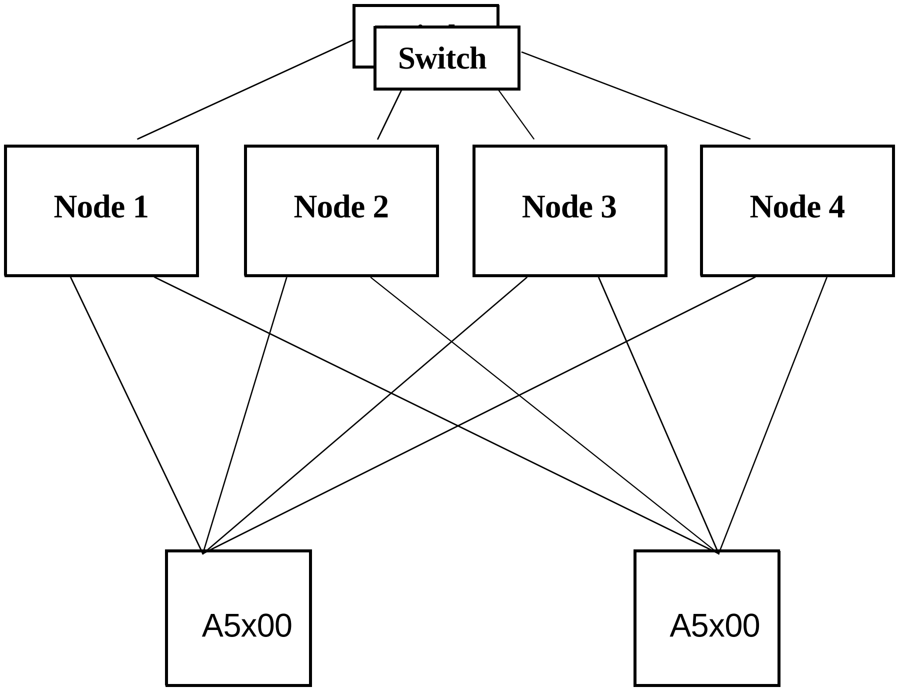

Switch
Node 1
Node 2
Node 3
Node 4
A
B
Storage
A
B
Storage
A
B
Storage

Cluster Topologies

N-to-N or Fully Connected

The N-to-N or Fully Connected topology is illustrated on the following page.

N to N or Fully Connected Topology



Note: The N to N Scalable topoplogy is currently NOT supported by Sun Cluster 3.0 due to storage array incompatibility issues (specifically lack of complete SCSI-3 PGR functionality). This topology may be qualified at a later date.

Failure Fencing and Quorum

What is Quorum and Failure Fencing

The concepts of quorum and failure fencing helps to address the problem of partitioned clusters. Partitioned clusters occur when concurrent multiple failures cause the cluster interconnects to fail.

When a node or set of nodes loses all of its cluster interconnects, the cluster may split into multiple partitions of separate clusters (this is also called a "split brain" partition). This situation can be dangerous since this would allow unsynchronized access to shared disk resources.

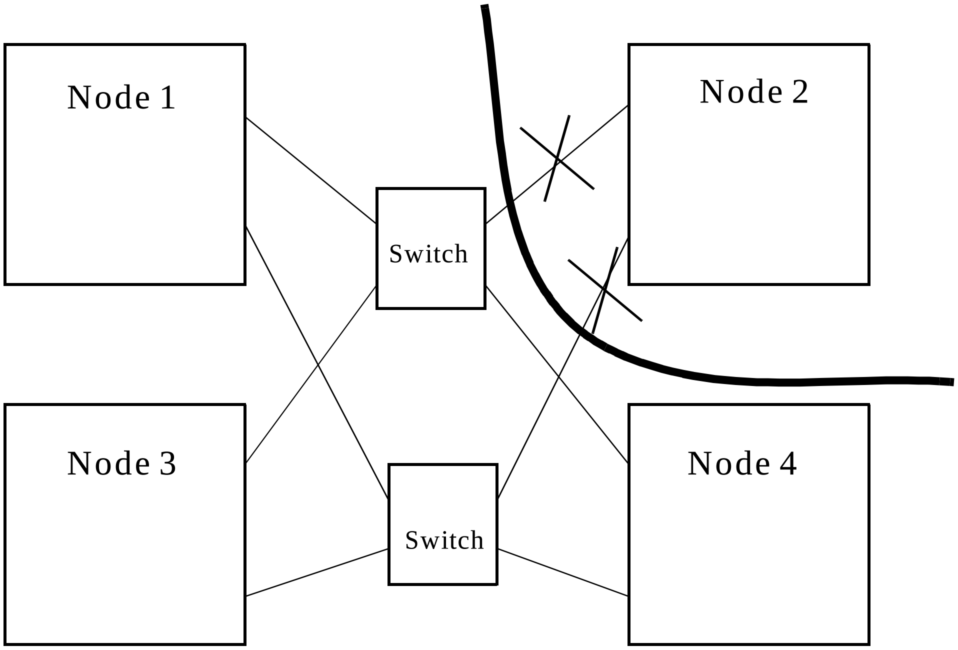
For example, if one of the nodes in the cluster loses its ability to utilize the cluster interconnect (either due to software or hardware failure), it may think that it is the only node in the cluster, and consequently, believe that it has full control of all disk resources. However, with the loss of the cluster interconnect, there is a possibility that the cluster has been partitioned, where there may be two (or sometimes more) sets of nodes competing to be the cluster, both trying to control the disk resources of the cluster without having the ability to coordinate with the other part of the partitioned cluster. To prevent what would be certain data corruption if this scenario occurs, a mechanism to ensure that only one of the "competing" clusters can continue is required.

There are basically two problems that need to be addressed when a cluster becomes partitioned:

1. Ensuring that only one of the partitions is allowed to continue - this is addressed by cluster quorum

2. Once the cluster has determined which nodes are allowed to continue, a mechanism to ensure that the "rogue" nodes cannot access the data housed on the multi-hosted storage must be invoked - this is failure fencing

Failure Fencing and Quorum



Example of a Partitioned Cluster: Node 2 is one partition, Nodes 1,3 and 4 are another partition

- When a node loses access to its cluster interconnect, a partitioned cluster may be formed, where groups of nodes are competing with each other to form a coherent cluster
- Quorum and failure fencing helps prevent the “competing” cluster partitions from causing data corruption due to uncoordinated access to the disk resources of the cluster

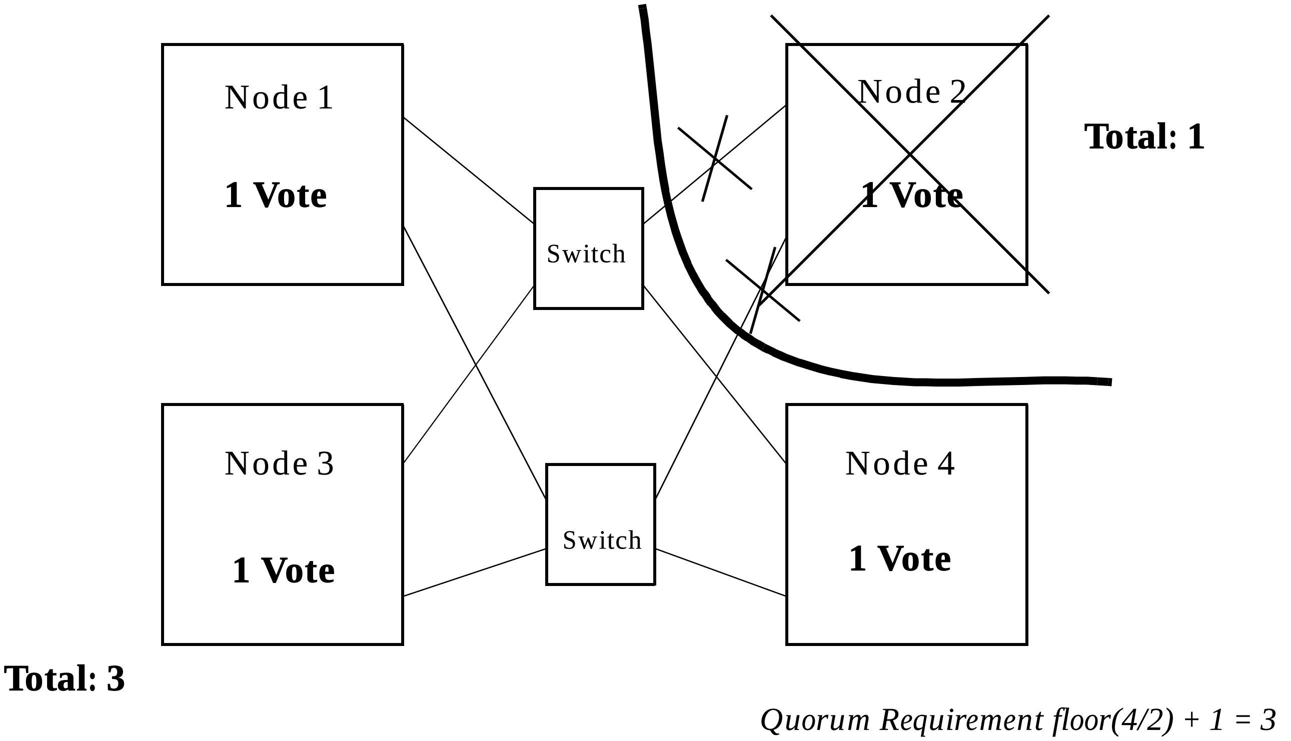
Failure Fencing and Quorum

Cluster Quorum

In order for a set of nodes to consider themselves a cluster, a minimum quorum requirement must be met. Each node in the cluster is assigned 1 quorum vote, with a simple majority (half + 1) constituting a quorum. Therefore, in a 4 node cluster, in order for the cluster to continue, there must be at least 3 nodes present. If the number of nodes drops below 3, the cluster will abort (the nodes will panic themselves out of the cluster).

When a cluster becomes partitioned, the CMMs on each node simply tally votes, counting the number of nodes that can be successfully contacted through the cluster interconnect. If the final tally meets the quorum requirement, the nodes are allowed to continue, if not, the nodes will leave the cluster (through a panic). In our previous partitioned cluster example, node 2 would be forced to leave the cluster while nodes 1, 3 and 4, having control of 3 votes, would be allowed to continue.

Cluster Quorum



- Each node is assigned 1 vote
- Each partition totals the number of votes it has
- Must be at least (Floor(Total Nodes / 2)+ 1) to be allowed to continue
- If quorum is not met, the node will panic, thus leaving the cluster

Failure Fencing and Quorum

Quorum Devices

One problem this algorithm has is that it places a strict requirement on the number of nodes required to have the cluster stay up -- in a 4 node cluster, a minimum of 3 nodes are required, in a 2 node cluster, both nodes must always be up for the cluster to be functional! This is not quite acceptable, especially for HA applications. To handle this problem, Sun Cluster allows the configuration of a ***quorum device***: essentially a disk device (or set of disk devices) that can also provide votes towards the cluster quorum.

Quorum devices are configured with N-1 votes where N is the number of nodes directly attached to the quorum device. A quorum device should be configured between each set of nodes sharing disk devices. For example, in a N-to-N scalable topology with 4 nodes, a single quorum device (with 3 votes) should be configured, in a N+1 topology with 4 nodes, there should be 3 quorum devices configured (each with 1 vote).

The quorum device can be used as a regular data disk in the cluster configuration; it’s duties as a quorum device do not affect the data holding capabilities of the disk.

Quorum Devices

- The requirement that half + 1 of the nodes be available is too strict, especially for a 2 node cluster
- Additional quorum votes can be assigned to the multi-hosted disk devices in the cluster, these are called *quorum devices*
- Quorum devices should be assigned between each set of nodes sharing disk devices
  - For an N-to-N scalable topology, a single quorum device would be assigned
  - For an N+1 topology, 3 quorum devices would be assigned
- A quorum device is assigned N-1 votes, where N is the number of nodes directly connected to the device.
- A device configured as a quorum device is still able to serve as a regular disk, holding data service data

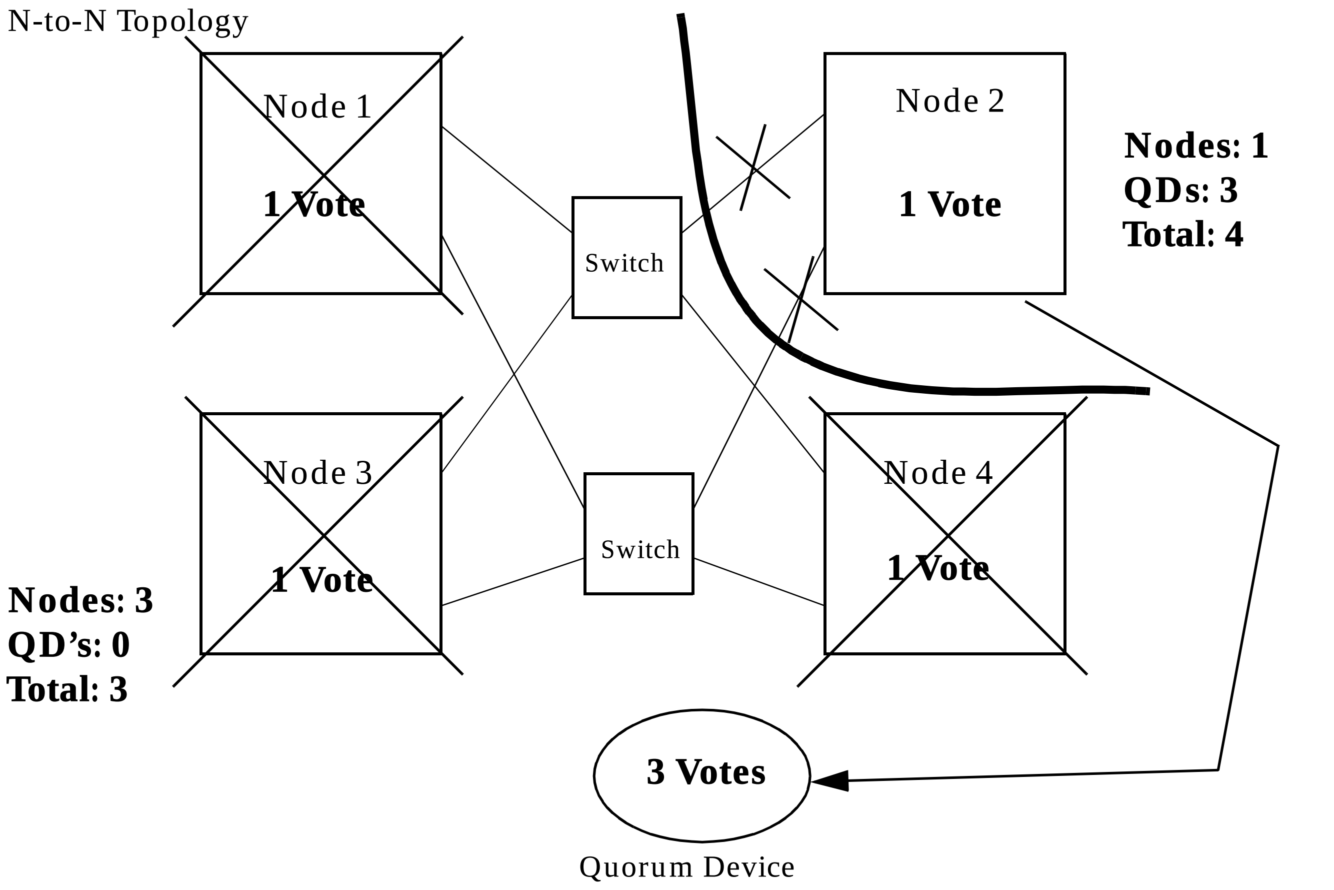
Failure Fencing and Quorum

Quorum Algorithm

Thus, with the use of quorum devices, the algorithm for handling a partitioned cluster is modified as follows:

- ➤ Each cluster partition will race to acquire control of as many quorum devices as it can
- ➤ To acquire control of a quorum device, a cluster partition places a SCSI-3 Persistent Group Reservation (PGR) on the device. This reservation will allow only the members of the partition to access the disk. It is also persistent, surviving reboots of the associated nodes (This feature may NOT be fully supported yet).
- ➤ The quorum requirement is now based on the total number of quorum votes in the cluster, not just number of nodes; thus, in a four node N-to-N Scalable cluster, with a three-vote quorum device, there are a total of seven quorum votes, and four are required to continue. In the previous example, it will come down to who can gain control of the quorum device first; if node 2 wins, it continues while nodes 1, 3 and 4 will be forced to abort. While this outcome may not be desirable, it is still an acceptable alternative to possible data corruption. The cluster can continue as long as there is at least a single node and the quorum device. If the quorum device fails, the cluster should still be able to continue, as long as all four nodes are present.
- ➤ If the node forced to leave the cluster attempts to rejoin the cluster without having its cluster interconnects repaired, it will not be able to acquire a sufficient quorum (it won’t be able to talk to any of the other nodes, and the quorum device is already reserved); thus, it will be prevented from starting up. This ensures that nodes that have been fenced out of the cluster remain out of the cluster until whatever problem causing the cluster partitioning is repaired.

Quorum Algorithm



Quorum Requirement 7/2 + 1 = 4

- Each partition races to reserve the quorum device
- The partition that wins adds the quorum device's votes to its total
- The quorum is adjusted to be a majority of the total votes (nodes + quorum devices)
- The larger partition may not always win

Failure Fencing and Quorum

Amnesia

One other problem that must be addressed by the cluster framework is a partition in time or *amnesia*. While the cluster is operational, all information about the state of the cluster and its data services is held in the CCR. The CCR is, in effect, the long term memory of the cluster. Let's consider the following scenario:

- Two-node cluster with nodes A and B
- Node A leaves the cluster
- Assuming a quorum device is configured, node B is allowed to continue by itself
- In the course of running, node B updates the CCR with new information
- Node B then leaves the cluster, thus aborting the entire cluster
- Node A is started first -- this would start the cluster with a "stale" version of the CCR, since it will have no way of knowing what updates node B made to the CCR while node B was running alone in the cluster

The only way this problem can be alleviated is to make sure that node B is started before node A. Fortunately, the quorum devices can help us here as well:

- When node A leaves the cluster, the only way node B will be allowed to continue is by placing a reservation on the configured quorum device. The reservation is a SCSI-3 ***persistent*** group reservation.
- Since the reservation is persistent (and survives reboots of the node or nodes asserting the reservation), when node A is started, it will notice that there is already a reservation on the quorum device made by node B, thus it will not be able to meet the quorum requirement and, consequently, will not be able to start up until node B is started

Amnesia

- ➤ Amnesia can occur if all the nodes leave the cluster in staggered groups. Example: in a 2 node cluster consisting of nodes A & B, node A leaves, and node B is in the cluster alone, and then sometime later it also leaves. If while node B was in the cluster alone, it happened to update the CCR, node A will have “amnesia”
- ➤ If node A is restarted before node B, it will start the cluster with a stale version of the CCR
- ➤ It must be ensured that node B is started before node A
- ➤ Quorum devices help ensure that node B is always restarted first:
    - ➤ When node A left the cluster, node B placed a SCSI persistent group reservation on the quorum device
    - ➤ The reservation is persistent, surviving reboots of the reserving node or nodes
    - ➤ When starting up, node A will notice the persistent reservation left by node B, and will not be able to acquire the quorum device and therefore will not be able to start up
    - ➤ Once node B starts and forms the cluster, node A will be allowed to join, it’s stale CCR will automatically be updated with node B’s copy of the CCR

CHAPTER 2

Installing Sun Cluster 3.0

Objectives

Purpose

In this chapter we will cover the steps required to install and configure a Sun Cluster 3.0 cluster.

Prerequisites

Knowledge of Solaris system administration

Knowledge of Sun Ultra Enterprise server maintenance

Knowledge of Sun Cluster 3.0 architecture and concepts

Objectives

Upon completion of this chapter, the participant will be able to:

- ➤ Prepare cluster nodes for Sun Cluster 3.0 installation
- ➤ Manually install the Sun Cluster 3.0 software on the cluster nodes
- ➤ Utilize a Jumpstart server to install Sun Cluster 3.0 on the cluster nodes
- ➤ Perform post installation configuration required for Sun Cluster 3.0

Objectives

- Prepare the cluster nodes for Sun Cluster 3.0 installation
- Learn the steps required to manually install the Sun Cluster 3.0 software
- Learn the steps required to configure a Jumpstart installation of Sun Cluster 3.0
- Learn the post-installation configuration tasks required by Sun Cluster 3.0

Installation Overview

Now that we have covered the basic concepts behind Sun Cluster 3.0, let's move into how to actually install and configure the cluster framework. The high level steps are:

1. Verify the cluster's hardware installation
2. Install and configure the administrative console
3. Install and configure the Solaris operating environment
4. Set up proper environment
5. Install Sun Cluster 3.0
6. Complete the cluster initialization
7. Install the volume management software
8. Install Sun Cluster 3.0 data service (agent) software
9. Configure the volume manager
10. Create and mount global file systems
11. Configure PNM
12. Update `ntp.conf` files
13. Verify the installation

Installation Overview

1. Verify the cluster’s hardware installation
2. Install and configure the administrative console
3. Install and configure the Solaris operating environment on the cluster nodes
4. Set up proper environment
5. Install Sun Cluster 3.0
6. Complete the cluster initialization
7. Install the Volume Management software.
8. Install Sun Cluster data service (agent) software
9. Configure the volume manager
10. Create and mount global file systems
11. Configure PNM
12. Update `ntp.conf` files
13. Verify the installation

Step 1 - Verify the cluster's hardware configuration

Before attempting to install Sun Cluster 3.0, verify that the cluster hardware has been configured properly:

- Check that the nodes and storage platforms have been cabled properly, in one of the supported topologies.
- Verify the cabling of the cluster interconnects. Note the following information:
  - Adapters used for the cluster interconnects on each node of the cluster
  - Number of switches used for the cluster interconnects
  - Cable endpoints for the cluster interconnects (e.g. "Cable 1 is from hme1 on node 1 to port 1 of the first Ethernet hub")
- Check the installation of the Terminal Concentrator. Note which Terminal Concentrator serial port each of the nodes is connected to

When checking the hardware, also make sure that there are no single points of failure. For example:

- Make sure that the storage arrays are spread across separate I/O boards, such that the failure of an I/O board does not cause both sides of a mirrored metadevice or volume to fail.
- Along the same line, cluster interconnects should be spread across different I/O boards to prevent a single I/O board failure from causing all cluster interconnects to go down for a node.
- Power to various cluster components should also be distributed. Plugging all cluster components into the same outlet would not be a good idea!

Step 1 - Verify the cluster's hardware configuration

- Check that the storage arrays are cabled properly and conform to one of the supported topologies
- Verify the cabling of the cluster interconnects. The following information should be noted:
  - Which adapters are used on each node
  - The configuration of any switches used
  - The cable endpoints
- Check the installation of the terminal concentrator
  - Note which port on the terminal concentrator each of the nodes is connected to
  - Note the IP address of the terminal concentrator
- Check to make sure there are no "single points of failure"
  - Storage array and cluster interconnect connections should be spread among different I/O boards
  - Power to cluster components should be distributed across separate power sources

Step 2 - Install and configure the administrative console

The administrative console is a workstation that can be used to run the Sun Cluster 3.0 administrative utilities, such as the cluster console or SyMON. The administrative console must have network access to each of the cluster nodes as well as the terminal concentrator. To prepare the administrative console, perform the following steps:

1. Install Solaris 8. At minimum, the End User System Support software group should be installed

2. If the administrative console will be used as the SyMON console or server for the cluster, install the standard SyMON console and/or server packages

3. Using the `pkgadd(1M)` utility, install the `SUNWccon` package from the Sun Cluster 3.0 CD (located in the `SunCluster_3.0/Product` directory)

```
# cd <Location of Sun Cluster 3.0 CD>/SunCluster_3.0/Packages
# pkgadd -d . SUNWccon
```

Step 2 - Install and configure the administrative console

- ➤ Install Solaris 8, End User System Support or above
- ➤ If the admin console will also be used as the SyMON console for the cluster, also install the SyMON console packages
- ➤ Install the `SUNWccon` package from the Sun Cluster 3.0 CD

```
# cd <Location of Sun Cluster 3.0 CD Image>/SunCluster_3.0/Packages
# pkgadd -d . SUNWccon
Processing package instance <SUNWccon> from
</cdrom/suncluster_3_0/SunCluster_3.0/Product>
Sun Cluster Console
(sparc) 3.0.0,REV=1999.10.25.16.28
Copyright 1999 Sun Microsystems, Inc. All rights reserved.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
Installing Sun Cluster Console as <SUNWccon>
## Installing part 1 of 1.
/opt/SUNWcluster/bin/cconsole
/opt/SUNWcluster/bin/ccp
.
... <Various pkgadd messages> ...
.
/opt/SUNWcluster/man/man4/clusters.4
/opt/SUNWcluster/man/man4/serialports.4
[ verifying class <none> ]
Installation of <SUNWccon> was successful.
```

Step 2 - Install and configure the administrative console (Continued)

4. Add an entry for each of the cluster nodes and the terminal concentrator into `/etc/hosts` and/or the appropriate name service (for example, NIS or DNS)

5. Configure the `/etc/clusters` and `/etc/serialports` files

```
Format of /etc/clusters:
   ClusterName   NameofNode1  NameofNode2 ... NameofNode8

Format of /etc/serialports:
   NodeName   TermConcenAddr   PortonTCforNode
```

6. Add `/opt/SUNWcluster/bin` to your PATH

7. Add `/opt/SUNWcluster/man` to your MANPATH

Step 2 - Install and configure the administrative console (Continued)

➤ Add an entry for each of the cluster nodes and the terminal concentrator into the `/etc/hosts` file and/or the appropriate name service (for example, NIS or DNSfor example,)

```
# cat /etc/hosts
127.0.0.1       localhost
204.96.148.238  admin loghost      # Administrative Console
204.96.148.230  tc-planets         # Terminal Concentrator
204.96.148.231  venus              # Node 1 in planets cluster
204.96.148.232  mars               # Node 2 on planets cluster
```

➤ Configure the `/etc/clusters` and `/etc/serialports` files

```
# cat /etc/clusters
planets  venus  mars
# cat /etc/serialports
venus  tc-planets  5002
mars   tc-planets  5003
```

➤ Configure the user environment

```
# PATH=$PATH:/opt/SUNWcluster/bin; export PATH
# MANPATH=$MANPATH:/opt/SUNWcluster/man; export MANPATH
```

Step 2 - Install and configure the administrative console (Continued)

8. Verify the installation by starting the Cluster Control Panel:

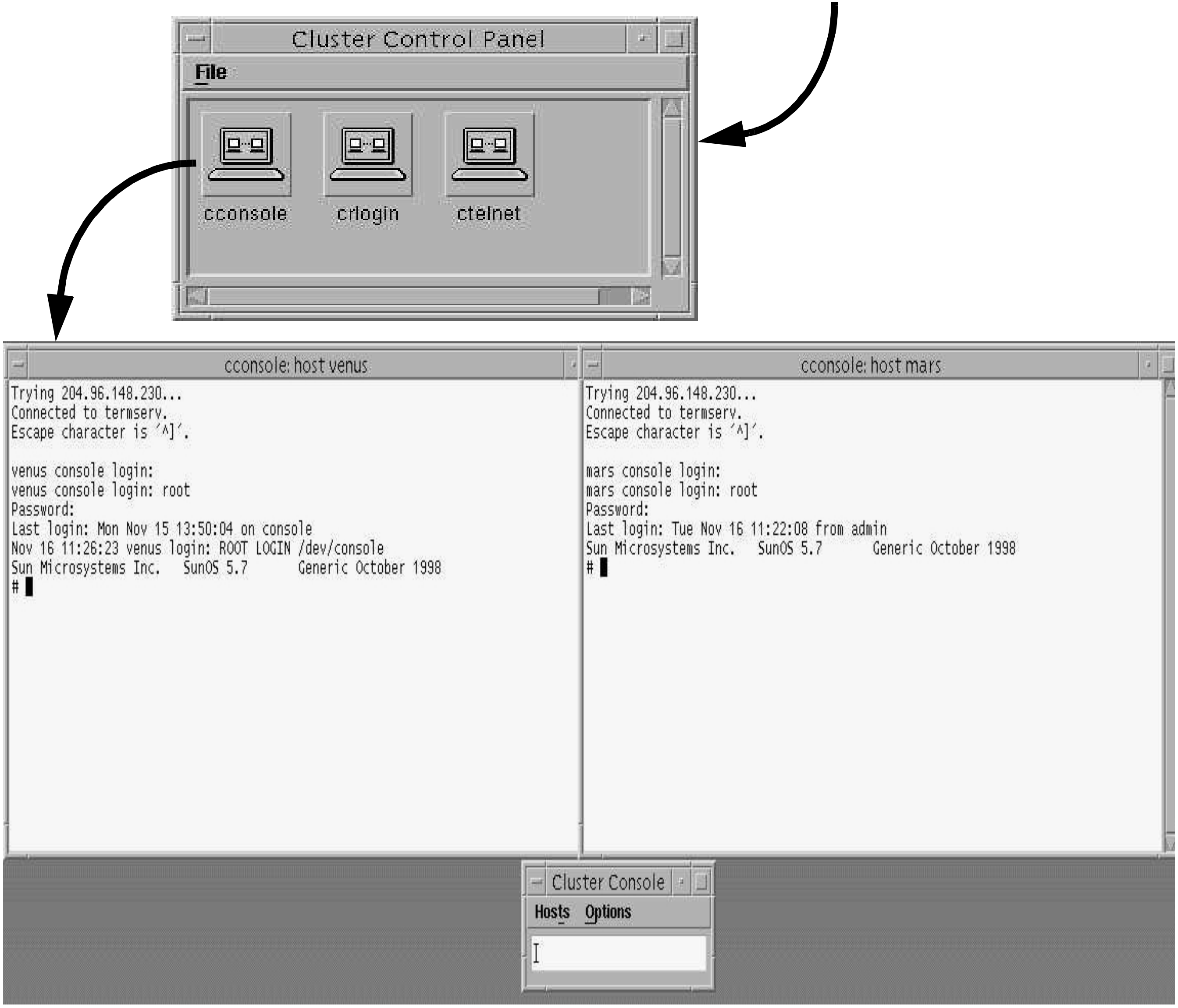
```
# ccp <NameofCluster>&
```

From the cluster control panel, double-click on the cconsole icon. Verify that you are able to communicate via the terminal concentrator with each node of the cluster.

Step 2 - Install and configure the administrative console (Continued)

➤ Verify the installation by starting the cluster control panel:

```
# ccp planets &
```



```
Trying 204.96.148.230...
Connected to termserv.
Escape character is '^]'.

venus console login:
venus console login: root
Password:
Last login: Mon Nov 15 13:50:04 on console
Nov 16 11:26:23 venus login: ROOT LOGIN /dev/console
Sun Microsystems Inc.   SunOS 5.7       Generic October 1998
#
```

```
Trying 204.96.148.230...
Connected to termserv.
Escape character is '^]'.

mars console login:
mars console login: root
Password:
Last login: Tue Nov 16 11:22:08 from admin
Sun Microsystems Inc.   SunOS 5.7       Generic October 1998
#
```

Step 3 - Install and configure the Solaris operating environment

On each node of the cluster, install Solaris 8 as instructed in the Solaris 8 installation documentation. Sun Cluster 3.0 requires Solaris 8. At minimum, the End User System Support software group should be installed. When installing Solaris 8, make sure to set your file system allocations to support Sun Cluster 3.0:

File System	Size
`root (/)`	Allocate all free space to the root partition
`swap`	At least 750MB or double the amount of system RAM, whichever is greater
`/globaldevices`	100MB
`Volume manager slice`	10MB

When prompted whether the system should automatically power off when idle for 30 minutes, make sure to answer “n”.

The `/globaldevices` is simply a placeholder used by `scinstall` to hold the global device namespace. It must be large enough to hold a copy of both the `/dev` and `/devices` directories from the local node.

The volume manager slice can be used by Solstice DiskSuite to hold local metadevice state database replicas or to create a simple `rootdg` disk group for Veritas Volume Manager.

After installing Solaris 8 on all nodes, make sure to install any required OS patches.

Step 3 - Install and configure the Solaris operating environment

- Install Solaris 8 (at minimum, the End User System Support software group) on each node of the cluster
- Make sure to set up file system allocations to support Sun Cluster.
  - `root (/)` - allocate all free space to the root partition
  - `swap` - allocate at least 750MB or double the amount of RAM, whichever is greater
  - `/globaldevices` - 100MB (for global device namespace)
  - Volume manager slice (no mountpoint) - 10MB (for local replicas or simple rootdg)
- Install any required OS patches

Step 4 - Set up the proper environment

Before installing the cluster software, set up appropriate software directory paths on each node. The following entries should be included in `root's` PATH:

Path	Description
`/usr/bin, /usr/sbin, /sbin`	Standard paths to Solaris executables
`/usr/cluster/bin`	Sun Cluster 3.0 command line utilities
`/opt/VRTSvmsa/bin` `/opt/VRTSvxvm/bin`	Veritas Volume Manager command line and GUI utilities (if using Veritas Volume Manager)

Also, set the MANPATH variable to include `/usr/cluster/man` and `/opt/VRTSvxvm/man`.

Export the MANPATH.

Step 4 - Set up the proper environment

- ➤ Set the PATH variable to include `/sbin`, `/usr/sbin`, `/usr/bin`, `/usr/cluster/bin` and `/opt/VRTSvxvm/bin` and `/opt/VRTSvmsa/bin` (for Veritas Volume Manager based clusters)
- ➤ Set the MANPATH variable to include `/usr/cluster/man` and `/opt/VRTSvxvm/man` (for VxVM)
- ➤ Export the MANPATH

Step 5 - Install Sun Cluster 3.0

From the Sun Cluster 3.0 CD-ROM, use the `scinstall` command to install the Sun Cluster software on the cluster nodes and instantiate the node in the cluster.

`scinstall` can either be run as an interactive application, where it will present you with menus and prompts, or as a command line utility. The steps to install Sun Cluster on the nodes are:

1. Change to the `Sun_Cluster_3.0/Tools` subdirectory of the Sun Cluster 3.0 CD-ROM image:

```
# cd <Location of SC 3.0 CD-ROM image>/Sun_Cluster_3.0/Tools
```

2. To run `scinstall` interactively, on the **first** node of the cluster, invoke `scinstall` with no arguments:

```
# ./scinstall
```

3. Choose option 1 at the Main Menu
4. Provide appropriate answers to the installation scripts prompts
5. Upon completion, `scinstall` will install the appropriate packages and prepare the node to become a member of the cluster
6. Repeat steps 1-4 on the remaining nodes of the cluster (using option 2 on the Main Menu)

Step 5 - Install Sun Cluster 3.0

- Locate the `Sun_Cluster_3.0/Tools` directory on the Sun Cluster 3.0 CD-ROM

- On the first node of the cluster, run the `scinstall(1M)` command with no arguments (for an interactive installation):

```
# cd <Location of SC 3.0 CD Image>/Sun_Cluster_3.0/Tools
# ./scinstall
```

- Choose option 1 from the Main Menu and confirm your choice (the * to the left of an option indicates a valid option):

```
  *** Main Menu ***

    Please select from one of the following (*) options:

      * 1) Establish a new cluster using this machine as the first node
      * 2) Add this machine as a node in an established cluster
        3) Configure a cluster to be JumpStarted from this install server
        4) Add support for a new data service to this cluster node
        5) Print release information for this cluster node

      * ?) Help with menu options
      * e) Exit

    Option:  1
```

```
  *** Establishing a new cluster ***

    This option is used to establish a new cluster using this machine as
    the first node in that cluster. You will be asked to provide both the
    name of the cluster and the names of the other nodes which will
    initially be joining that cluster.

    In addition, you will be asked to provide certain cluster transport
    configuration information.

    Press Ctrl-d at any time to return to the Main Menu.

    Do you want to continue (yes/no) [yes]? yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

- ➤ Set the name of the cluster
- ➤ Set the names of the other cluster members

Step 5 - Sun Cluster 3.0 Installation (Continued)

- Provide the name of the cluster:

```
>>> Cluster Name <<<

  Each cluster has a name assigned to it. The name can be made up of
  any characters other than whitespace. It may be up to 256 characters
  in length. And, you may want to assign a cluster name which will be
  the same as one of the failover logical host names in the cluster.
  Create each cluster name to be unique within the namespace of your
  enterprise.

  What is the name of the cluster you want to establish? planets
```

- Provide the name of the other nodes that will be configured in the cluster:

```
>>> Cluster Nodes <<<

  This release of Sun Cluster supports a total of up to 8 nodes.

  Please list the names of the other nodes planned for the initial
  cluster configuration. List one node name per line. When finished,
  type Control-D:

  Node name:  mars
  Node name (Ctrl-D to finish):  ^D


  This is the complete list of nodes:

      venus
      mars

  Is it correct (yes/no) [yes]?  yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ Select DES authentication option

Step 5 - Sun Cluster 3.0 Installation (Continued)

- The cluster can be configured to use DES authentication to authenticate nodes that attempt to add themselves to the cluster (default is to use no authentication):

```
   >>> Authenticating Requests to Add Nodes <<<

    Once the first node establishes itself as a single node cluster,
    other nodes attempting to add themselves to the cluster configuration
    must be found on the list of nodes you just provided. The list can be
    modified once the cluster has been established using scconf(1M), or
    other tools.

    By default, nodes are not securely authenticated as they attempt to
    add themselves to the cluster configuration. This is generally
    considered adequate, since nodes which are not physically connected
    to the private cluster interconnect will never be able to actually
    join the cluster. However, DES authentication is available. If DES
    authentication is selected, you must configure all necessary
    encryption keys before any node can join (see keyserv(1M),
    publickey(4)).

    Do you need to use DES authentication (yes/no) [no]?  no
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ Set the private cluster transport address and netmask

Step 5 - Install Sun Cluster 3.0 (Continued)

- ➤ Select the network address used for the private cluster transport. `yes` for the default of 172.16.0.0, `no` to select a new address:

```
  >>> Network Address for the Cluster Transport <<<

    The private cluster transport uses a default network address of
    172.16.0.0. But, if this network address is already in use elsewhere
    within your enterprise, you may need to select another address from
    the range of recommended private addresses (see RFC 1597 for
    details).

    If you do select another network address, please bear in mind that
    the Sun Clustering software requires that the rightmost two octets
    always be zero.

    The default netmask is 255.255.0.0; you may select another netmask,
    as long as it minimally masks all bits given in the network address
    and does not contain any “holes”.

    Is it okay to accept the default network address (yes/no) [yes]?  yes

    Is it okay to accept the default netmask (yes/no) [yes]?  yes
```

If you want to provide a non-default private cluster transport network address and/or netmask, type `no` at the prompts and type in a new network address in the format XXX.YYY.0.0 (last two octets must be 0) and/or netmask:

```
    Is it okay to accept the default network address (yes/no) [yes]?  no

    What network address do you want to use?   192.168.0.0

    Is it okay to accept the default netmask (yes/no) [yes]?  no

    What netmask do you want to use [255.255.0.0]?  255.255.0.0
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

- ➤ If the cluster is a two-node cluster, the installation script asks if the private cluster transport utilizes transport junctions (Ethernet switches). 2 node clusters can be connected in a point-to-point configuration using crossover Ethernet cables.
- ➤ If the cluster consists of more than three nodes, the installation script assumes that transport junctions are being used and automatically continue to the next section

Step 5 - Install Sun Cluster 3.0 (Continued)

- If the cluster is a two node cluster, `scinstall` will ask if transport junctions (Ethernet switches) are being used in the private cluster transport:

```
  >>> Point-to-Point Cables <<<

    The two nodes of a two-node cluster may use a directly-connected
    interconnect. That is, no cluster transport junctions are configured.
    However, when there are greater than two nodes, this interactive form
    of scinstall assumes that there will be exactly two cluster transport
    junctions.

    Does this two-node cluster use transport junctions (yes/no) [yes]?  yes
```

- If the cluster consists of more than two nodes, `scinstall` will assume that transport junctions are being used and will automatically continue to the transport junction definition section:

```
  >>> Point-to-Point Cables <<<

    The two nodes of a two-node cluster may use a directly-connected
    interconnect. That is, no cluster transport junctions are configured.
    However, when there are greater than two nodes, this interactive form
    of scinstall assumes that there will be exactly two cluster transport
    junctions.

    Since this is not a two-node cluster, you will be asked to configure
    two transport junctions.


Hit ENTER to continue:
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ If transport junctions are to be configured, `scinstall` will now ask you to name the transport junctions. If the cluster is a two node directly-connected cluster, this section will be skipped.

Step 5 - Install Sun Cluster 3.0 (Continued)

- If transport junctions are being configured, `scinstall` will ask for the names of the transport junctions:

```
>>> Cluster Transport Junctions <<<

  Note that interactive scinstall assumes that all transport junctions
  are of type "switch" and that they accept no special properties
  settings. If the junctions which you are using do not fall into this
  category, you may need to use non-interactive scinstall by specifying
  a complete set of command line options. For more information, refer
  to the scconf_transp_jct family of man pages (e.g.,
  scconf_transp_jct_etherhub(1M)).

  What is the name of the first junction in the cluster [switch1]?  switch1

  What is the name of the second junction in the cluster [switch2]?  switch1
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ Configure the transport adapters to be used for the private cluster transports. `scinstall` will ask for the following information:

- ❏ The device name of each of the host adapters to be used for the cluster transport
- ❏ If transport junctions are being used in the cluster, `scinstall` will ask for the name of the transport junction each adapter is connected to. Also you may optionally name the port on the transport junction each transport adapter is cabled to
- ❏ If transport junctions are not being used (as in a directly connected two-node cluster), `scinstall` will ask for the transport adapters on the other node to which the local transport adapters are cabled to

Step 5 - Install Sun Cluster 3.0 (Continued)

- Next, `scinstall` will ask for information about the transport adapters being used for the cluster transport:

```
>>> Cluster Transport Adapters and Cables <<<

  You must configure at least two connection points to the private
  cluster transport for each node in the cluster. More than two
  connection points are allowed, but this interactive form of scinstall
  assumes exactly two.

  Note that interactive scinstall does not allow you to specify any
  special transport adapter properties settings. If your adapters have
  special properties which must be set, you may need to use
  non-interactive scinstall by specifying a complete set of command
  line options. For more information, please refer to the man pages for
  your adapters in the scconf_transp_adap family of man pages (e.g.,
  scconf_transp_adap_hme(1M)).

  What is the name of the first cluster transport adapter ?  hme1

  All transport adapters support the "dlpi" transport type. Ethernet
  adapters are supported only with the "dlpi" transport; however, other
  adapter types may support other types of transport. For more
  information on which transports are supported with which adapters,
  please refer to the scconf_transp_adap family of man pages
  (scconf_transp_adap_hme(1M), ...).

  Adapter "hme1" is an Ethernet adapter.
  The "dlpi" transport type will be set for this cluster.
  Name of the junction to which "hme1" is connected [switch1]?  switch1

  Each adapter is cabled to a particular port on a transport junction.
  And, each port is assigned a name. You may explicitly assign a name
  to each port. Or, for certain junction types, you may allow scinstall
  to assign a default name for you. The default port name assignment
  sets the name to the node number of the node hosting the transport
  adapter at the other end of the cable.

  Please remember that some types of cluster transport junctions are
  not compatible with the default port naming assignments. For more
  information regarding port naming requirements, refer to the
  scconf_transp_jct family of man pages (e.g.,
  scconf_transp_jct_dolphinswitch(1M)).

  Okay to use the default for the "hme1" connection [yes]?  yes
  What is the name of the second cluster transport adapter [hme2]?  hme2
  Name of the junction to which "hme2" is connected [switch2]?  switch2
  Use the default port for the "hme2" connection [yes]? yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ A file system is used by SC 3.0 to house the global device namespace. This filesystem should be at least 100MB in size and must be completely empty. By default, `scinstall` will use a file system called `/globaldevices`. Another mounted file system may be used, as long as its empty and at least 100MB, or `scinstall` can build a new file system on a given raw disk partition

Step 5 - Install Sun Cluster 3.0 (Continued)

- Configure the file system or disk partition to be used for the global device namespace:

```
>>> Global Devices File System <<<

  Each node in the cluster must have a local file system mounted on
  /global/.devices/node@<nodeID> before it can successfully participate
  as a cluster member. Since the “nodeID” is not assigned until
  scinstall is run, scinstall will set this up for you. However, in
  order to do this, you must supply the name of either an
  already-mounted file system or raw disk partition at this time. This
  file system or partition should be at least 100 MB in size.

  If an already-mounted file system is used, the file system must be
  empty. If a raw disk partition is used, a new file system will be
  created for you.

  The default is to use /globaldevices.

  Is it okay to use this default (yes/no) [yes]?  yes
```

- If you do not want to use the default (the `/globaldevices` filesystem), you may optionally supply the name of another mounted file system (it must be empty and at least 100MB) or a raw disk partition (must be at least 100MB, all existing data will be lost):

```
  Is it okay to use this default (yes/no) [yes]?  no
  Do you want to use an already existing file system (yes/no) [yes]?  yes
  What is the name of the file system  /sparefs
```

```
  Is it okay to use this default (yes/no) [yes]?  no
  Do you want to use an already existing file system (yes/no) [yes]?  no
  What is the name of the disk partition you want to use /dev/dsk/c0t0d0s4
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

- ➤ Configure whether `scinstall` should automatically reboot the node after performing the SC 3.0 installation
- ➤ `scinstall` will echo the non-interactive command line that is going to be executed and ask for a final confirmation before actually installing and configuring the appropriate packages. If automatic reboot was chosen, the node will reboot itself into the cluster

Step 5 - Install Sun Cluster 3.0 (Continued)

- Configure whether scinstall should automatically reboot the node after completing the installation and configuration of the cluster packages:

```
  >>> Automatic Re-boot <<<

    Once scinstall has successfully installed and initialized the Sun
    Cluster software for this machine, it will be necessary to re-boot.
    After the re-boot, this machine will be established as the first node
    in the new cluster.

    Do you want scinstall to re-boot for you (yes/no) [yes]?  yes
```

- Final confirmation and package installation and configuration:

```
  >>> Confirmation <<<

    Your responses indicate the following options to scinstall:

      scinstall -i \
           -C planets \
           -N venus \
           -T node=venus,node=mars,authtype=sys \
           -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2 \
           -B type=switch,name=switch1 -B type=switch,name=switch2 \
           -m endpoint=:hme1,endpoint=switch1 \
           -m endpoint=:hme2,endpoint=switch2

    Are these the options you want to use [yes]? yes

    Do you want to continue with the install (yes/no) [yes]? yes

** Installing Sun Cluster 3.0 **
        SUNWscr.....done.
        SUNWscdev...done.
        SUNWscu.....done.
...
... < Various installation messages edited out > ...
...
Rebooting ...
```

Step 5 - Install Sun Cluster 3.0 (Continued)

Once the first node has been installed, the remaining nodes can be installed (actually, all nodes can be installed simultaneously, the additional nodes will wait for their “sponsoring” node (usually the first node) to become active before completing the installation)

To install the other nodes in the cluster:

1. Change to the `Sun_Cluster_3.0/Tools` subdirectory of the Sun Cluster 3.0 CD-ROM image:

```
# cd <Location of SC 3.0 CD-ROM image>/Sun_Cluster_3.0/Tools
```

2. To run `scinstall` interactively, on the **first** node of the cluster, invoke `scinstall` with no arguments:

```
# ./scinstall
```

3. Choose option 2 at the Main Menu

4. Provide appropriate answers to the installation script prompts

Step 5 - Install Sun Cluster 3.0 (Continued)

➤ On the remaining nodes of the cluster, run the `scinstall(1M)` command with no arguments (for an interactive installation):

```
# cd <Location of SC 3.0 CD Image>/Sun_Cluster_3.0/Tools
# ./scinstall
```

➤ Choose option 2 from the Main Menu and confirm your choice (the * to the left of an option indicates a valid option):

```
  *** Main Menu ***

    Please select from one of the following (*) options:

      * 1) Establish a new cluster using this machine as the first node
      * 2) Add this machine as a node in an established cluster
        3) Configure a cluster to be JumpStarted from this install server
        4) Add support for a new data service to this cluster node
        5) Print release information for this cluster node

      * ?) Help with menu options
      * e) Exit

    Option:  2
```

```
  *** Adding a node to an established cluster ***

    This option is used to add this machine as a node in an
    already-established cluster. You will be asked to provide both the
    name of the cluster and the name of one of the nodes already in the
    cluster. If this “sponsoring node” is not already a member of the
    cluster, scinstall will wait for up to 24 hours for it to join.

    In addition, you will be asked to provide certain cluster transport
    configuration information.

    Press Ctrl-d at any time to return to the Main Menu.

    Do you want to continue (yes/no) [yes]?  yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

➤ Specify a sponsoring node - this can be any node that is already a member of the cluster. On new cluster installs, this is usually the first node

➤ Specify the name of the cluster this node should become a part of

➤ Specify whether transport junctions are used in the cluster

Step 5 - Install Sun Cluster 3.0 (Continued)

- Specify the sponsoring node and cluster name and cluster size:

```
  >>> Sponsoring Node <<<

    For any machine to join a cluster, it must identify a node in that
    cluster willing to “sponsor” its membership in the cluster. When
    configuring a new cluster, this “sponsor” node is typically the first
    node used to build the new cluster. However, if the cluster is
    already-established, the “sponsoring” node can be any node in that
    cluster.

    Already-established clusters can keep a list of hosts which are able
    to configure themselves as new cluster members. This machine should
    be in the join list of any cluster which it tries to join. If the
    list does not include this machine, you may need to add it using
    scconf(1M), or other tools.

    And, if the target cluster uses DES to authenticate new machines
    attempting to configure themselves as new cluster members, the
    necessary encryption keys must be configured before any attempt to
    join.

    What is the name of the sponsoring node?  venus
```

```
  >>> Cluster Name <<<

    Each cluster has a name assigned to it. When adding a node to the
    cluster, you must identify the name of the cluster you are attempting
    to join. A sanity check is performed later to verify that the
    “sponsoring” node is a member of that cluster.

    What is the name of the cluster you want to join ?  planets
```

```
  >>> Point-to-Point Cables <<<

    The two nodes of a two-node cluster may use a directly-connected
    interconnect. That is, no cluster transport junctions are configured.
    However, when there are greater than two nodes, this interactive form
    of scinstall assumes that there will be exactly two cluster transport
    junctions.

    Is this a two-node cluster (yes/no) [yes]?  yes

    Does this two-node cluster use transport junctions (yes/no) [yes]?  yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

The following page continues the example interactive Sun Cluster 3.0 installation using `scinstall`.

- ➤ `scinstall` will then proceed to ask for the following information:
  - ❑ Names of the transport junctions (if transport junctions are being used)
  - ❑ Names and types of the transport adapters in the node
  - ❑ If transport junctions are being used, the ports each transport adapter is cabled to, or if this is a 2 node point-to-point cluster, the transport adapters on the other node each local transport adapter is connected to
  - ❑ The target location for the global device namespace
  - ❑ Whether the node should automatically reboot itself following installation and configuration by `scinstall`
- ➤ After final confirmation of the installation parameters, scinstall will proceed to install the appropriate packages and prepare the node to become a member of the cluster. On the next reboot, the node will become a member of the cluster

Step 5 - Install Sun Cluster 3.0 (Continued)

➤ Configure cluster transport and global device namespace options and complete `scinstall` (Help text and extra blank lines have been edited out):

```
>>> Cluster Transport Junctions <<<

  What is the name of the first junction in the cluster [switch1]?  switch1
  What is the name of the second junction in the cluster [switch2]?  switch2

>>> Cluster Transport Adapters and Cables <<<

  What is the name of the first cluster transport adapter ?  hme1
  Is “hme1” an ethernet adapter [yes]?  yes
  The “dlpi” transport type will be set for this cluster.
  Name of the junction to which “hme1” is connected [switch1]?  switch1
  Okay to use the default for the “hme1” connection [yes]?  yes
  What is the name of the second cluster transport adapter ?  hme2
  Name of the junction to which “hme2” is connected [switch2]?  switch2
  Use the default port for the “hme2” connection [yes]?   yes

>>> Global Devices File System <<<

  The default is to use /globaldevices.
  Is it okay to use this default (yes/no) [yes]?  no
  Do you want to use an already existing file system (yes/no) [yes]?  no
  What is the name of the disk partition you want to use  /dev/dsk/c0t0d0s4

>>> Automatic Re-boot <<<

  Do you want scinstall to re-boot for you (yes/no) [yes]?  yes

>>> Confirmation <<<

  Your responses indicate the following options to scinstall:
    scinstall -i \
         -C planets \
         -N venus \
         -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2 \
         -m endpoint=:hme1,endpoint=switch1 \
         -m endpoint=:hme2,endpoint=switch2

  Are these the options you want to use [yes]?   yes
  Do you want to continue with the install (yes/no) [yes]? yes
```

Step 5 - Install Sun Cluster 3.0 (Continued)

You may also run scinstall non-interactively, using command line arguments to install Sun Cluster 3.0.

➤ To install a cluster node from the command line, invoke `scinstall` with the appropriate arguments. This command can be run in parallel on multiple cluster nodes (see the `scinstall(1M)` man page for a complete list and description of command line arguments):

```
./scinstall -i -C <ClusterName> -F -N <NodeName>                \
   -T node=<nodename>,node=<nodename>,...,authtype=<Authtype>\
   -A <FirstAdapter> -A <SecondAdapter>                      \
   -B <FirstJunct> -B <SecondJunct>                          \
   -m endpoint=[<node>]:<FirstAdapter>,                      \
      endpoint=<FirstJunct>[@port]                           \
   -m endpoint=[<node>]:<SecondAdapter>,                     \
      endpoint=<SecondJunct>[@port]

  -i Indicates install mode
  -C Denotes the name of the cluster to be installed
  -F Denotes that this is the first node of the cluster
  -N Denotes the name of the first node of the cluster
  -T Denotes the authentication options for the cluster.
     This argument is only valid on the first node being
     installed in the cluster. Only listed nodes can be
     added to the cluster, Authtype can be sys or des, des
     authorization will also require that each listed node have
     a key entry in the publickey database
  -A Denotes the devices to be used as cluster transport
     adapters
  -B Denotes the names of the cluster transport junctions
  -m Denotes the endpoints of the cluster transport cables.
     2 endpoints must be given for each cable.  If installing a
     2 node cluster using a direct connect cables (i.e. no
     transport junction), use the following syntax for the -m
     argument:
     -m endpoint=[<node>]:<adapter>,endpoint=<node>:adapter
```

`scinstall` will add the appropriate packages and configure the node to be a member of the cluster. `scinstall` will trigger a reconfiguration reboot of the node.

Step 5 - Install Sun Cluster 3.0 (Continued)

➤ To install the nodes non-interactively, run the `scinstall(1M)` command, providing the appropriate information for the cluster name and cluster transports:

```
# ./scinstall -i -C planets -F -N venus \
-T node=venus,node=mars,authtype=sys  \
-A hme1 -A hme2 -B switch1 -B switch2 \
-m endpoint=:hme1,endpoint=switch1    \
-m endpoint=:hme2,endpoint=switch2

** Installing SunCluster 3.0 **

   SUNWscr.....done.
   SUNWscdev...done.
   SUNWscu.....done.
   SUNWscman...done.
   SUNWscsal...done.
   SUNWscsam...done.
   SUNWrsmop...done.
   SUNWsci.....done.
   SUNWscid....done.
   SUNWscidx...done.
   SUNWscvm....done.
   SUNWmdm.....done.

Initializing cluster name to “planets” ... done
Initializing authentication options ... done
Initializing configuration for adapter “hme1” ... done
Initializing configuration for adapter “hme2” ... done
Initializing configuration for junction “switch1” ... done
Initializing configuration for junction “switch2” ... done
Initializing configuration for cable ... done
Initializing configuration for cable ... done

Setting the node ID for “venus” ... done (id=1)

<... Various other installation messages edited out ...>

Rebooting ...
```

Step 6 - Complete the Cluster Initialization

After the initial install of Sun Cluster 3.0, only node 1 will be configured with a quorum vote (this is called “installmode” and is needed to allow the first node to exist in the cluster alone). After completing the installation on all nodes in the cluster, the quorum vote for each node must be set to one, in addition, you may optionally assign quorum votes to shared storage devices in the cluster. If this step is skipped, *the cluster will not survive the loss of node 1*.

The following configuration guidelines must be followed when configuring shared quorum devices (that is, assigning quorum votes to disk devices residing in the multi-hosted storage of the cluster):

- Configuring at least one shared quorum device is required in a two node cluster, it is optional in cluster consisting of more than 2 nodes
- Even in cluster consisting of more than 2 nodes, if **any** shared quorum devices are configured, each node pair (or node set if a fully connected topology is used) with shared storage devices **must** have at least 1 quorum device configured between them
- Multiple quorum devices can be configured between node pairs or sets, however, if creating more than 1 shared quorum device between a pair or set of nodes, always create an **odd number** of shared quorum devices, each in different disk array enclosures.

Step 6 - Complete the Cluster Initialization

- After the initial installation of Sun Cluster 3.0, only node 1 will have a quorum vote, all other nodes will have a vote of 0. In order for the failure fencing features of the cluster to work correctly, the quorum votes for the nodes and any shared quorum devices must be configured properly
- Each node needs to have 1 quorum vote
- Shared quorum devices can be configured per the following guidelines:
  - A shared quorum device is a disk device in one of the multi-hosted storage arrays which has been assigned quorum votes
  - In a 2 node cluster, at least one shared quorum device is required, in a cluster consisting of more than 2 nodes, shared quorum devices are optional
  - Even in greater than 2 node clusters, if **any** shared quorum devices are configured then all node pairs (or node sets if using a fully connected topology) **must** have at least one shared quorum device configured.
  - Multiple quorum devices may be configured between node pairs or sets, however, the number of quorum devices defined between node pairs or sets should always be an **odd number**
- The `scsetup` or `scconf` command is used to configure quorum

Step 6 - Complete the Cluster Initialization (Continued)

To set the proper quorum votes in the cluster, use the scsetup utility. This should be run from only one node in the cluster (can be run from any node of the cluster):

```
# /usr/cluster/bin/scsetup
```

Step 6 - Complete the Cluster Initialization (Continued)

➤ Using the `scsetup` utility to configure quorum:

```
# /usr/cluster/bin/scsetup
  >>> Initial Cluster Setup <<<

    This program has detected that the cluster "installmode" attribute is ...
    ... <Help text edited out to save space > ...

    Please do not proceed if any additional nodes have yet to join the
    cluster.

    Is it okay to continue (yes/no) [yes]?  yes

    Do you want to add any quorum disks (yes/no) [yes]?  yes

    Dual-ported SCSI-2 disks may be used as quorum devices in two-node ...
    ... <Help text edited out to save space > ...

    Which global device do you want to use (d<N>)?  d2

    Is it okay to proceed with the update (yes/no) [yes]?  yes

scconf -a -q globaldev=d2

    ...<If this command is run on the console you may see console messages here>...

    Command completed successfully.

Hit ENTER to continue:

    Do you want to add another quorum disk (yes/no)?  no

    Once the "installmode" property has been reset, this program will
    skip "Initial Cluster Setup" each time it is run again in the future.
    However, quorum devices can always be added to the cluster using the
    regular menu options. Resetting this property fully activates quorum
    settings and is necessary for the normal and safe operation of the
    cluster.

    Is it okay to reset "installmode" (yes/no) [yes]?  yes

scconf -c -q reset

    ...<If this command is run on the console you may see console messages here>...

    Cluster initialization is complete.
    Hit ENTER to proceed to the main menu:
```

Step 6 - Complete the Cluster Initialization (Continued)

Alternatively, the scconf command can be used to setup the proper quorum votes on each node and configure shared quorum devices:

```
scconf -a -q globaldev=<GlobalDevice>

-a - specifies the “add” form of the command
-q - specifies that quorum devices are going to be added
globaldev=<GlobalDevice> - specifies the global device (dN) to be
                           used as a shared quorum device
```

```
scconf -c -q reset

-c - specifies the “change” form of the command
-q - specifies that quorum options are going to be changed
node=<nodename>,reset - resets the quorum vote to 1 for <nodename>
```

To list the status of the cluster quorum, use the `scstat(1M)` command:

```
scstat -q
```

Step 6 - Complete the Cluster Initialization (Continued)

- The `scconf(1M)` command can also be used to configure shared quorum devices and reset the vote count for each node:

```
# /usr/cluster/bin/scconf -a -q globaldev=d2
```

```
# /usr/cluster/bin/scconf -c -q reset
```

- The `scstat(1M)` command can be used to check the quorum status:

```
# /usr/cluster/bin/scstat -q
Quorum
  Current Votes:                                   3
  Votes Configured:                                3
  Votes Needed:                                    2
    Node Quorum
      Node Name:                                   venus
        Votes Configured:                          1
        Votes Contributed:                         1
        Status:                                    Online

      Node Name:                                   mars
        Votes Configured:                          1
        Votes Contributed:                         1
        Status:                                    Online

    Device Quorum
    Quorum Device Name:                            /dev/did/rdsk/d2s2
    Votes Configured:                              1
    Votes Contributed:                             1
    Nodes Having Access:
      venus                                        Enabled
      mars                                         Enabled
    Owner Node:                                    venus
    Status:                                        Online
```

Step 7 - Install the volume management software

Once we have completed cluster initialization, install the packages required by the volume management software (SDS or VxVM).

Refer to Chapter 3: *Configuring Solstice DiskSuite with Sun Cluster 3.0* and Chapter 4: *Configuring Veritas Volume Manager with Sun Cluster 3.0* or the *Sun Cluster 3.0 Installation Guide* for more detailed information on installing the volume manager software.

Step 7 - Install the volume management software

➤ Install the volume management software packages

For Solstice DiskSuite:

```
# cd <Location of SDS 4.2.1 CD Image>/products/DiskSuite_4.2.1/sparc
# pkgadd -d . SUNWmdu SUNWmdr SUNWmdx
```

Edit the /kernel/drv/md.conf file

For Veritas Volume Manager:

```
# cd <Location of VxVM 3.0.4 CD Image>/pkgs
# pkgadd -d . VRTSvxvm VRTSvmman VRTSvmdev VRTSvmsa
```

Install VxVM licenses, if required

➤ Refer to chapters 3 and 4 of this manual or the *Sun Cluster 3.0 Installation Guide* for more details regarding this topic

Step 8 - Install Sun Cluster data service software

Now that the cluster framework has been installed on all nodes of the cluster, the cluster data service framework (resource types and methods) for selected data services must be installed on all nodes of the cluster.

`scinstall` (when invoked with no arguments) can be used to interactively install the appropriate SC 3.0 data service packages:

```
# /usr/cluster/bin/scinstall
```

Step 8 - Install Sun Cluster data service software

➤ Install the appropriate Sun Cluster data services from the Sun Cluster 3.0 Data Services CD-ROM. Use the `scinstall` to interactively specify the services to be installed:

```
# /usr/cluster/bin/scinstall
  *** Main Menu ***

    Please select from one of the following (*) options:

        1) Establish a new cluster using this machine as the first node
        2) Add this machine as a node in an established cluster
        3) Configure a cluster to be JumpStarted from this install server
      * 4) Add support for a new data service to this cluster node
      * 5) Print release information for this cluster node

      * ?) Help with menu options
      * e) Exit

    Option:  4
```

```
  *** Adding data service software ***

    This option is used to add support for a data service to a node
    already configured as a Sun Cluster cluster node.

    You will be asked to supply both the location of the media and the
    identifier for the data service you want to install.

    Where is the data services CD?  /cdrom/scdataservices_3_0
```

Step 8 - Install Sun Cluster data service software (Continued)

The following page continues the interactive installation example for installing SC 3.0 data service packages.

- Select the appropriate data service package to install
- After installation of the chosen data service package, `scinstall` will return to the Main Menu
- Alternatively, you may also invoke `scinstall` with the appropriate arguements to non-interactively install an SC 3.0 data service:

```
scinstall -i -k -s <ServiceName> -d <SC3.0 DataSvcs CD Location>

-i - Indicates an installation
-k - Specifies that the base Sun Cluster 3.0 packages will
     not be installed
-s - Specifies the names of the services to be installed.
     Multiple service names can be specified in a comma
     separated list.
-d - Specifies the path the the Sun Cluster 3.0 Data Service
     CD Image
```

The service names for the available data services are listed below:

Service Name	Description
`apache`	Sun Cluster Apache Web Server components
`dns`	Sun Cluster HA DNS data service components
`nfs`	Sun Cluster HA NFS data service components
`iws`	Sun Cluster Netscape Web Server components
`nsldap`	Sun Cluster HA LDAP data service components
`oracle`	Sun Cluster HA Oracle data service components

Step 8 - Install Sun Cluster data service software

➤ Add choose a data service package to install:

```
    This is the list of data services on this CD:

        Identifier    Description

        apache        Sun Cluster - Highly Available Apache
        dns           Sun Cluster - Highly Available Domain Name Server
        iws           Sun Cluster - Highly Available iPlanet Web Server
        nsldap        Sun Cluster - Highly Available Netscape Directory Server
        nfs           Sun Cluster - Highly Available NFS Server
        oracle        Sun Cluster - Highly Available Oracle DBMS

    Please list all the data services you want to add. List one data
    service identifier per line. When finished, type Control-D:

    Data service identifier (Ctrl-D to finish): apache
    Data service identifier (Ctrl-D to finish): dns
    Data service identifier (Ctrl-D to finish): nfs
    Data service identifier (Ctrl-D to finish): iws
    Data service identifier (Ctrl-D to finish): nsldap
    Data service identifier (Ctrl-D to finish): oracle
    Data service identifier (Ctrl-D to finish): ^D

    This is the complete list of data services:

        apache
        dns
        nfs
        iws
        nsldap
        oracle

    Is it correct (yes/no) [yes]? yes
    Is it okay to add the software for this data service package [yes]?
```

➤ Non-Interactive Installation:

```
# cd /usr/cluster/bin
# ./scinstall -ik -s oracle,nshttp,nfs -d <SC3.0 Data Services CD location>
```

Step 9 - Configure the volume manager

Initialize and configure the volume manager entities.

For SDS:

1. Initialize the local metadevice state databases
2. Create disksets to house data service data
3. Add drives to each diskset
4. Partition disks in the disksets
5. Create the metadevices for each diskset

For VxVM:

1. Initialize the `rootdg` disk group
2. Create disk groups to house data service data
3. Create mirrored volumes in each data service disk group
4. Register the disk groups with the Sun Cluster 3.0 framework

Refer to Chapter 3: *Configuring Solstice DiskSuite with Sun Cluster 3.0* and Chapter 4: *Configuring Veritas Volume Manager with Sun Cluster 3.0* or the *Sun Cluster 3.0 Installation Guide* for more detailed information on initializing and configuring the appropriate volume manager software.

Step 9 - Configure the volume manager

- Initialize and configure the volume manager (SDS or VxVM) that will be used

  For SDS:

  1.Initialize the local metadevice state databases

  2.Create disksets to house data service data

  3.Add drives to each diskset

  4.Partition disks in the disksets

  5.Create the metadevices for each diskset

  For VxVM:

  1.Initialize the `rootdg` disk group

  2.Create disk groups to house data service data

  3.Create mirrored volumes in each data service disk group

  4.Register the disk groups with the Sun Cluster 3.0 framework

- Refer to chapters 3 & 4 of this manual or the *Sun Cluster 3.0 Installation Guide* for more details regarding this topic

Step 10 - Create and mount global filesystems

Create any global file systems required by the cluster's data services on the diskset metadevices or disk group volumes created in Step 9:

1. Use `newfs` (1M) to create filesystems on the appropriate metadevice(s) or volume(s).

```
newfs <raw metadevice file>
```

2. Create appropriate mount points on **each** node of the cluster. The global file systems require a mount point on each node of the cluster, regardless of the disk topology. It is recommended that the global mount points be placed under the `/global` directory:

```
mkdir /global/<MountPoint>
```

3. Update the `/etc/vfstab` files on the nodes which are directly connected to the disks in the disksets.

   All cluster file systems must be mounted using the `global` and `syncdir` mount options. The `global` option will make the file system available globally, throughout the cluster while the `syncdir` option will ensure that changes to file system directory information are always performed synchrounously

   Previous versions of the product ignored the "mount at boot" column. In Sun Cluster 3.0, you must enter "yes" in this column for the file systems to be mounted automatically by the clustering software.

Step 10 - Create and mount global filesystems

- ➤ Use newfs to create filesystems on the appropriate diskset metadevices

```
# newfs /dev/md/web_data_1/rdsk/d10
newfs: construct a new file system /dev/md/web_data_1/rdsk/d10: (y/n)? y
/dev/md/web_data_1/rdsk/d10:    4149600 sectors in 2730 cylinders of 19 tracks, 80
sectors
        2026.2MB in 86 cyl groups (32 c/g, 23.75MB/g, 5888 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 48752, 97472, 146192, 194912, 243632, 292352, 341072, 389792, 438512,
 487232, 535952, 584672, 633392, 682112, 730832, 779552, 828272, 876992,
 925712, 974432, 1023152, 1071872, 1120592, 1169312, 1218032, 1266752, 1315472,
 1364192, 1412912, 1461632, 1510352, 1556512, 1605232, 1653952, 1702672,
 1751392, 1800112, 1848832, 1897552, 1946272, 1994992, 2043712, 2092432,
 2141152, 2189872, 2238592, 2287312, 2336032, 2384752, 2433472, 2482192,
 2530912, 2579632, 2628352, 2677072, 2725792, 2774512, 2823232, 2871952,
 2920672, 2969392, 3018112, 3066832, 3112992, 3161712, 3210432, 3259152,
 3307872, 3356592, 3405312, 3454032, 3502752, 3551472, 3600192, 3648912,
 3697632, 3746352, 3795072, 3843792, 3892512, 3941232, 3989952, 4038672,
 4087392, 4136112,
#
```

- ➤ Create appropriate mount points on **each** node of the cluster

```
# mkdir /global/web-data
```

- ➤ Update the `/etc/vfstab` file on all nodes that are directly connected to the disks used in the disksets

```
#device     device       mount    FS      fsck    mount     mount
#to mount   to fsck      point    type    pass    at boot   options
#
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s1       -       -       swap    -       no      -
/dev/dsk/c0t0d0s0       /dev/rdsk/c0t0d0s0      /       ufs     1       no
-
swap    -       /tmp    tmpfs   -       yes     -
/dev/md/web_data_1/dsk/d10  /dev/md/web_data_1/rdsk/d10  /global/web-data  ufs  2   \
yes global,syncdir **
**The logging option must be added if this is not a trans device. For trans devices
SDS logging is automatically enabled.**
```

Step 11 - Configure PNM

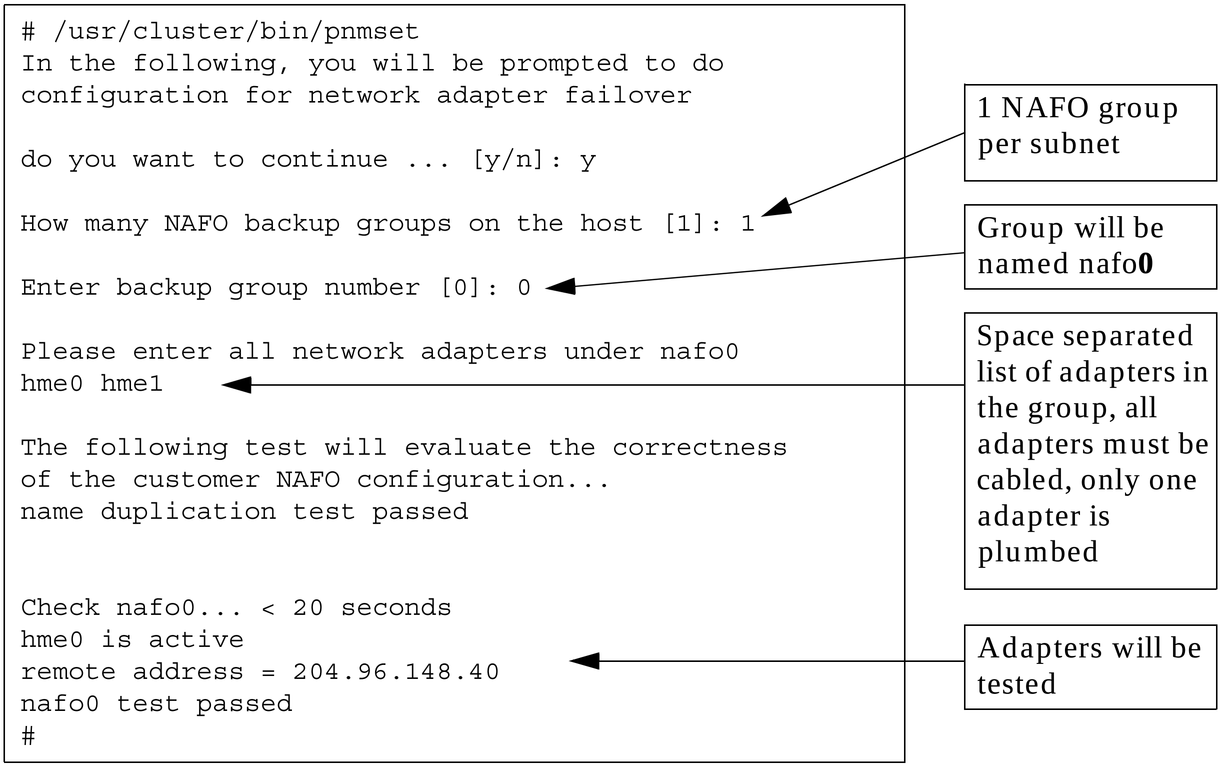
All network interfaces that are to be used by client systems to access the logical host's data services must be placed under PNM control. This is accomplished by using the `pnmset` command.

PNM status can be checked using the `pnmstat` command:

```
pnmstat -l

-l list mode - lists all NAFO groups and the current status
```

Step 11 - Configure PNM

- Use the `pnmset` command to configure PNM:



```
# /usr/cluster/bin/pnmset
In the following, you will be prompted to do
configuration for network adapter failover

do you want to continue ... [y/n]: y

How many NAFO backup groups on the host [1]: 1

Enter backup group number [0]: 0

Please enter all network adapters under nafo0
hme0 hme1

The following test will evaluate the correctness
of the customer NAFO configuration...
name duplication test passed


Check nafo0... < 20 seconds
hme0 is active
remote address = 204.96.148.40
nafo0 test passed
#
```

1 NAFO group per subnet

Group will be named nafo**0**

Space separated list of adapters in the group, all adapters must be cabled, only one adapter is plumbed

Adapters will be tested

- Create one NAFO group per public subnet on each node of the cluster

- NAFO groups may consist of just a single adapter

- All adapters in a group must be cabled properly, however, only one adapter should be plumbed and active

- Use the `pnmstat -l` command to check the status of the configured NAFO groups

```
# /usr/cluster/bin/pnmstat -l
bkggrp   r_adp     status   fo_time   live_adp
nafo0    hme0:hme1 OK       NEVER     hme0
```

Step 12 - Update `ntp.conf` files

On each node of the cluster, the `/etc/inet/ntp.conf` file should be updated. Remove all entries for private hostnames that are not being used by the cluster. Also, if you have changed the private hostnames of the cluster nodes, update this file accordingly.

You may also make other modifications to meet your NTP requirements.

Step 12 - Update `ntp.conf` files

- Remove any entries for private hostnames that are not configured in the cluster

```
...
peer clusternode1-priv prefer
peer clusternode2-priv
peer clusternode3-priv
peer clusternode4-priv
peer clusternode5-priv
peer clusternode6-priv
peer clusternode7-priv
peer clusternode8-priv
...
```

On a four node cluster using the default private hostnames, these entries should be removed

- If the private hostnames for the nodes in the cluster are changed (using `scconf -a -P` or `scsetup`), this file must be updated to reflect the proper private hostnames.
- Make any other modifications to meet your NTP requirements

Step 13 - Verify the installation

Use the `scconf(1M)` and `scstat(1M)` utilities to verify the installation of the cluster.

- ➤ `scconf -p` can be run to print the configuration information for the cluster

Step 13 - Verify the installation

- Use `scconf -p` command to list the cluster configuration:

```
Cluster name:                                      planets
Cluster ID:                                        0x3831F35A
Cluster install mode:                              disabled
Cluster private net:                               172.16.0.0
Cluster private netmask:                           255.255.0.0
Add node auth type:                                unix
Add node auth list:                                venus mars
Cluster nodes:                                     venus mars

Cluster node name:                                 venus
  Node ID:                                         1
  Node enabled:                                    yes
  Node private hostname:                           clusternode1-priv
  Node quorum vote count:                          1
  Node reservation key:                            0x3831F35A00000001
  Node transport adapters:                         hme1 hme2

  Node transport adapter:                          hme1
    Adapter enabled:                               yes
    Adapter transport type:                        dlpi
    Adapter property:                              device_name=hme
    Adapter property:                              device_instance=1
    Adapter property:                              dlpi_heartbeat_timeout=10000
    Adapter property:                              dlpi_heartbeat_quantum=2000
    Adapter property:                              nw_bandwidth=80
    Adapter property:                              bandwidth=10
    Adapter property:                              netmask=255.255.255.128
    Adapter property:                              ip_address=172.16.0.129
    Adapter port names:                            0

    Adapter port:                                  0
      Port enabled:                                yes

  Node transport adapter:                          hme2
    Adapter enabled:                               yes
    Adapter transport type:                        dlpi
    Adapter property:                              device_name=hme
    Adapter property:                              device_instance=2
    Adapter property:                              dlpi_heartbeat_timeout=10000
    Adapter property:                              dlpi_heartbeat_quantum=2000
    Adapter property:                              nw_bandwidth=80
    Adapter property:                              bandwidth=10
    Adapter property:                              netmask=255.255.255.128
    Adapter property:                              ip_address=172.16.1.1
    Adapter port names:                            0
...<Output truncated to save space>...
```

Step 13 - Verify the installation (Continued)

- ➤ Use scstat -p to print the current status of cluster components

Step 13 - Verify the Installation (Continued)

- Use `scstat -p` to print the status of cluster components

```
Node
  Node Name:                                       venus
  Status:                                          Online

  Node Name:                                       mars
  Status:                                          Online

Path
  venus:hme2 - mars:hme1                           Path online
  venus:hme1 - mars:hme0                           Path online

Device Group
  Device Group Name:                               dsk/d1
  Status:                                          Offline
  Primary:
  Secondary:
  Spare:
  Inactive:
  Transition:

...<Some Device Group Data edited out to save space > ...

  Device Group Name:                               web_data_1
  Status:                                          Offline
  Primary:
  Secondary:
  Spare:
  Inactive:
  Transition:

Quorum
  Current Votes:                                   3
  Votes Configured:                                3
  Votes Needed:                                    2
    Node Quorum
      Node Name:                                   venus
        Votes Configured:                          1
        Votes Contributed:                         1
        Status:                                    Online

      Node Name:                                   mars
        Votes Configured:                          1
        Votes Contributed:                         1
        Status:                                    Online

    Device Quorum
    Quorum Device Name:                            /dev/did/rdsk/d2s2
    Votes Configured:                              1
    Votes Contributed:                             1
    Nodes Having Access:
      venus                                        Enabled
      mars                                         Enabled
    Owner Node:                                    venus
    Status:                                        Online
```

Installing Sun Cluster 3.0 from a Jumpstart server

Sun Cluster 3.0 can be installed from a Jumpstart server. To set up the Jumpstart server to perform an installation of SC 3.0:

1. Set up the Jumpstart install server as a Solaris install server. Refer to the *Solaris Advanced Installation Guide* for instructions on setting up a Solaris Jumpstart install server.

   a. Copy the Solaris 8 CD image to the Jumpstart server.

```
Insert the Solaris 8 CD into the CDROM drive
# mkdir <InstallPath>
# cd /cdrom/cdrom0/s2/Solaris_2.8/Tools
# ./setup_install_server <InstallPath>

  <InstallPath> Empty directory where the Solaris 8
                CD image is to be created
```

   b. Create and populate a custom Jumpstart directory.

```
# mkdir <CustomDir>
# cd <CustomDir>
# cp -r <InstallPath>/Solaris_2.8/Misc/jumpstart_sample/* .
# share -F nfs -o ro <CustomDir>

      <CustomDir> is the path to a directory which will
                  hold custom jumpstart config files
```

2. Add the cluster nodes as install clients. Make sure each cluster node exists in the `/etc/hosts` file and/or any appropriate name services. Refer to the *Solaris Advanced Installation Guide* for instructions on adding install clients.

```
# cd <InstallPath>/Solaris_2.8/Tools
# ./add_install_client -c <JumpServer>:<CustomDir> <nodename>\
                          <arch>

    -c <JumpServer>:<CustomDir> - specifies the location of
                                  the custom Jumpstart
                                  Directory
    <nodename> - Hostname of cluster node to add
    <arch>     - architecture of the cluster node (e.g. sun4u)
```

Installing Sun Cluster 3.0 from a Jumpstart server

➤ Setup the Jumpstart server as a Solaris install server

```
# mkdir /export/jumpstart/solaris8
# cd /cdrom/cdrom0/s2/Solaris_2.8/Tools
# ./setup_install_server /export/jumpstart/solaris8
```

➤ Create and populate a custom Jumpstart directory. Make sure to share the custom Jumpstart directory

```
# mkdir /jumpstart
# cd /jumpstart
# cp -r /export/jumpstart/solaris8/Solaris_2.8/Misc/jumpstart_sample/* .
# share -F nfs -o ro /jumpstart   (you may want to add an entry in the dfstab file)
```

➤ Add the cluster nodes as install clients

```
# cd /export/jumpstart/solaris8/Solaris_2.8/Tools
# ./add_install_client -c jumpserv:/jumpstart venus sun4u
updating /etc/bootparams
copying inetboot to /tftpboot
# ./add_install_client -c jumpserv:/jumpstart mars sun4u
updating /etc/bootparams
```

Installing Sun Cluster 3.0 from a Jumpstart Server (Continued)

3. Copy the Sun Cluster 3.0 CD image to the Jumpstart server

```
Insert the Sun Cluster 3.0 CD into the CDROM drive

# cd /cdrom/suncluster_3_0/SunCluster_3.0/Tools
# ./scinstall -a <SC-InstallPath>

  <SC-InstallPath> - Empty directory where the Sun Cluster 3.0
                     CD image is to be created
```

4. Use `scinstall` to configure the Jumpstart server for each of the cluster nodes. `scinstall` must be run from the CD image created in step 3. `scinstall` will ask for cluster configuration information for each node of the cluster:

```
# cd <SC-InstallPath>/SunCluster_3.0/Tools
# ./scinstall
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

➤ Copy the Sun Cluster 3.0 CD to the Jumpstart server

```
# cd /cdrom/suncluster_3_0/SunCluster_3.0/Tools
# ./scinstall -a /export/jumpstart/suncluster_3_0
Copying “/cdrom/suncluster_3_0”
110907 blocks
Completed copy of “/cdrom/suncluster_3_0”
```

➤ Use `scinstall(1M)` to configure the Jumpstart server

```
# cd /export/jumpstart/suncluster_3_0/SunCluster_3.0/Tools
# ./scinstall
  *** Main Menu ***

    Please select from one of the following (*) options:

        1) Establish a new cluster using this machine as the first node
        2) Add this machine as a node in an established cluster
      * 3) Configure a cluster to be JumpStarted from this install server
        4) Add support for a new data service to this cluster node
        5) Print release information for this cluster node

      * ?) Help with menu options
      * e) Exit

    Option:  3
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

The following page continues the example scinstall session.

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

```
*** Custom JumpStart ***

  This option is used to configure each node in a cluster to be
  JumpStarted from this Solaris install server. Before this option can
  be used, this server must already be set up as a Solaris install
  server and configured to JumpStart each node as a Solaris install
  client. Refer to the Solaris documentation for more information on
  how to set up a Solaris install server, Solaris install clients, and
  a custom JumpStart directory.

  You will be asked to provide all of the information usually needed to
  directly add each node to a cluster. This information will be stored
  for later use under whatever custom JumpStart directory you specify.
  The rules file will be updated to point to both default Solaris
  install profile and a special custom JumpStart finish script.

  Press Ctrl-d at any time to return to the Main Menu.

  Do you want to continue (yes/no) [yes]?  yes

>>> Custom JumpStart Directory <<<

  In order to set up an install server to install and configure Sun
  Cluster nodes using custom JumpStart, each node must already be set
  up in the usual way for Solaris JumpStart installation. In
  particular, you must have already run add_install_client(1M) with a
  -c option specifying a JumpStart directory on this install server. In
  addition, this JumpStart directory must already exist and must
  contain the “check” utility. However, it is not necessary to create a
  “rules” file; scinstall will create or update this file with the
  necessary install rules for each cluster node.

  For more information regarding JumpStart and setting up a Solaris
  install client, please refer to the install_scripts(1M) man page and
  the Solaris installation documentation.

  What is your JumpStart directory name?   /jumpstart

>>> Cluster Name <<<

  Each cluster has a name assigned to it. The name can be made up of
  any characters other than whitespace. It may be up to 256 characters
  in length. And, you may want to assign a cluster name which will be
  the same as one of the failover logical host names in the cluster.
  Create each cluster name to be unique within the namespace of your
  enterprise.

  What is the name of the cluster you want to establish? planets

>>> Cluster Nodes <<<

  This release of Sun Cluster supports a total of up to 8 nodes.

  Please list the names of all cluster nodes planned for the initial
  cluster configuration. You must enter at least two nodes. List one
  node name per line. When finished, type Control-D:

  Node name:  venus
  Node name:  mars
  Node name (Ctrl-D to finish):  ^D

  This is the complete list of nodes:

      venus
      mars

  Is it correct (yes/no) [yes]?  yes
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

The following page continues the example `scinstall` session.

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

```
  >>> Authenticating Requests to Add Nodes <<<

    Once the first node establishes itself as a single node cluster,
    other nodes attempting to add themselves to the cluster configuration
    must be found on the list of nodes you just provided. The list can be
    modified once the cluster has been established using scconf(1M), or
    other tools.

    By default, nodes are not securely authenticated as they attempt to
    add themselves to the cluster configuration. This is generally
    considered adequate, since nodes which are not physically connected
    to the private cluster interconnect will never be able to actually
    join the cluster. However, DES authentication is available. If DES
    authentication is selected, you must configure all necessary
    encryption keys before any node can join (see keyserv(1M),
    publickey(4)).

    Is it okay to forgo DES authentication (yes/no) [yes]?  yes

  >>> Network Address for the Cluster Transport <<<

    The private cluster transport uses a default network address of
    172.16.0.0. But, if this network address is already in use elsewhere
    within your enterprise, you may need to select another address from
    the range of recommended private addresses (see RFC 1597 for
    details).

    If you do select another network address, please bear in mind that
    the Sun Clustering software requires that the rightmost two octets
    always be zero.

    The default netmask is 255.255.0.0; you may select another netmask,
    as long as it minimally masks all bits given in the network address
    and does not contain any “holes”.

    Is it okay to accept the default network address (yes/no) [yes]? yes

  >>> Point-to-Point Cables <<<

    The two nodes of a two-node cluster may use a directly-connected
    interconnect. That is, no cluster transport junctions are configured.
    However, when there are greater than two nodes, this interactive form
    of scinstall assumes that there will be exactly two cluster transport
    junctions.

    Does this two-node cluster use transport junctions (yes/no) [yes]?  yes

  >>> Cluster Transport Junctions <<<

    Note that interactive scinstall assumes that all transport junctions
    are of type “switch” and that they accept no special properties
    settings. If the junctions which you are using do not fall into this
    category, you may need to use non-interactive scinstall by specifying
    a complete set of command line options. For more information, refer
    to the scconf_transp_jct family of man pages (e.g.,
    scconf_transp_jct_etherhub(1M)).

    What is the name of the first junction in the cluster [switch1]?  switch1

    What is the name of the second junction in the cluster [switch2]?  switch2
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

The following page continues the example `scinstall` session.

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

```
 >>> Cluster Transport Adapters and Cables <<<

    You must configure at least two connection points to the private
    cluster transport for each node in the cluster. More than two
    connection points are allowed, but this interactive form of scinstall
    assumes exactly two.

    Note that interactive scinstall does not allow you to specify any
    special transport adapter properties settings. If your adapters have
    special properties which must be set, you may need to use
    non-interactive scinstall by specifying a complete set of command
    line options. For more information, please refer to the man pages for
    your adapters in the scconf_transp_adap family of man pages (e.g.,
    scconf_transp_adap_hme(1M)).

For node “venus”,
    What is the name of the first cluster transport adapter?  hme1

    All transport adapters support the “dlpi” transport type. Ethernet
    adapters are supported only with the “dlpi” transport; however, other
    adapter types may support other types of transport. For more
    information on which transports are supported with which adapters,
    please refer to the scconf_transp_adap family of man pages
    (scconf_transp_adap_hme(1M), ...).

For node “venus”,
    Is “hme1” an ethernet adapter [yes]?  yes

    The “dlpi” transport type will be set for this cluster.

For node “venus”,
    Name of the junction to which “hme1” is connected [switch1]?  switch1

    Each adapter is cabled to a particular port on a transport junction.
    And, each port is assigned a name. You may explicitly assign a name
    to each port. Or, for certain junction types, you may allow scinstall
    to assign a default name for you. The default port name assignment
    sets the name to the node number of the node hosting the transport
    adapter at the other end of the cable.

    Please remember that some types of cluster transport junctions are
    not compatible with the default port naming assignments. For more
    information regarding port naming requirements, refer to the
    scconf_transp_jct family of man pages (e.g.,
    scconf_transp_jct_dolphinswitch(1M)).

For node “venus”,
    Okay to use the default for the “hme1” connection [yes]?  yes
For node “venus”,
    What is the name of the second cluster transport adapter?  hme2
For node “venus”,
    Name of the junction to which “hme2” is connected [switch2]?  switch2
For node “venus”,
    Use the default port for the “hme2” connection [yes]?  yes
For node “mars”,
    What is the name of the first cluster transport adapter?  hme1
For node “mars”,
    Name of the junction to which “hme1” is connected [switch1]?  switch1
For node “mars”,
    Okay to use the default for the “hme1” connection [yes]?  yes
For node “mars”,
    What is the name of the second cluster transport adapter?  hme2
For node “mars”,
    Name of the junction to which “hme2” is connected [switch2]?  switch2
For node “mars”,
    Use the default port for the “hme2” connection [yes]?  yes
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

The following page continues the example scinstall session.

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

```
 >>> Global Devices File System <<<

   Each node in the cluster must have a local file system mounted on
   /global/.devices/node@<nodeID> before it can successfully participate
   as a cluster member. Since the “nodeID” is not assigned until
   scinstall is run, scinstall will set this up for you. However, in
   order to do this, you must supply the name of either an
   already-mounted file system or raw disk partition at this time. This
   file system or partition should be at least 100 MB in size.

   If an already-mounted file system is used, the file system must be
   empty. If a raw disk partition is used, a new file system will be
   created for you.

   The default is to use /globaldevices.

For node “venus”,
   Is it okay to use this default (yes/no) [yes]?

For node “mars”,
   Is it okay to use this default (yes/no) [yes]?

 >>> Confirmation <<<

   Your responses indicate the following options to scinstall:

For node “venus”,
     scinstall -c /jumpstart -h venus \
          -C planets \
          -N venus \
          -T node=venus,node=mars,authtype=sys \
          -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2 \
          -B type=switch,name=switch1 -B type=switch,name=switch2 \
          -m endpoint=:hme1,endpoint=switch1 \
          -m endpoint=:hme2,endpoint=switch2

   Are these the options you want to use [yes]?  yes

For node “mars”,
     scinstall -c /jumpstart -h mars \
          -C planets \
          -N venus \
          -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2 \
          -m endpoint=:hme1,endpoint=switch1 \
          -m endpoint=:hme2,endpoint=switch2

   Are these the options you want to use [yes]?  yes
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

The following page continues the example `scinstall` session.

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

```
Do you want to continue with JumpStart set up (yes/no) [yes]?

Created “autoscinstall.d/3.0”
Copied  “autoscinstall.class” to autoscinstall.d/3.0
Copied  “autoscinstall.finish” to autoscinstall.d/3.0
Copied  “autoscinstall.finish_ksh” to autoscinstall.d/3.0
Created “/jumpstart/autoscinstall.d/nodes”
Created “/jumpstart/autoscinstall.d/nodes/venus”
Created “/jumpstart/autoscinstall.d/nodes/../clusters”
Created “/jumpstart/autoscinstall.d/nodes/../clusters/planets”
Created “/jumpstart/autoscinstall.d/nodes/../clusters/planets/venus”
Created “/jumpstart/autoscinstall.d/nodes/venus/autoscinstall.data”
Updating “rules” file for host “venus” ...

Running the “check” utility...
--------------------------------
Validating rules...
Validating profile autoscinstall.d/3.0/autoscinstall.class...
Validating profile host_class...
Validating profile any_machine...
The custom JumpStart configuration is ok.
--------------------------------

Created “/jumpstart/autoscinstall.d/nodes/mars”
Created “/jumpstart/autoscinstall.d/nodes/../clusters/planets/mars”
Created “/jumpstart/autoscinstall.d/nodes/mars/autoscinstall.data”
Updating “rules” file for host “mars” ...

Running the “check” utility...
--------------------------------
Validating rules...
Validating profile autoscinstall.d/3.0/autoscinstall.class...
Validating profile host_class...
Validating profile any_machine...
The custom JumpStart configuration is ok.
--------------------------------

Hit ENTER to continue:
```

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

5. scinstall will generate a custom profile and finish script as well as add an entry for each node in the rules.ok file. The profile and finish script may be modified to suit individual needs. These files are located in <CustomDir>/autoscinstall.d/3.0.

Profile File	autoscinstall.class
Finish Script	autoscinstall.finish, autoscinstall.finish_ksh

6. Boot each cluster node from the Jumpstart server

```
ok boot net - install
```

7. Perform Sun Cluster 3.0 Post Installation steps as outlined earlier in this chapter:

   a. Install the volume manager

   b. Set up the proper environment

   c. Install Sun Cluster data service software

   d. Configure the volume manager

   e. Create and mount global filesystems

   f. Configure PNM

   g. Update ntp.conf files

   h. Enable and configure quorum

   i. Verify the installation

Installing Sun Cluster 3.0 from a Jumpstart server (Continued)

- You may modify the finish script and profile generated by `scinstall`.

```
# cd /jumpstart/autoscinstall.d/3.0
# ls
autoscinstall.class       autoscinstall.finish      autoscinstall.finish_ksh
```

- Boot each cluster node from the install server:

```
ok boot net - install
```

- Complete the Sun Cluster 3.0 installation:
    - Install the volume manager
    - Set up the proper environment
    - Install Sun Cluster data service software
    - Configure the volume manager
    - Create and mount global filesystems
    - Configure PNM
    - Update ntp.conf files
    - Enable and configure quorum
    - Verify the installation

CHAPTER 3

Configuring Solstice DiskSuite with Sun Cluster 3.0

Objectives

Purpose

In this chapter, we will provide some basic background information on installing and configuring DiskSuite for use with Sun Cluster 3.0.

Prerequisites

Understanding of basic disk management principles

Objectives

Upon completion of this chapter, the participant will be able to:

- ➤ Describe the features and basic concepts behind Solstice DiskSuite
- ➤ Describe how to install and configure Solstice DiskSuite for Sun Cluster 3.0

Objectives

- Learn basic DiskSuite concepts and architecture
- Learn how to install and configure DiskSuite for use in Sun Cluster 3.0

An Introduction To Solstice DiskSuite

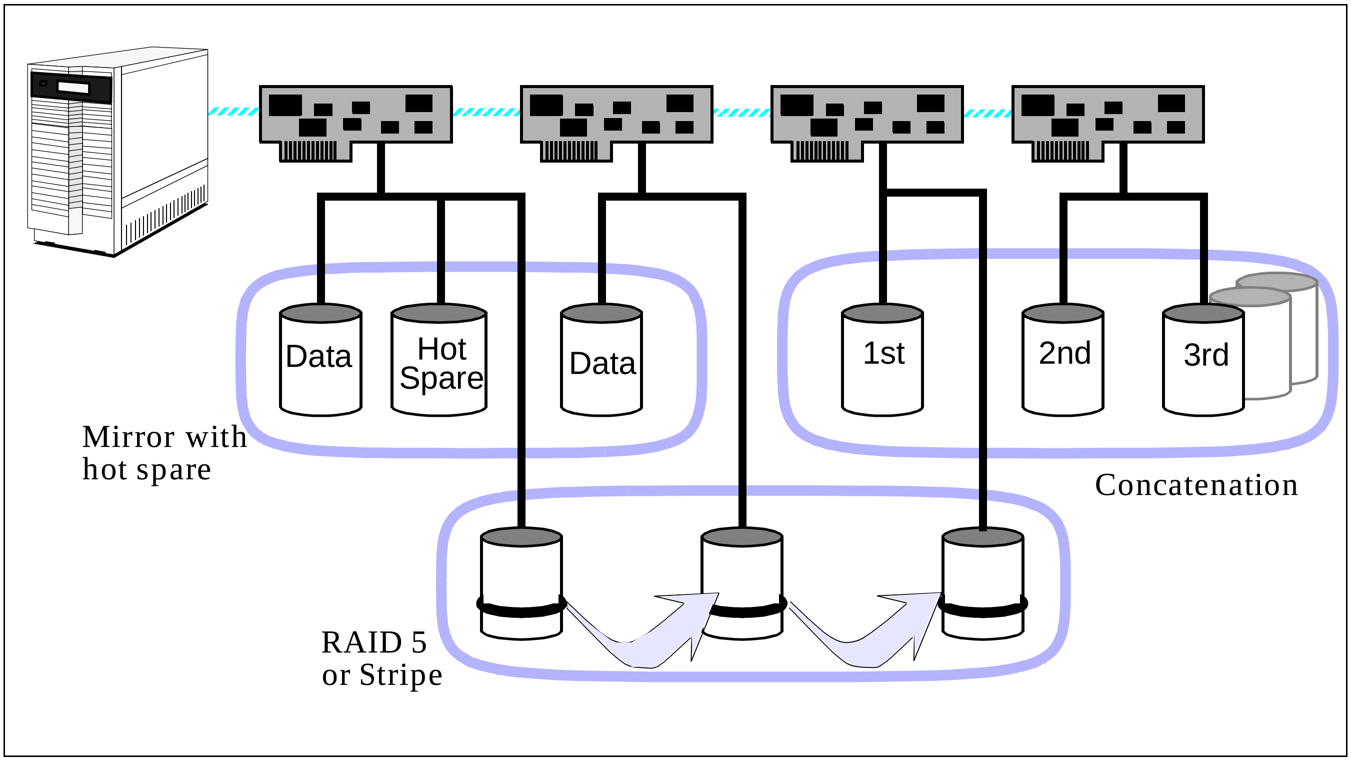
Overview & Features

Solstice DiskSuite (SDS) provides volume management for critical servers. It employs disk striping and mirroring technologies, a journaling file system and mechanisms to support very large file systems and data sets. Any disk supported by Solaris can be managed by Solstice DiskSuite.

Solstice DiskSuite supports the following features:

- **Disk Mirroring (RAID 1)** - multiple copies of data are maintained on different physical devices; 2-way and 3-way mirrors are supported
- **Disk Striping (RAID 0)** - data is interlaced among multiple physical devices
- **Disk Concatenation (RAID 0)** - two or more physical devices are combined into a single logical device
- **Disk Mirroring and Striping (RAID 1+0, 0+1)** - Disk mirroring and striping can be combined to provide high performance and high availability
- **RAID 5** - data and parity is interlaced among multiple physical devices
- **Logged UFS** - UFS updates are recorded in a log (called a *logging device*) before they are applied to the actual file system device (called a *master device*). Recovery of the file system involves reading and reconciling the log device instead of checking the entire file system device.
- **Expandable UFS file systems** - SDS allows increasing the size of a Unix file system without having to recreate the entire file system
- **Disksets** - Drives can be logically separated into disksets, allowing the cluster to manipulate the logical set of drives as a single entity
- **Hot Spares** - Drives can be assigned to be automatically substituted for a failed component of a mirrored or RAID 5 device

An Introduction to Solstice DiskSuite



- Solstice DiskSuite (SDS) supports:
  - Disk Mirroring (RAID 1)
  - Disk Striping (RAID 0)
  - Disk Concatenation (RAID 0)
  - Disk Mirroring and Striping (RAID 1+0, RAID 0+1)
  - RAID-5 (Not Supported in SC 3.0)
  - Logged UFS
  - Expandable UFS
  - Disksets
  - Hot Spares

An Introduction to Solstice DiskSuite

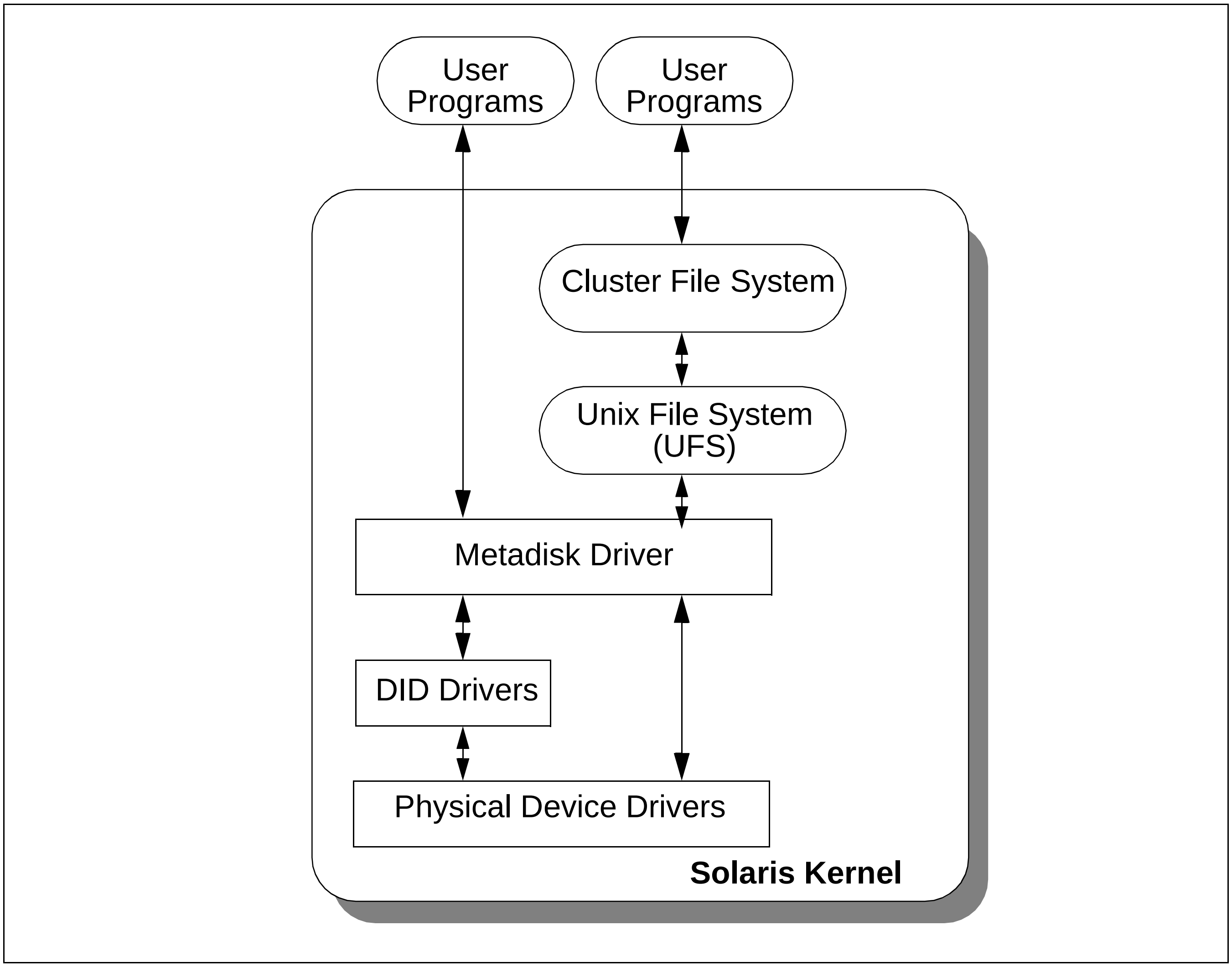
Architecture

User programs interact with logical volumes, called ***metadevices***, through the metadisk driver. Metadevices are used like physical disk partitions.

The metadisk driver is a loadable driver with standard interfaces. This metadisk driver “knows” the location of the physical disks through the use of a ***metadevice state database***.

In Sun Cluster 3.0, the metadisk driver resides between the file system or data service applications and the disk ID pseudo driver or actual physical device drivers.

An Introduction to DiskSuite



- User programs or Unix file systems interact with logical volumes called *metadevices*. Metadevice access is handled by the metadisk driver
- The metadisk driver resides between the file system drivers and the DID and / or physical device layer in the kernel

Metadevices

Metadevices are logical devices that are made up of one or more physical disk partitions. After they are created, metadevices are used like disk partitions. The use of metadevices is transparent to application software.

Metadevice Types

Metadevices are the usable constructs in SDS. There are several metadevice types that can be created:

- Simple metadevices - concatenations or stripes created from actual disk partitions (DID or physical disk slices)
- Metamirror - mirrors of simple metadevices
- RAID 5 metadevice
- Metatrans device - Logged UFS device, usually made up of metamirrors

Metadevice Names

Metadevices are located in `/dev/md/dsk` (block devices) and `/dev/md/rdsk` (raw devices). If disksets are used (disksets are usually used in a cluster), the metadevices for a diskset are located in `/dev/md/`*`<DisksetName>`*`/dsk` or `/dev/md/`*`<Diskset Name>`*`/rdsk`. Metadevice names begin with “d” followed by a number. For example:

`/dev/md/ds-1/dsk/d10`	Metadevice d10 in the ds-1 diskset (block device)
`/dev/md/rdsk/d28`	Metadevice d28 made from disks not allocated to any diskset (raw device)

Metadevices

- Metadevices are logical volumes made up of one or more disk partitions
- Metadevice types:
    - Simple Metadevices - concatenation or stripes of disk partitions
    - Metamirrors - mirrors made up of simple metadevices
    - RAID-5 Metadevices - RAID 5 devices
    - Metatrans device - Logged UFS device, usually built on metamirrors
- Metadevices names:
    - Located in `/dev/md/[r]dsk` or if part of a diskset in `/dev/md/<Diskset Name>/[r]dsk`
    - Metadevice names begin with a “d” followed by a number
    - Examples:
      `/dev/md/ds-1/dsk/d10`
      `/dev/md/rdsk/d28`

Metadevice Hierarchy

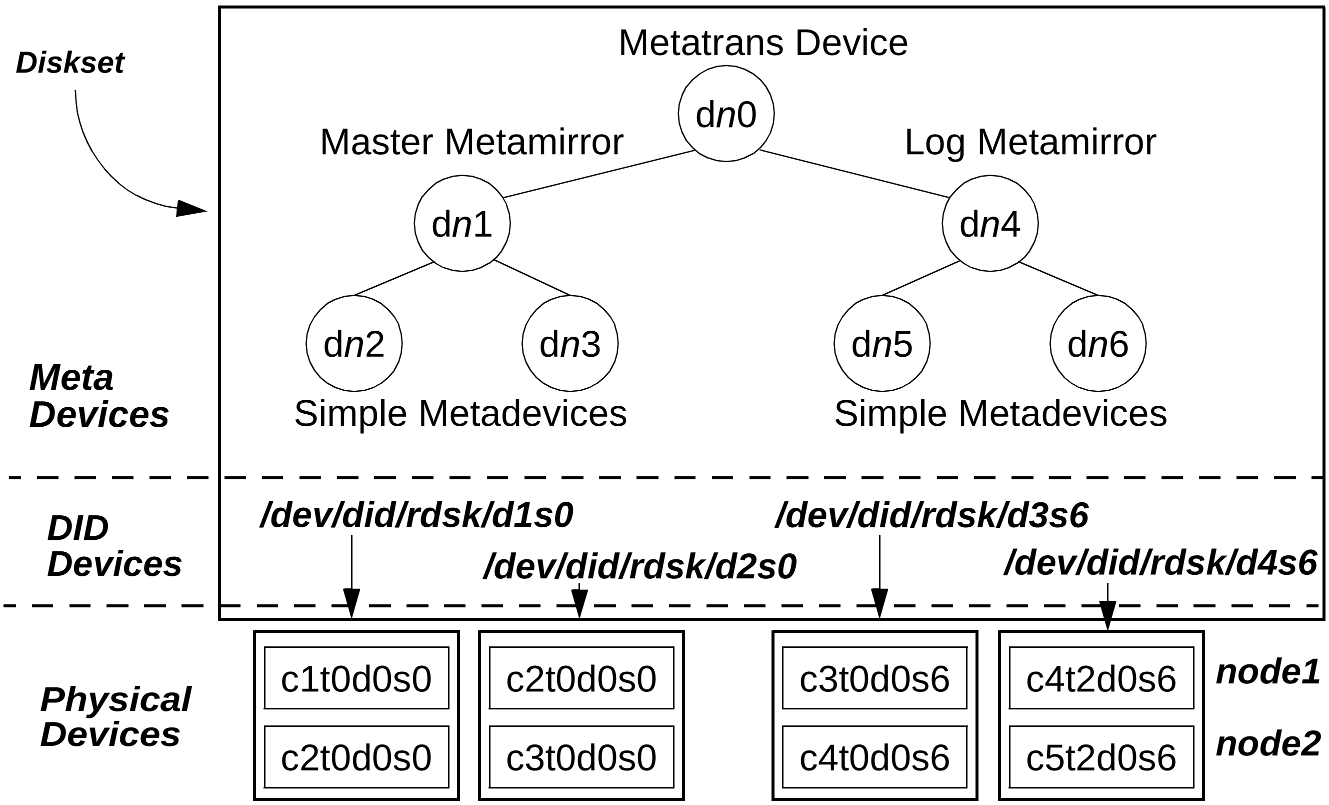
SDS is a “bottom-up” tool, that is, construction of metadevices is done starting at the bottom, with simple metadevices, and progressing up to “top-level” metadevices.

First of all, when using SDS in a cluster, physical disk drives (or their DID device equivalent) are grouped together into *disksets.* A disk drive can only be a member of a single diskset.

Once the drives are assigned to a diskset, disk slices or partitions from the drives are used to construct simple *metadevices*, the simple metadevices can then be put together into *metamirrors* (mirrored devices), and metamirrors, in turn, can be put together into *metatrans* devices (logged devices for hosting a Unix file system). A file system would be built and mounted using the “top-level” metatrans device.

A suggested metadevice name convention is to use names ending in “0” (e.g. d10, d20, d110, etc.) for your top-level devices.

Metadevice Namespace



- SDS is a “bottom-up” tool, you start by assigning physical disks (or DID devices) to a diskset, within a diskset, you create simple metadevices (based on disk partitions) at the bottom level and use these metadevices to construct other, higher level metadevices
- “Top-level” metadevices are used by the file system or application programs
- Recommendation: name top-level metadevices with numbers ending in “0”

Metadevice State Database

The metadevice state database holds configuration and status information about all the metadevices configured in the system or diskset. Information such as which disk slices are part of each simple metadevice, which simple metadevices make up particular metamirrors and which metamirrors are used in a metatrans master and log devices are held in the metadevice state database.

In addition, SDS will store current status information about each of the metadevices in the metadevice state database. A failure of a submirror is an example of a metadevice state change that would be stored in the state database.

Obviously, the state database is a very important component of SDS, loss of the state database means that SDS would no longer have any idea as to how the disks are being used! To protect the contents of the state database, copies of the state database, called replicas, are maintained by SDS. SDS recommends that there be at least 3 copies of the state database created. Furthermore, at least half of the replicas must be accessible in order for SDS to continue running. SDS will render the metadevices unavailable unless **more** than half are accessible at boot time.

Each diskset in SDS has its own distinct set of state database replicas. SDS will automatically create 2 copies on each disk added to a diskset. A dedicated slice of at least 2MB is required to house the replicas. A set of replicas (at least 3 copies) must be created on the local disk drives of each nodes in order properly initialize SDS.

By default, each replica occupies 517 kilobytes (1034 disk blocks), however, larger replicas can be created for installations with a large number of metadevices.

Metadece State Database

- Contains information about the configuration and status of the metadevices
- Copies of the metadevice state database, called replicas, are maintained by SDS
- There should be a minimum of 3 replicas
- SDS can survive the loss of up to half of the total number of replicas, however SDS will not boot (i.e., metadevices will be unavailable) unless there are **more** than half the replicas available
- Disksets maintain their own set of replicas, automatically allocating space for 2 replicas on each disk in the diskset
- Replicas are placed in a separate dedicated slice of a disk
- By default replicas are 517k (1034 blocks), however, larger replicas can be created for installation with a large number of metadevices

Dual String Mediators

What are Dual-String Mediators?

Dual string mediators are required on DiskSuite based clusters where disksets are configured across exactly two disk arrays (two *strings* of disks - hence the name *Dual String* mediators) which are shared by two cluster nodes.

Strictly speaking, a mediator is one or all the nodes in the cluster. Each mediator contains the location of the other mediators in the cluster as well as the current commit count of each of the DiskSuite state databases.

Why are mediators needed?

In order to determine the state of the disk drives (i.e. who is currently mastering a diskset, which mirrors are current, how stripes and concatenations are laid out, etc.) DiskSuite maintains a metadevice state database. This database is replicated across various disk devices and controllers. If there is any inconsistency in the database replicas, DiskSuite will go with what the majority (half + 1) of the database replicas contain. This is called a replica quorum. If a majority of the replica databases cannot be contacted, SDS effectively shuts down.

When a diskset is configured in a dual-string (i.e. 2 disk arrays) configuration, SDS will split the number of replicas for each diskset evenly across both arrays. If one of the disk arrays fails, there will be exactly half of the disk replicas available and a majority cannot be reached. This is where the mediators come in to play. If exactly half the replicas are available, the information in the mediators (namely the state database commit count) makes it possible for SDS to determine the integrity of the remaining replicas, effectively casting the tie-breaking vote, allowing DiskSuite to continue as if it had a replica quorum.

Dual String Mediators

- Dual string mediators are required when disksets are configured across exactly 2 disk strings (disk enclosures) connected to 2 cluster nodes
- Mediators are housed on the nodes of the cluster which are connected to the dual disk strings
- Mediators contain information on who the other mediators are and the current commit count held in the state database replicas of each diskset
- Mediators come into play when exactly half of the total state database replicas are accessible (i.e. when one of the disk arrays fails). In this case, SDS is not able to achieve replica quorum (half + 1 of the replicas accessible), which would normally cause SDS to shut itself down.
- In this case, mediators are able to cast the “deciding” vote (assuming that the commit counts in the replicas and the commit counts in the mediators agree), allowing SDS to continue without a true replica quorum.

Dual String Mediators (Continued)

What are *Golden* mediators

In the case that exactly half of the state database replicas are accessible, before the mediators can cast the deciding vote, a *mediator* quorum must exist (half + 1 of the total number of mediators - *this is different from a replica quorum*). If the mediator quorum does not exist, the mediators are not able to cast the deciding vote and SDS will effectively shut down.

However, the mediator quorum requirement is suspended if the accessible mediator is in a *Golden* state. A mediator is set to a golden state via the following algorithm:

1. If exactly half of the state database replicas are accessible, and a state database change is required, the commit counts on all mediators are updated and the status on all mediators is marked as Not Golden.

2. All accessible database replicas are updated with the new information, including the current commit count

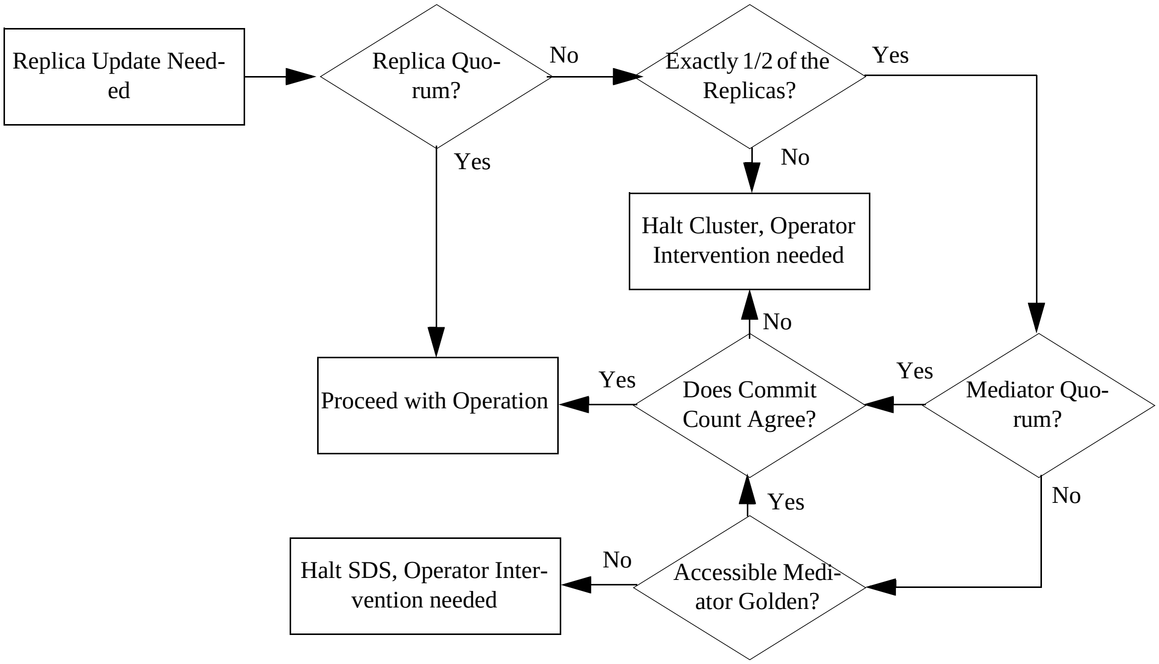
3. If, during steps 1 and 2 all mediators were successfully updated AND the number of replicas successfully updated is exactly half of the total number of replicas, all mediators are set to Golden status, which indicates all mediators contain valid information.

The golden status is held in volatile memory, such that a reboot of a mediator host will cause the Golden status to be revoked on that node.

Dual String Mediators

- Mediators can only cast a deciding vote if a majority of the mediators are available (mediator quorum)
- Golden mediators allow the mediators to work when a mediator quorum is not available
- A Two-Phase commit algorithm is used to set the golden status
- Helps protect against certain double failures - a disk string failing followed by a host failing.

Summary of Dual-String Mediator Operation


Replica Update Needed
Replica Quorum?
No
Yes
Exactly 1/2 of the Replicas?
Yes
No
Halt Cluster, Operator Intervention needed
No
Proceed with Operation
Yes
Does Commit Count Agree?
Yes
Mediator Quorum?
No
Yes
Halt SDS, Operator Intervention needed
No
Accessible Mediator Golden?

DiskSuite Command Summary

The following commands are used by DiskSuite to manipulate metadevices. We will be covering specific uses of several of these commands later in this chapter.

Command	Description
`growfs`	Non-destructively expands a UFS file system
`metaclear`	Clears active metadevices
`metadb`	Creates and deletes replicas of the metadevice state database
`metahs`	Manages metadevice hot spares and hot spare pools
`metainit`	Configures metadevices
`metaoffline`	Offlines submirrors
`metaonline`	Onlines submirrors
`metaparam`	Modifies parameters of metadevices and metamirrors
`metareplace`	Replaces components of submirrors
`metaroot`	Sets up system files for root metadevice
`metaset`	Configures disksets
`metastat`	Prints status information for metadevice
`metasync`	Handles mirror resync during reboot
`metadetach`	Detaches a metadevice from a metamirror or metatrans
`metattach`	Attaches a metadevice to a metamirror or metatrans
`medstat`	Check status of dual-string mediators

All SDS commands are located in `/usr/sbin`.

DiskSuite Command Summary

- The following commands can be used to manipulate DiskSuite metadevices:
  - `growfs`
  - `metaclear`
  - `metadb`
  - `metahs`
  - `metainit`
  - `metaoffline`
  - `metaonline`
  - `metaparam`
  - `metareplace`
  - `metaroot`
  - `metaset`
  - `metastat`
  - `metasync`
  - `metadetach`
  - `metattach`
  - `medstat`
- All commands are located in `/usr/sbin`

Installing and Configuring SDS in Sun Cluster 3.0

Installing and configuring Solstice DiskSuite for use in a Sun Cluster 3.0 environment consists of the following steps:

1. Install the appropriate packages for Solstice DiskSuite
2. Install the SDS T-patch 108693-02
3. Modify the `md.conf` file appropriately
4. Reboot all nodes in the cluster.
5. Create `/.rhosts` files or add root to group 14
6. Initialize the local state databases
7. Create disksets to house HA data service data
8. Add drives to each diskset
9. Partition disks in the disksets
10. Create the metadevices for each diskset
11. Configure dual string mediators

Installing and Configuring SDS in Sun Cluster 3.0

1. Install the appropriate packages for Solstice DiskSuite
2. Install the SDS T-patch 108693-02
3. Modify the `md.conf` file appropriately
4. Reboot all nodes in the cluster
5. Create `/.rhosts` files or add root to group 14
6. Initialize the local state databases
7. Create disksets to house data service data
8. Add drives to each diskset
9. Partition disks in the disksets
10. Create the metadevices for each diskset
11. Configure dual string mediators

Step 1 - Install the Solstice DiskSuite Packages

Use the `pkgadd` utility to install the Solstice DiskSuite packages on all nodes. The Solstice DiskSuite packages are located on the Solaris 8 CD under

`<path of Solaris8 Software 2 of 2 CD>/Solaris_8/EA/products/DiskSuite_4.2.1/sparc/packages/`.

DiskSuite is made up of the following packages:

- ➤ `SUNWmdr` - Solstice DiskSuite Driver (required)
- ➤ `SUNWmdu` - Solstice DiskSuite Commands (required)
- ➤ `SUNWmdx` - Solstice DiskSuite 64-bit drivers (required for 64-bit kernel)
- ➤ `SUNWmdg` - DiskSuite GUI tool (optional)
- ➤ `SUNWmdnr` - SNMP trap generator for SDS configuration files (optional)
- ➤ `SUNWmdnu` - SNMP trap generator for SDS (optional)
- ➤ `SUNWmdja` - DiskSuite Japanese localization (not needed)

The following package is located on the SC3.0 CD and is installed by `scinstall`:

- ➤ `SUNWmdm` - DiskSuite Dual String Mediators

Step 1 - Install the Solstice DiskSuite Packages

```
# cd <path of Solaris8 CD>/Solaris_8/EA/products/DiskSuite_4.2.1/sparc
# pkgadd -d .
...
<Answer questions asked by pkgadd ...>
...
#
```

- Use `pkgadd` to install the SDS packages from the Solaris 8 CD
- Required packages are: `SUNWmdr`, `SUNWmdu`. If you are booting a 64-bit kernel, `SUNWmdx` is also required.

Step 2 - Install any Solstice DiskSuite Patches

Use the `patchadd` utility to add any required SDS patches.

Step 2 - Install any Solstice DiskSuite Patches

- ➤ Use `patchadd` to install any DiskSuite patches recommended by Sun

Step 3 - Modify the `md.conf` file

Based on your planned implementation, you may need to update DiskSuite's kernel configuration file: `/kernel/drv/md.conf`. There are 2 variables which may need to be updated:

Variable	Default Value	Description
nmd	128	Maximum number of metadevices. DiskSuite uses this setting to limit the ***names*** of the metadevices as well. If you are going to have 100 metadevices, but you want to name them d1000 through d1100, you need to set this to 1101, not 100. The maximum value for nmd is 8192.
md_nsets	4	Maximum number of disksets. This number should be set to the number of disksets you plan to create in your cluster (probably equal to the number of logical hosts you plan to have -- assuming one diskset per logical host). The maximum value for md_nsets is 32.

This file must be kept identical on all nodes of the cluster. Changes to this file take effect after a reconfiguration reboot is performed.

Note: Before continuing to the next step, make sure to perform a reconfiguration reboot of the node, to ensure that the metadisk device driver is initialized properly.

Step 3 - Modify the `md.conf` file

```
#
#ident “@(#)md.conf   1.7     94/04/04 SMI”
#
# Copyright (c) 1992, 1993, 1994 by Sun Microsystems, Inc.
#
name=”md” parent=”pseudo” nmd=128 md_nsets=4;
```

- Location of the kernel configuration file is:

  `/kernel/drv/md.conf`

- If your DiskSuite namespace will have metadevice names greater than 128, you will need to increase the nmd parameter

- If you are going to create more than 4 disksets in the cluster, you will need to increase the md_nsets parameter

- Keep this file identical on all nodes of the cluster

- Changes are put into effect after a reconfiguration reboot

Step 4 - Reboot all nodes in the cluster

If `md.conf` was modified, shut down the cluster then do a reconfiguration reboot. Execute the `scshutdown` command on one node; then type `boot -r` at the `ok` prompt on all nodes.

If `md.conf` was not modified, there is no need to do a reconfiguration reboot. Execute the `scshutdown` command with the reboot (`-r`) option on one node.

Step 4 - Reboot all Nodes in the Cluster

- ➤ If `md.conf` was modified:

  `# scshutdown -g 0` *(run on one node)*

  `ok boot -r` *(on all nodes)*

- ➤ If `md.conf` was *not* modified:

  `# scshutdown -g 0` *(run on one node)*

  `ok boot` *(on all nodes)*

- ➤ Run `/usr/cluster/bin/scgdevs` on one node of the cluster after the reboot

Step 5 - Create `/.rhosts` files or add root to group 14

In order to allow SDS to coordinate its activities with the other nodes of the cluster, each cluster node must be granted root-level access on the other nodes. This can be accomplished either creating `/.rhosts` files on each node or by adding the root user to the `sysadmin` group (group 14).

Step 5 - Create `/.rhosts` files or add root to group 14

- SDS requires root level access on each node of the cluster
- **Option 1 -** Create `/.rhosts` files listing the names of all the cluster nodes on each node of the cluster:

```
# hostname
venus
# cat /.rhosts
mars
venus


# hostname
mars
# cat /.rhosts
mars
venus
```

- **Option 2 -** Add root to the sysadmin group on all nodes of the cluster:

```
# cat /etc/group
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:root ◄──────────
nobody::60001:
noaccess::60002:
nogroup::65534:
```

Step 6 - Initialize the local metadevice state databases

Solstice DiskSuite must be initialized on each node by creating a set of local metadevice state database replicas. You must have at least 3 metadevices state database replicas and they are created on a free slice of one of the local disk drives. To create the local state database replicas, perform the following steps:

1. Ensure that you have a free slice at least 2MB on a local disk on each node.

2. Create the local metadevice state database replicas using the metadb command:

```
metadb -a -f -c 3 cWtXdYsZ

-a attach new metadevice state database
-f with -a indicates initial creation of state databases
-c number of copies to create
CWtXdYsZ indicates disk and slice to be used for the
replicas
```

Note: The example shown above and on the following page illustrates creating the state database replicas on a single disk. It is strongly recommended that the metadevice state database replicas be spread across at least 3 separate disks; such that a single disk failure does not cause a loss of a majority of the metadevice state database replicas.

3. Verify that your local replicas were created with the `metadb` command.

Step 6 - Initialize the local metadevice state databases

1. Make sure you have a free slice of at least 2MB on a local disk of each cluster node

```
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -  377      280.55MB    (378/0/0)   574560
  1       swap    wu     378 -  550      128.40MB    (173/0/0)   262960
  2     backup    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  3        var    wm     551 -  820      200.39MB    (270/0/0)   410400
  4 unassigned    wm     821 -  955      100.20MB    (135/0/0)   205200
  5 unassigned    wm     956 - 2044      808.24MB    (1089/0/0) 1655280
  6        usr    wm    2045 - 2718      500.23MB    (674/0/0)  1024480
  7 unassigned    wm    2719 - 2732       10.39MB    (14/0/0)     21280
```

2. Create the metadevice state database replicas on the slice identified in step 1

```
# metadb -a -f -c 3 c0t0d0s7
```

3. Verify that the state databases were created:

```
# metadb
        flags          first blk       block count
     a        u         16              1034            /dev/dsk/c0t0d0s7
     a        u         1050            1034            /dev/dsk/c0t0d0s7
     a        u         2084            1034            /dev/dsk/c0t0d0s7
```

Step 7 - Create disksets for the data services

The following steps will create disksets to hold the cluster's data service data:

1. On one of the cluster nodes, use the `metaset` command to create the disksets.

```
metaset -s <DiskSet Name> -a -h <Host1 Host2 ...>

-s DiskSet Name
-a Add new hosts eligible to master the diskset
-h List of hosts eligible to master the diskset
```

2. Check the status of the newly created disksets by running the metaset command.

Step 7 - Create disksets for the data services

- Make sure all nodes are up and running in the cluster
- Make sure that the local metadevice state database replicas have been created on all nodes
- Use the `metaset` command to create the disksets. Perform this step on only one node:

```
# metaset -s web_data_1 -a -h mars venus
```

- Verify the creation of the diskset using the `metaset` command:

```
# metaset

Set name = web_data_1, Set number = 1

Host                Owner
  mars
  venus
```

Step 8 - Add drives to each diskset

After creating the disksets for the cluster, the next step is to use the `metaset` command to add disk drives into each diskset:

```
metaset -s <DiskSet Name> -a <List of disks to add>

-s DiskSet Name
-a Add disks to diskset
```

You should evenly distribute the disks in each diskset across at least 2 arrays to accommodate mirroring of data. Make sure to use the DID device name instead of the actual disk drive names.

Adding a disk to a diskset will cause the disk to be repartitioned as follows:

- A small portion of the drive (starting at cylinder 0) is placed into slice 7 to be used for metadevice state database replicas (usually at least 2MB)
- The rest of the drive is placed into slice 0

The drive will **not** be repartitioned if slice 7 has the following characteristics:

- Starts at cylinder 0
- At least 2MB in size (large enough to hold a state database)
- Has the V_UNMT flag set (unmountable flag)
- Not read only

After adding the drive to a diskset, the drive may be repartitoned as necessary, ***with the exception that slice 7 must be left alone***.

Verify that the contents of each diskset using the `metaset` command:

```
metaset -s <DiskSet Name>

-s DiskSet Name
```

Step 8 - Add drives to each diskset

- Use the `metaset` command to add drives to each diskset:

```
# metaset -s web_data_1 -a /dev/did/rdsk/d2 /dev/did/rdsk/d5
```

- Each disk added will be repartitioned (all data will be lost) with a small (~2MB) slice 7 starting at cylinder 0 (used to hold a metadevice database replica) and the rest of the drive in slice 0
- Disks may be repartitioned later, as long as slice 7 is left alone
- If slice 7 already exists with the proper characteristics, the drive will not be automatically repartitioned:
  - Starts at cylinder 0
  - At least 2MB
  - Unmountable flag set
  - Not read only
- Contents of the diskset can be checked with the `metaset` command:

```
# metaset -s web_data_1

Set name = web_data_1, Set number = 1

Host                Owner
  mars               Yes
  venus

Drive               Dbase
  /dev/did/dsk/d2    Yes
  /dev/did/dsk/d5    Yes
```

Step 9 - Repartition disks in diskset

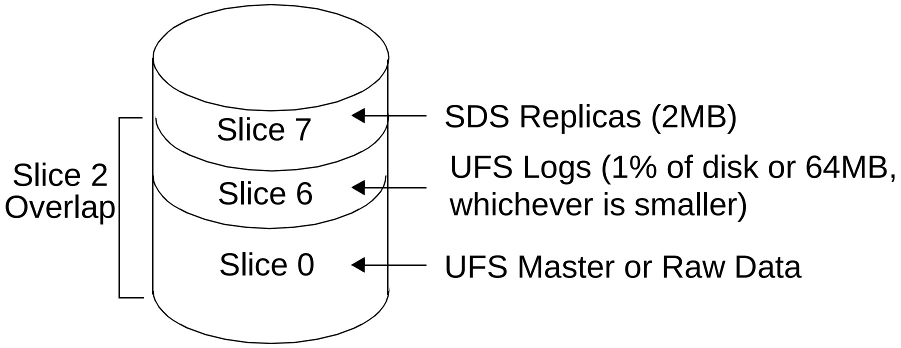
Unless you have used pre-partitioned disks with a proper slice 7 (as outlined in step 5), you will probably need to repartition each of the disks in the disksets you have created.

You may repartition the disks in any manner, the only restriction is that slice 7 be left alone. A **recommended** partitioning scheme is:

Slice	Recommended Size	Usage
0	Rest of Disk	Master Device for Data Service File System (or can be used raw)
2	Slices 0 + 6	Overlap (can be used for raw devices)
6	1% of Disk	Log Device for Data Service File System
7	2MB	SDS Replica

Step 9 - Repartition disks in diskset

Recommended Diskset Disk Partitioning Scheme



- Disks may be repartitioned in any manner, as long as slice 7 is left alone
- The UFS Log partitions (slice 6) are not required if only raw devices are to be used by the data service

Step 10 - Create the metadevices for each diskset

The final step in setting up Solstice DiskSuite is to create the metadevices on which we can build logged Unix file systems or use as raw devices. Metadevices can be created manually, using the proper form of the `metainit` command or by setting up and running an md.tab file.

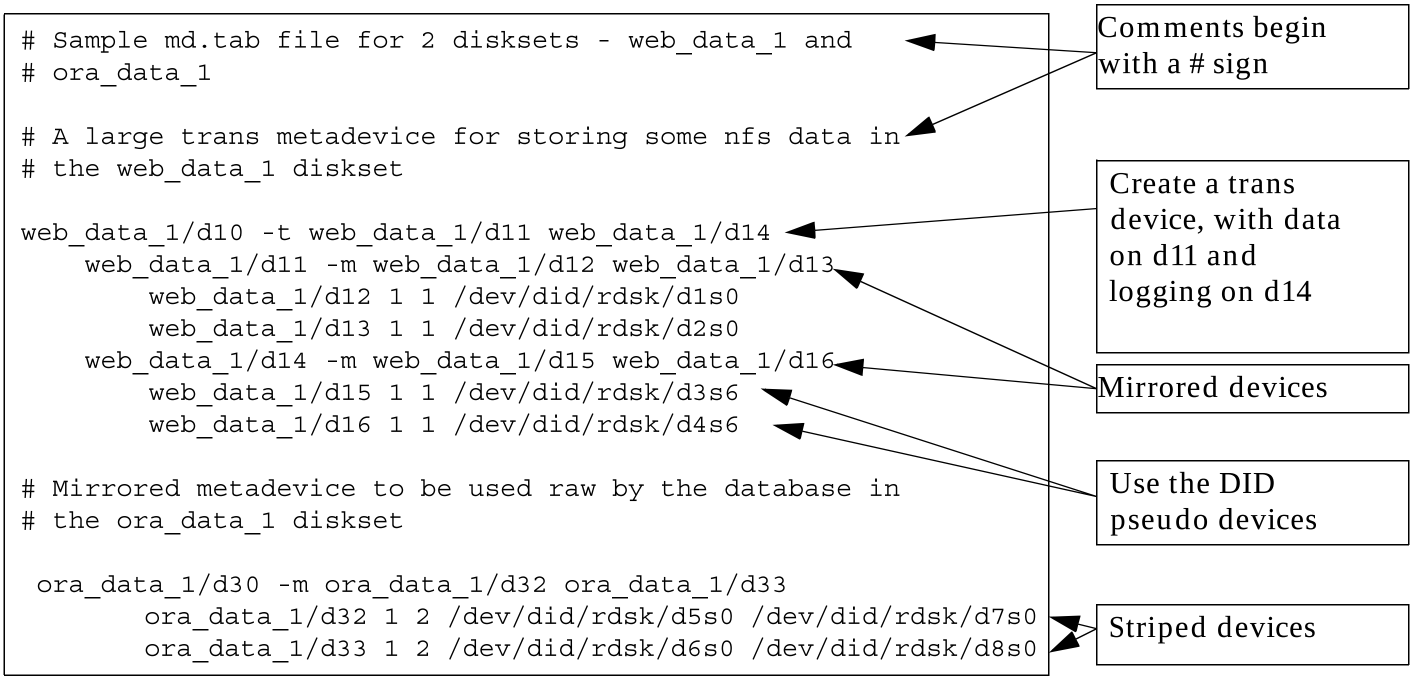
An `md.tab` file contains information that can be used to create all the metadevices required in a batch mode. Creating metadevices in Solstice DiskSuite is done using a "bottom up" method. Simple metadevices are first created to provide striping, concatenation, or concatenation of stripes. These metadevices are then combined into mirrored metadevices and these mirrored metadevices, in turn, are then either used raw or as a part of a trans metadevice (logged UFS device). The `md.tab` file outlines the parameters used by `metainit` to create each of these different device types. The `md.tab` file may also contain hot spare pool definitions.

Once the `md.tab` file is created, it is "run" by using the `metainit` command:

```
metainit -s <DiskSet Name> -a

-s DiskSet name to create metadevices for
-a Use /etc/lvm/md.tab for metadevice definitions
```

Step 10 - Create metadevices for each disksets

- Create an `/etc/lvm/md.tab` to create the metadevices required for each diskset:



```
# Sample md.tab file for 2 disksets - web_data_1 and
# ora_data_1

# A large trans metadevice for storing some nfs data in
# the web_data_1 diskset

web_data_1/d10 -t web_data_1/d11 web_data_1/d14
    web_data_1/d11 -m web_data_1/d12 web_data_1/d13
        web_data_1/d12 1 1 /dev/did/rdsk/d1s0
        web_data_1/d13 1 1 /dev/did/rdsk/d2s0
    web_data_1/d14 -m web_data_1/d15 web_data_1/d16
        web_data_1/d15 1 1 /dev/did/rdsk/d3s6
        web_data_1/d16 1 1 /dev/did/rdsk/d4s6

# Mirrored metadevice to be used raw by the database in
# the ora_data_1 diskset

 ora_data_1/d30 -m ora_data_1/d32 ora_data_1/d33
        ora_data_1/d32 1 2 /dev/did/rdsk/d5s0 /dev/did/rdsk/d7s0
        ora_data_1/d33 1 2 /dev/did/rdsk/d6s0 /dev/did/rdsk/d8s0
```

Comments begin with a # sign

Create a trans device, with data on d11 and logging on d14

Mirrored devices

Use the DID pseudo devices

Striped devices

Step 10 - Create metadevices for each diskset (continued)

As an alternative to using the `md.tab` file, metadevices can be created manually. To manually create metadevices at the command line, use the appropriate form of the `metainit(1M)` command:

Simple Metadevices (Concats/Stripes)

```
metainit -s <DiskSet Name> <MD name> <Stripes> <Width> <Components>

<MD name> - Metadevice name (i.e. d21)
<Stripes> - Number of stripes - Concatenations: should be number of
                                slices to be concatenated
                                Stripes: should be set to 1
<Width> - Width of the stripe - Concatenations: should be set to 1
                                Stripes: should be set to number
                                of slices to stripe across
<Components> - Disk Slices to use
```

Metamirrors

```
metainit -s <DiskSet Name> <MD name> -m <SubMirrors>

<MD name> - Metadevice name (i.e. d21>
<SubMirrors> - list of metadevices to mirror
```

Metatrans devices

```
metainit -s <DiskSet Name> <MD name> -t <Master> <Log>

<MD name> - Metadevice name (i.e. d21>
<Master> - Device to use as master device
<Log> - Device to use for logging
```

Note: Refer to the *Solstice DiskSuite 4.2 User's Guide* or the man pages for the `metainit(1M)` command for more information on creating metadevices

Step 10 - Create metadevices for each diskset (continued)

- Instead of using the `md.tab` file, metadevices can be created manually, using the appropriate form of the `metainit` command
- Simple metadevices:
  - `metainit -s web_data_1 d12 1 1 /dev/did/rdsk/d1s0`
  - `metainit -s ora_data_1 d32 1 2 /dev/did/rdsk/d5s0 /dev/did/rdsk/d7s0`
- Metamirrors:
  - `metainit -s web_data_1 d11 -m d12 d13`
- Metatrans device:
  - `metainit -s web_data_1 d10 -t d11 d14`

Step 11 - Configure dual string mediators

If any disksets were created using exactly two disk arrays (which are connected to two cluster nodes), dual string mediators must be configured. The following rules apply when configuring dual string mediators:

- Disksets using dual strings and two hosts must be configured with exactly two mediator hosts, and these hosts must the same two hosts used for the diskset
- A diskset cannot have more than two mediator hosts
- Mediators cannot be configured for disksets which do not meet the two-string, two-host criteria

Note that mediators are not only for use in two-node clusters, cluster having more than two nodes may also require the use of mediators, depending on the topology and how the disksets are constructed.

You may add new mediators using the `metaset` command:

```
metaset -s <DiskSet Name> -a -m <Mediator Host List>

-s DiskSet Name to add mediators for
-a Add new mediators
-m Mediator host names to add (comma separated list)
```

Once the mediators have been added, you can check the status of the mediators using the `medstat` command:

```
medstat -s <DiskSet Name>

-s DiskSet Name to check mediator status for
```

Mediator information can be deleted by using the `metaset` command:

```
metaset -s <DiskSet Name> -d -m <Mediator Host List>

-s DiskSet Name to add mediators for
-d Delete mediators
-m Mediator host names to delete
```

Step 11 - Configure dual string mediators

- If there are disksets which are configured across exactly two disk enclosures which are connected to two cluster nodes, use the `metaset` command to configure the dual string mediators for the diskset:

```
# metaset -s nfs_data_1 -a -m mars venus
```

- Use the `medstat` command to check the status of the mediators:

```
# medstat -s nfs_data_1
Mediator                Status Golden
mars                    OK     No
venus                   OK     No
#
```

- Mediators may be deleted by using the `metaset` command:

```
# metaset -s nfs_data_1 -d -m mars venus
# medstat -s nfs_data_1
No mediator hosts configured for set “nfs_data_1”.
# metastat -s nfs_data_1 -a -m mars venus
#
```

- To repair bad mediators, delete and the re-add the mediators using the `metaset` command

CHAPTER 4

Configuring Veritas Volume Manager with Sun Cluster 3.0

Objectives

Purpose

In this chapter, we will learn how to install and configure Veritas Volume Manager (VxVM) with Sun Cluster 3.0

Prerequisites

Basic understanding of disk management principles

Previous experience with Veritas Volume Manager

Objectives

Upon completion of this chapter, the participant will be able to:

- Describe the features and basic concepts of Veritas Volume Manager
- Describe how to install and configure Veritas Volume Manager for use in Sun Cluster 3.0

Objectives

- Describe the features and basic concepts of Veritas Volume Manager (VxVM)
- Describe how to install and configure Veritas Volume Manager for use in Sun Cluster 3.0

An Introduction to Veritas Volume Manager

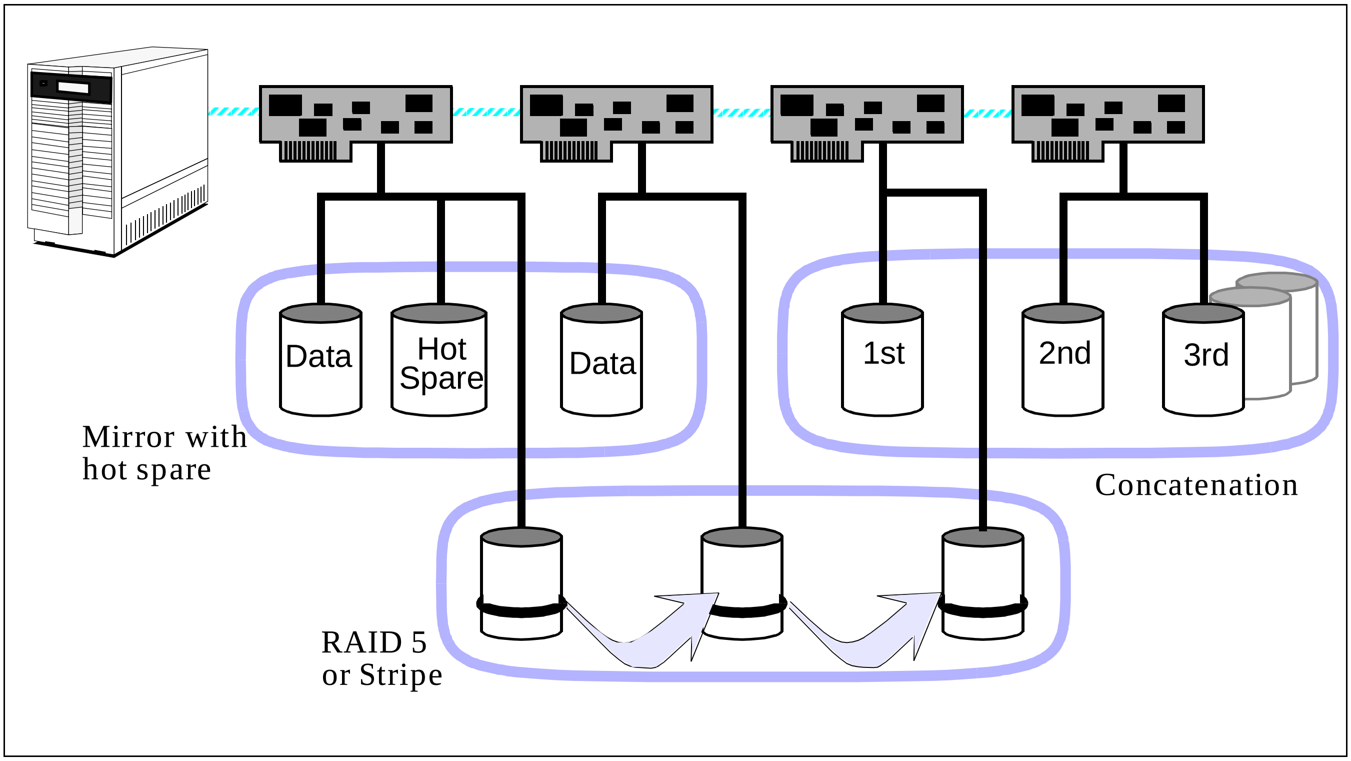
Overview and Features

Veritas Volume Manger (VxVM) is a program developed by Veritas Software which provides management of storage resources on critical servers. Sun Cluster 3.0 supports VxVM 3.0.4.

Veritas Volume Manager 3.0.4 supports the following features:

- **Disk Mirroring (RAID 1)** - multiple copies of data are maintained on different physical devices; two-way and three-way mirrors are supported. Can be combined with disk striping (RAID 1+0 or RAID 0+1)
- **Disk Striping (RAID 0)** - data is interlaced among multiple physical devices
- **Disk Concatenation (RAID 0)** - two or more physical devices are combined into a single logical device
- **Disk Mirroring and Striping (RAID 1+0, 0+1)** - disk mirroring and striping can be combined to provide both high performance and high availability
- **RAID 5** - data and parity information is interlaced among multiple physical devices
- **Expandable UFS file systems** - VxVM allows increasing the size of a Unix file system without having to recreate the entire file system
- **Disk Groups** - Drives can be logically separated into disk groups, allowing the cluster to manipulate the logical set of drives and volumes as a single entity
- **Hot Spares** - Drives can be assigned to be automatically substituted for a failed component of a mirrored or RAID 5 device. Even without an explicitly assigned hot spare disk, VxVM can use unallocated space to for data reconstruction in the event of a disk failure (Hot Relocation).

An Introduction to Veritas Volume Manager



- Veritas Volume Manager (VxVM) supports:
  - Disk Mirroring (RAID 1)
  - Disk Striping (RAID 0)
  - Disk Concatenation (RAID 0)
  - Disk Mirroring and Striping (RAID 1+0, RAID 0+1)
  - RAID-5
  - Expandable UFS
  - Disk groups
  - Hot Spares
- Sun Cluster 3.0 supports VxVM 3.0.4

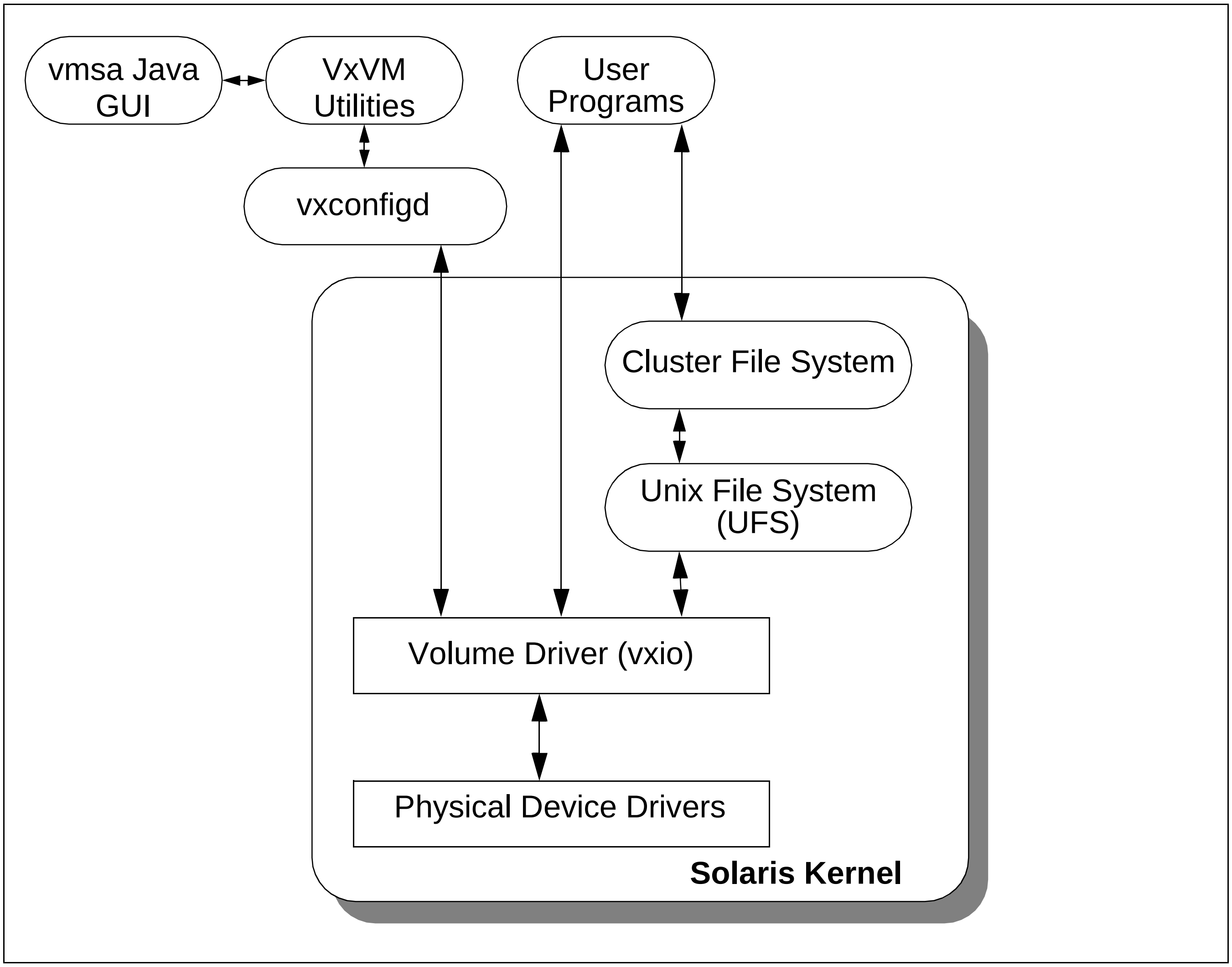
An Introduction to Veritas Volume Manager

Architecture

VxVM consists of a set of utilities, a Java-based GUI, a database to configure monitor and manage the system and a volume driver. The principal component, the volume driver, resides above the physical device drivers and below the file systems (regular and cluster) and/or user applications. It performs requested I/O and configuration changes. File systems and applications access *volumes* instead of traditional UNIX physical disk partitions.

The volume configuration daemon (`vxconfigd`) manages the configuration database. Volume management utilities make calls through `vxconfigd`. `vxconfigd` validates the requests and updates the database and the volume driver automatically.

An Introduction To Veritas Volume Manager



- VxVM consists of a set of utilities, a Java-GUI, a configuration database and a volume driver (`vxio`)
- The volume driver resides between the file system drivers and the physical device drivers
- The volume configuration daemon (`vxconfigd`) manages the configuration database

VxVM Objects

Volume Manager defines a hierarchy of storage. This hierarchy is illustrated on the following page. The components of this hierarchy are:

- **VM Disks** - When a disk partition is placed under Volume Manager control, a VM disk is assigned to the partition. A VM disk has a one-to-one relationship with a partition. A VM disk is assigned a disk media name, which is supplied when the physical partition is placed under Volume Manager control. The default disk media name for Volume Manager is `<DiskGroup>##` (for example `scdg101` for the first disk in the scdg1 disk group). Usually, *entire disks* are placed under Volume Manager control (this is recommended to make it easier to administer), in which case a VM disk has a one-to-one relationship with a physical disk.

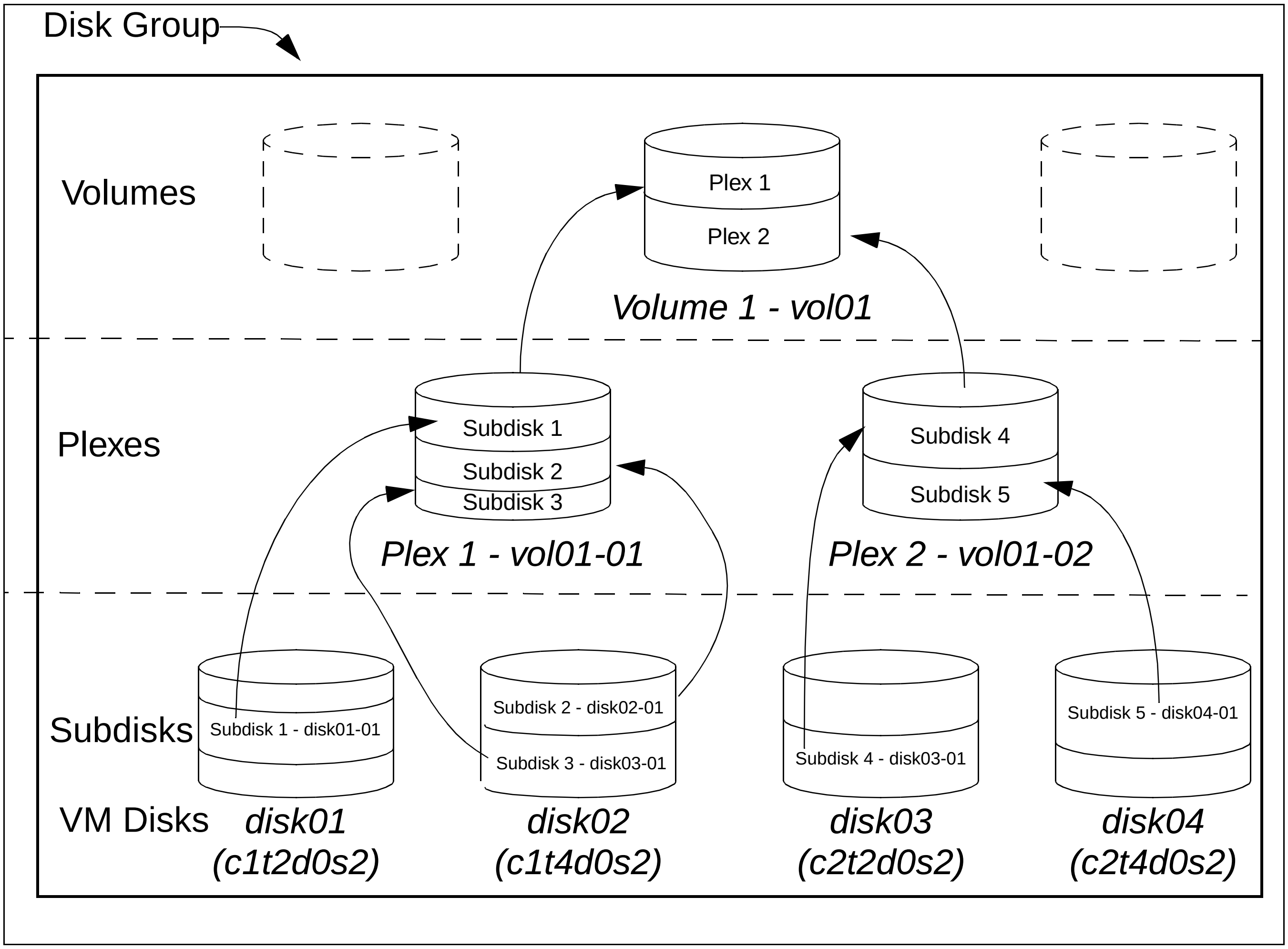
- **Subdisks** - VM disks are subdivided into an entity called subdisks. Subdisks are contiguous regions on a VM disk and are the fundamental building block for constructing higher level Volume Manager objects.

- **Plexes** - Plexes are mirrors or copies of data. Each volume must consist of at least one plex (if a volume is made up of a single plex, then there is only one copy of the data and it is *not* mirrored). If you are mirroring or logging, then you would configure multiple plexes. Plexes contain one or more subdisks, which can reside on different VM disks within the disk group.

- **Volumes** - Volumes are the top-level entities of Volume Manager. They can be thought of as a *logical* partition. Volumes are made up of one or more plexes. File systems may be built on volumes or they may be used raw by an application program (such as a database system).

- **Disk Groups** - A disk group is collection of related volumes, subdisks, plexes and VM disks. VM disks can only be assigned to one disk group and volumes and plexes may **not** span multiple disk groups. Disk groups (and all that they contain) may be migrated from one host to another via a process called *importing* and *deporting.*

VxVM Objects



- Disk groups are created as a container to hold VM disks, subdisks, plexes and volumes
- Volumes are the top level entity which may be used by application programs and/or file systems
- Volumes are made up of one or more plexes (mirrors), which, in turn, are made up of one or more subdisks. Subdisks are contiguous pieces of the VM disks (physical disk partitions)

How VxVM Manages Physical Disks

When physical disks are placed under Volume Manager control, Volume Manager will initialize a small area of the disk as the *Private Region*, leaving the rest of the disk as the *Public Region*.

The private region is used by Volume Manager to hold administrative data, including copies of configuration database for all the disks in the disk group.

The public region is the area of the disk which Volume Manager uses for general space allocation. Contiguous parts of the public region are what Volume Manager allocates as subdisks.

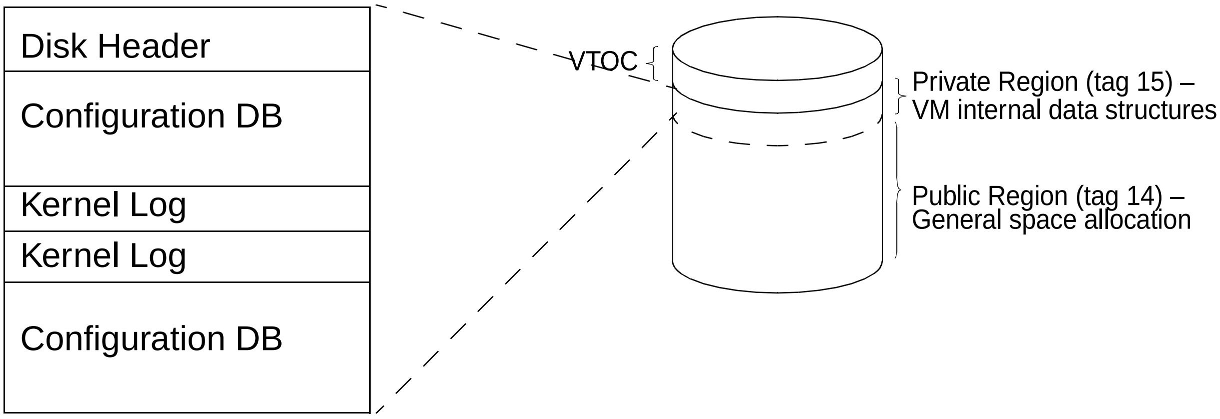
There are two ways to bring physical disks under Volume Manager control: *encapsulation* and *initialization*.

- **Encapsulation** - Encapsulation will preserve any existing data on the disk. A private region will be created on an unused portion of the disk, while maintaining any existing partitions as the public region of the disk. In order for encapsulation to succeed, the disk must have 2 free cylinders (to hold the private region) at either the beginning or end of the disk and two free partitions (slices).

  If desired, the boot disk may be encapsulated and placed under Volume Manager control (this will allow the boot disk to be mirrored to enhance system availability).

- **Initialization** - If a disk is initialized when it is brought under Volume Manager control, the existing VTOC on the disk will be wiped out and new VTOC installed. Volume Manager will create a small partition for the private region (usually slice 3), allocating the rest of the disk in another partition (usually slice 4) for the public region. Needless to say, any existing data on the disk will be lost.

How VxVM Manages Physical Disks



- When disks are placed under VxVM control, VxVM will create a small *Private Region* and a large *Public Region* on the disk
  - The *Private Region* contains administrative data used by VxVM to manage the disk group
  - The *Public Region* is used by VxVM to allocate subdisks for use in VxVM plexes and volumes
- Physical disks can either be *encapsulated* or *initialized* when placed under VxVM control
  - *Encapsulation* preserves any existing data on the disk. There must be enough space to create a private region.
  - *Initialization* will install a new VTOC, creating a small slice (default is slice 3) for the private region and a large slice (default is slice 4) for the public region

The `rootdg` Disk Group

Configuration of disk groups in Volume Manager is completely up to the user. Which physical disks to allocate, the subdisks, plexes and volumes in the disk groups and even the name of the disk group itself can be configured by the administrator. There is, however, one requirement: on every node of the cluster, there must be a “local” disk group named `rootdg`. This disk group is required by Volume Manager and must have the following characteristics:

- Each node must have a `rootdg` disk group consisting of at least one VM disk
- If the boot disk is encapsulated, it must be in the `rootdg` disk group
- The `rootdg` disk group cannot be migrated from one host to another

The rootdg Disk Group

- Disk Group configuration is completely flexible; The contents and the name of the disk group can be configured by the VxVM administrator
- Each node **must** have a disk group called rootdg with the following characteristics:
    - Each node must have a rootdg disk group consisting of at least one VM disk
    - If the boot disk is encapsulated, it must be placed in the rootdg disk group
    - The rootdg disk group cannot be migrated from one host to another

VxVM Namespace

The naming of disk groups, VM disks and volumes is completely user configurable. Volumes are the top level, Solaris accessible objects, thus in order to create and use file systems or have an application use raw VxVM volumes, both the disk group name and the volume name must be known. The method by which the VxVM namespace is presented to Solaris is via the following device path:

```
/dev/vx/[r]dsk/<DiskGroupName>/<VolumeName>
```

For example, to construct and mount a file system on a VxVM volume named `vol-01` in the `scdg1` disk group, the following `newfs` and `mount` commands would need to be executed:

```
newfs /dev/vx/rdsk/scdg1/vol-01

mount -o global,syncdir,logging /dev/vx/dsk/scdg1/vol-01 \
     /global/web_data
```

VxVM Namespace

- VxVM allows the administrator to configure the naming of disk groups and volumes
- The top level object is the volume, thus in order to utilize the volume from an OS or application level, both the disk group and the volume name must be known
- VxVM volumes are presented to Solaris as /dev/vx/[r]dsk/*<DiskGroupName>*/*<VolumeName>*:

```
# ls /dev/vx/dsk/scdg1
vol-01  vol-02
```

The example above show the "contents" of the disk group scdg1. There are two volumes, named vol-01 and vol-02.

To create and mount a file system on vol-01:

```
# newfs /dev/vx/rdsk/scdg1/vol-01
newfs: construct a new file system /dev/vx/rdsk/scdg1/vol-01: (y/n)? y
/dev/vx/rdsk/scdg1/vol-01:   204800 sectors in 100 cylinders of 32 tracks, 64
sectors
        100.0MB in 7 cyl groups (16 c/g, 16.00MB/g, 7680 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 32864, 65696, 98528, 131360, 164192, 197024,
# mkdir /global/web_data
# mount -o global,syncdir,logging /dev/vx/dsk/scdg1/vol-01 /global/web_data
```

VxVM Command Summary

The following commands are used by VxVM to manipulate VxVM objects:

Command	Description
vxassist	Used to create, relayout, convert, mirror, backup, grow, shrink, delete and move volumes
vxdctl	Controls the volume configuration daemon (vxconfigd)
vxdg	Manages VxVM disk groups
vxdisk	Defines and manages VM disks
vxdiskadd	Adds physical disks for use with VxVM
vxdiskadm	Menu-driven VM disk administrator
vxedit	Sets or changes attributes of VxVM objects
vxinfo	Prints the accessibility and usability of volumes
vxinstall	Used to initialize volume manager
vxlicense	License key utility
vxmake	Creates VxVM objects
vxmend	Repairs simple problems in configuration records
vxplex	Performs VxVM operations on plexes
vxprint	Displays VxVM configuration
vxrecover	Performs volume recovery tasks
vxrelayout	Converts online storage from one layout to another
vxsd	Performs VxVM operations on subdisks
vxstat	VxVM statistics management utility
vxtask	Lists and administers VxVM tasks (long running operations)
vxtrace	Traces operations on volumes
vxvol	Performs VxVM operations on volumes

VxVM commands are located in `/usr/sbin`.

VxVM Command Summary

- The following commands can be used to manipulate VxVM objects:
  - vxassist
  - vxdctl
  - vxdg
  - vxdisk
  - vxdiskadd
  - vxdiskadm
  - vxedit
  - vxinfo
  - vxinstall
  - vxlicense
  - vxmake
  - vxmend
  - vxplex
  - vxprint
  - vxrecover
  - vxrelayout
  - vxsd
  - vxstat
  - vxtask
  - vxtrace
  - vxvol
- All VxVM commands are located in `/usr/sbin`

Installing and Configuring VxVM in Sun Cluster 3.0

Installing and configuring VxVM 3.0.4 in Sun Cluster 3.0 consists of the following steps:

1. Disable dynamic multipathing support
2. Configure the disks to be used by the `rootdg` disk group
3. Install the appropriate packages for VxVM 3.0.4
4. Check the major number for the `vxio` device on all nodes
5. License VxVM
6. Initialize the `rootdg` disk group
7. Create and populate VxVM disk groups to house data service data
8. Create VxVM volumes in each disk group
9. Register the disk group with the cluster framework

Installing and Configuring VxVM in Sun Cluster 3.0

1. Disable dynamic multipathing support
2. Configure the disks to be used by the `rootdg` disk group
3. Install the appropriate packages for VxVM 3.0.4
4. Check the major number for the `vxio` device on all nodes
5. License VxVM
6. Initialize the rootdg disk group
7. Create and populate VxVM disk groups to house data service data
8. Create VxVM volumes in each disk group
9. Register the disk group with the cluster framework

Step 1 - Disable dynamic multipathing support

Sun Cluster 3.0 does not currently support the Dynamic MultiPathing (DMP) feature of VxVM. To prevent VxVM from enabling this feature when it is installed, perform the following steps **before** you add any VxVM packages:

1. Create a `vx` directory in `/dev`

```
mkdir /dev/vx
```

2. Create a symbolic link between `/dev/dsk` and `/dev/vx/dmp`

```
ln -s /dev/dsk /dev/vx/dmp
```

3. Create a symbolic link between `/dev/rdsk` and `/dev/vx/rdmp`

```
ln -s /dev/rdsk /dev/vx/rdmp
```

Step 1 - Disable dynamic multipathing support

- Sun Cluster 3.0 does not currently support VxVM DMP
- To make sure that DMP is not enabled when VxVM is installed:

```
# mkdir /dev/vx
# ln -s /dev/dsk /dev/vx/dmp
# ln -s /dev/rdsk /dev/vx/rdmp
```

Step 2 - Configure the disks to be used by the `rootdg` disk group

Any disks that are going to used in the `rootdg` disk group must have the following characteristics:

- Slice 2 must encompass the entire disk
- If the disk is to be encapsulated, there must be 2 free cylinders available either at the end or the beginning of the disk as well as 2 unused slices

Step 2 - Configure the disks to be used by the `rootdg` disk group

- Make sure that any disks that are going to be used in the rootdg disk group meet the following criteria:
  - Slice 2 encompasses the entire disk
  - If the disk is going to be encapsulated, there must be 2 unused slices and as well as 2 unassigned cylinders at either the beginning or the end of the disk

Step 3 - Install the appropriate packages for VxVM 3.0.4

Use the pkgadd utility to install the VxVM packages on all nodes of the cluster. VxVM is made up of the following packages:

- VRTSvxvm - Veritas Volume Manager binaries (required)
- VRTSvmdev - Veritas Volume Manager header files and libraries (required)
- VRTSvmman - VxVM man pages (optional)
- VRTSvmsa - VxVM Storage Administratior GUI (optional)
- VRTSvmdoc - VxVM user documentation (optional)

After installing the packages add /opt/VRTSvxva/man to your MANPATH and make sure that /usr/sbin and /opt/VRTSvmsa/bin (if the Storage Administrator package was installed) in in your PATH.

Step 3 - Install the appropriate packages for VxVM 3.0.4

```
# cd <Location of VxVM CD image>/????
# pkgadd -d . VRTSvxvm VRTSvmdev VRTSvmman

Processing package instance <VRTSvxvm> from </cdrom/3.0.4>

VERITAS Volume Manager, Binaries
(sparc) 3.0.4,REV=08.11.1999.01.12
Copyright (c) 1990-1999 VERITAS Software Corporation.
...
<Various pkgadd messages and prompts>
...
Copy //sbin/vxconfigd.SunOS_5.8 to //sbin/vxconfigd...
NOTICE: DMP driver was previously disabled, so will not be enabled ....

Adding vxio driver for SunOS version 5.8...
Adding vxspec driver for SunOS version 5.8...
Adding vxspec lines to //etc/devlink.tab...
Running /usr/sbin/devlinks -r / -t //etc/devlink.tab ...
Adding vxio vxspec lines to //etc/system...
Copy libthread.so.1 to //etc/vx...
Copy libc.so.1 to //etc/vx...

Installation of <VRTSvxvm> was successful.

Processing package instance <VRTSvmdev> from </cdrom/3.0.4>
...
<Various pkgadd messages and prompts>
...
```

- Use the pkgadd utility to install the VxVM packages from the Volume Manager 3.0.4 CD-ROM image

- Add `/opt/VRTSvxvm/man` to your MANPATH. Verify that `/usr/sbin` and `/opt/VRTSvmsa/bin` are in your PATH.

Step 4 - Check the major number for the `vxio` device on all nodes

In order for a VxVM volume to be globally available, the major number for the vxio pseudo-device must be identical on all nodes of the cluster. In order to verify this, check the `vxio` entry in the `/etc/name_to_major` file on all nodes. If any nodes have a different major number assigned to the vxio pseudo-driver, adjust the entry in the `name_to_major` file on the differing nodes.

> **Note:** When adjusting the entry, make sure that you are not using a major number assigned to another device. Scan the entire `/etc/name_to_major` file to avoid any major number conflicts within the file.

Step 4 - Check the major number for the vxio device on all nodes

- The major number assigned to the vxio pseudo-device should be identical on all nodes. Check the vxio entry in the /etc/name_to_major file on all nodes of the cluster to verify the assigned major number.
- If the major numbers differ between the nodes of the cluster, check the /etc/name_to_major files to determine a major number which may be used on all nodes of the cluster and edit the /etc/name_to_major files accordingly:

```
# hostname                               # hostname
venus                                    mars

# grep vxio /etc/name_to_major           # grep vxio /etc/name_to_major
vxio 60                                  vxio 63
# grep 63 /etc/name_to_major             # grep 63 /etc/name_to_major
vxspec 63                                vxio 63
# grep 60 /etc/name_to_major             # grep 60 /etc/name_to_major
vxio 60                                  vxdmp 60
# grep 64 /etc/name_to_major             # grep 64 /etc/name_to_major
#                                        vxspec 64
# grep 65 /etc/name_to_major             # grep 65 /etc/name_to_major
#                                        #

<Since neither node has an entry for 65, we can use this
  number as the major number for the vxio
  pseudo-device>

# vi /etc/name_to_major                  # vi /etc/name_to_major
<Set the vxio entry to 65>               <Set the vxio entry to 65>

# grep vxio /etc/name_to_major           # grep vxio /etc/name_to_major
vxio 65                                  vxio 65
```

Step 5 - License VxVM

Sun SparcStorage Arrays (SSAs) and Sun StorEdge A5x00 come with an embedded VxVM license. If there are any SSA's or A5x00's connected to the host, licensing of VxVM is automatic.

If there are no SSA's or A5x00s in the configuration, VxVM must be explicitly licensed. To license VxVM:

1. Obtain a licence key for VxVM
2. Use `vxlicense` to install the license key on each node

```
vxlicense -c
```

Step 5 - License VxVM

- ➤ If there are no SSAs or StorEdge A5x00s in the configuration, VxVM requires the installation of a license key
- ➤ To install a licence key for VxVM, use the `vxlicense` command:

```
# vxlicense -c
Please enter your key: 9937 2049 0626 6870 1015 999

vrts:vxlicense: INFO: Feature name: CURRSET [95]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Jan 02 02:00:00 2000 (40.7 days from
now)
vrts:vxlicense: INFO: Release Level: 20
vrts:vxlicense: INFO: Machine Class: All
vrts:vxlicense: INFO: Key successfully installed in /etc/vx/elm/95.
```

Step 6 - Initialize the `rootdg` Disk Group

The next step is to initialize the `rootdg` disk group. Each node is required to have its own `rootdg`. There are two options for configuring a rootdg disk group on a node:

1. Encapsulate or initialize a non-root disk as the `rootdg` disk group

2. Encapsulate the root disk (automatically becomes a member of the `rootdg` disk group)

Step 6 - Initialize the `rootdg` disk group

- ➤ Each node requires a `rootdg` disk group
- ➤ There are two options for creating a `rootdg` disk group:
  - ➤ Encapsulate or initialize a non-root disk (or disks) as the `rootdg` disk group
  - ➤ Encapsulate the root disk

Step 6 - Initialize the rootdg disk group (Continued)

Option 1 - Encapsulate or Initialize a non-root disk as the rootdg disk group

The procedure to use a non-root disk as the rootdg disk group is:

1. On each node of the cluster, execute vxinstall
2. At the main menu of vxinstall, choose Custom Install
3. When given the option, choose not to encapsulate the boot disk
4. vxinstall will present you with a list of disks on each controller:
   a. If the disk or disks you want to use ***is not*** on the controller being presented by vxinstall, choose "Leave these disks alone" for the controller
   b. If the disk or disks you want to use ***is*** on the controller being presented by vxinstall, choose the "Install one disk at a time" option. When the chosen disk or disks is presented, choose the appropriate option for the disk (Encapsulate or Initialize)
5. Repeat step 4 for all the disk controllers installed in the node
6. Do **not** have vxinstall automatically shutdown and reboot the node
7. After exiting vxinstall, use the scshutdown utility to shutdown and reboot the cluster node

Option 1 - Encapsulate or initialize a non-root disk as the `rootdg` disk group

```
# vxinstall
...
<vxinstall Messages>
...
Volume Manager Installation Options
Menu: VolumeManager/Install

 1      Quick Installation
 2      Custom Installation

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 2

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom

  The c0t0d0 disk is your Boot Disk.  You can not add it as a new
  disk.  If you encapsulate it, you will make your root file system
  and other system areas on the Boot Disk into volumes.  This is
  required if you wish to mirror your root file system or system
  swap area.

Encapsulate Boot Disk [y,n,q,?] (default: n) n

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom/c0
Generating list of attached disks on c0....

<excluding root disk c0t0d0>
No disks were found attached to controller c0 !
Hit RETURN to continue.

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom/c1
Generating list of attached disks on c1....


  The Volume Manager has detected the following disks on controller c1:

  c1t3d0 c1t4d0 c1t5d0

Hit RETURN to continue.


Installation options for controller c1
Menu: VolumeManager/Install/Custom/c1

 1      Install all disks as pre-existing disks. (encapsulate)
 2      Install all disks as new disks. (discards data on disks!)
 3      Install one disk at a time.
 4      Leave these disks alone.

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 3
```

Option 1 - Initialize or encapsulate a non-root disk as the `rootdg` disk group (Continued)

The following page continues the example `vxinstall` session which initializes a non-root disk as the `rootdg` disk group.

Option 1 - Encapsulate or initialize a non-root disk as the `rootdg` disk group (Continued)

```
Installation options for disk c1t3d0
Menu: VolumeManager/Install/Custom/c1/c1t3d0

 1      Install as a pre-existing disk. (encapsulate)
 2      Install as a new disk. (discards data on disk!)
 3      Leave this disk alone.

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 2


Are you sure (destroys data on c1t3d0) [y,n,q,?] (default: n) y

Enter disk name for c1t3d0 [<name>,q,?] (default: disk01) disk01

Installation options for disk c1t4d0
Menu: VolumeManager/Install/Custom/c1/c1t4d0

 1      Install as a pre-existing disk. (encapsulate)
 2      Install as a new disk. (discards data on disk!)
 3      Leave this disk alone.

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 3

...
<vxinstall will present similar menus for other disks and controllers on the node>
<All other disks were left alone>
...

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom

  The following is a summary of your choices.

        c1t3d0  New Disk

Is this correct [y,n,q,?] (default: y) y

  The Volume Manager is now reconfiguring (partition phase)...

  Volume Manager: Partitioning c1t3d0 as a new disk.

  The Volume Manager is now reconfiguring (initialization phase)...

  Volume Manager: Adding disk01 (c1t3d0) as a new disk.

  The Volume Daemon has been enabled for transactions.


  The system now must be shut down and rebooted in order to continue
the reconfiguration.

Shutdown and reboot now [y,n,q,?] (default: n) n

# /usr/cluster/bin/scshutdown -o node -y -g0 -r
```

Step 6 - Initialize the rootdg disk group (Continued)

Option 2- Encapsulate the root disk

To create a `rootdg` disk group by encapsulating the root disk:

1. On each node of the cluster, execute `vxinstall`
2. At the main menu, choose `Custom Install`
3. When given the option, choose to encapsulate the boot disk. Specify a unique boot disk name on each node.
4. `vxinstall` will present you with a list of disks on each controller, choose the "`Leave these disks alone`" option for the rest of the disks in the node (unless there are specific disks you want to add to the `rootdg` disk group)
5. Do **not** have `vxinstall` automatically shutdown and reboot the node

(Continued ...)

Option 2 - Encapulate the root disk

- Run `vxinstall`:

```
# vxinstall
Volume Manager Installation Options
Menu: VolumeManager/Install

 1      Quick Installation
 2      Custom Installation

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 2

  The c0t0d0 disk is your Boot Disk.  You can not add it as a new
  disk.  If you encapsulate it, you will make your root file system
  and other system areas on the Boot Disk into volumes.  This is
  required if you wish to mirror your root file system or system
  swap area.

Encapsulate Boot Disk [y,n,q,?] (default: n) y

Enter disk name for  [<name>,q,?] (default: rootdisk) root-n1

  The c0t0d0 disk has been configured for encapsulation.

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom/c0
Generating list of attached disks on c0....

- <excluding root disk c0t0d0>
  No disks were found attached to controller c0

Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom/c1
Generating list of attached disks on c1....

  The Volume Manager has detected the following disks on controller c1:

  c1t3d0 c1t4d0 c1t5d0

Installation options for controller c1
Menu: VolumeManager/Install/Custom/c1

 1      Install all disks as pre-existing disks. (encapsulate)
 2      Install all disks as new disks. (discards data on disks!)
 3      Install one disk at a time.
 4      Leave these disks alone.

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 4
...
<vxinstall will repeat this menu for the rest of the controllers installed on the node>
...
Volume Manager Custom Installation
Menu: VolumeManager/Install/Custom

  The following is a summary of your choices.

        c0t0d0  Encapsulate


Is this correct [y,n,q,?] (default: y) y

  The system now must be shut down and rebooted in order to continue
the reconfiguration.

Shutdown and reboot now [y,n,q,?] (default: n) n
```

Option 2 - Encapsulate the root disk (Continued)

6. Modify the entry in the /etc/vfstab file for the /global/.devices filesystem. Make sure that the actual physical partition address (c#t#d#s#) is used and **not** the DID device (If there is a commented-out entry for the /globaldevices file system, this entry should be the c#t#d#s# that should be used for /global/.devices)

7. Use shutdown (or scshutdown, if you are performing the encapsulation on all nodes simultaneously) to shutdown the node (or cluster) and **(IMPORTANT)** boot the node into **non-**clustered mode (boot -x). VxVM performs the encapsulation of the boot disk, and the node reboots itself into cluster mode as part of this process.

```
scshutdown -y -gN

-y  Preanswers yes to confirmation question
-gN Specifies the grace period of N seconds before starting
    shutdown
```

```
boot -x

-x specifies that this node should **not** boot as part of
   cluster
```

During this reboot, all but one node will fail to fsck and mount the /global/.devices/node@*X* file system. The following error message may appear:

```
WARNING - Unable to repair the /global/.devices/node@1
filesystem.
Run fsck manually (fsck -F ufs /dev/vx/rdsk/rootdisk3vol).
Exit the shell when done to continue the boot process.
Type control-d to proceed with normal setup.
(or give root password for system maintenance):
```

Type Control-D to proceed. Do not run fsck manually.

(Continued ...)

Option 2 - Encapsulate the root disk (Continued)

- ➤ Modify the `/etc/vfstab` file to reflect the actual physical device instead of the DID device for the `/global/.devices/node@`*`X`* file system

Before modification:

```
#device            device             mount          FS    fsck  mount    mount
#to mount          to fsck            point          type  pass  at boot  options
#
...
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6 /usr            ufs    1     yes       -
#/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /globaldevices ufs    2     yes       -
...
/dev/did/dsk/d1s4 /dev/did/rdsk/d1s4 /global/.devices/node@1 ufs 2 no global
```

After modification:

```
#device            device             mount          FS    fsck  mount    mount
#to mount          to fsck            point          type  pass  at boot  options
#
...
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6 /usr            ufs    1     yes       -
#/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /globaldevices ufs    2     yes       -
...
/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /global/.devices/node@1 ufs 2 no global
```

- ➤ Reboot the node in non-clustered mode:

```
# shutdown -y -g0
INIT New run level: 0
The system is coming down
...
<Various shutdown messages>
...
Program Terminated
ok boot -x
```

Option 2 - Encapsulate the root disk (Continued)

8. Use the `mount` command to determine which `/global/.devices` file system was mounted and then unmount that single file system.

9. Make the base minor number for `rootdg` unique on all nodes of the cluster. Suggestion: use 100 * NodeNumber as the base minor number for `rootdg`.

```
vxdg reminor <NewBaseMinorNum>
```

10. If there was a separate /usr partition, manually update the minor number for the `/usr` partition:

    a. Remove the existing device nodes

    b. Use `vxprint` to obtain the newly assigned minor number for the `usr` volume

```
vxprint -l -v <NameofVol>
```

    c. Manually create the device nodes using the `mknod` command

```
mknod <DeviceNode> b|c <MajorNum> <MinorNum>

<DeviceNode> - Device node to create
b|c - specifies block (b) or character (c)
<MajorNum> - Major number for device node
<MinorNum> - Minor number for device node
```

11. Repeat Step 10 for the `/var` partition.

Option 2 - Encapsulate the root disk (Continued)

- Unmount the single /global/.devices file system that was successfully mounted
- Reminor the rootdg disk group to be unique on each node:

```
On Node 1
# vxdg reminor 100

On Node 2
# vxdg reminor 200
```

- If there was a separate `/usr` partition, the `usr` volume must be reminored manually:

```
# rm /dev/vx/dsk/usr /dev/vx/dsk/rootdg/usr
# rm /dev/vx/rdsk/usr /dev/vx/rdsk/rootdg/usr
# vxprint -l -v usr
Disk group: rootdg

Volume:   usr
info:     len=1024480
type:     usetype=fsgen
state:    state=ACTIVE kernel=ENABLED cdsrecovery=0/0 (clean)
assoc:    plexes=usr-01
policies: read=ROUND exceptions=GEN_DET_SPARSE
flags:    open writeback
logging:  type=NONE
apprecov: seqno=0
recov_id=0
device:   minor=103 bdev=65/103 cdev=65/103 path=/dev/vx/dsk/rootdg/usr
perms:    user=root group=root mode=0600
# grep vxio /etc/name_to_major
vxio 65
# mknod /dev/vx/dsk/usr b 65 103
# mknod /dev/vx/dsk/rootdg/usr b 65 103
# mknod /dev/vx/rdsk/usr c 65 103
# mknod /dev/vx/rdsk/rootdg/usr c 65 103
```

- Repeat the preceding step for the `/var` partition

Option 2 - Encapsulate the root disk (Continued)

12. (Necessary only if you didn’t assign a unique boot disk name on each node during Veritas install; see Page 4-36.) If the name of the VxVM volume used for the `/global/.devices/node@X` file system is not unique on the different nodes of the cluster, change it accordingly:

    a. Use `vxedit` to rename the volume name

    ```
    vxedit rename <OldVolumeName> <NewVolumeName>
    ```

    b. Edit the `/etc/vfstab` files on all nodes to reflect the new volume names

13. Use `shutdown` (or `scshutdown` if you are performing the encapsulation on all nodes of the cluster simultaneously) to reboot the node or cluster into clustered mode:

    ```
    [sc]shutdown -y -gN “Shutdown Message”

    -y  Preanswers yes to confirmation question
    -gN Specfies the grace period of N seconds before starting
        shutdown
    ```

Option 2 - Encapsulate the root disk (Continued)

- (Unnecessary if you previously specified uniques names; see Page 4-36.) Verify that the `/global/.devices/node@X` volume names are unique on all nodes of the cluster:

```
# mount | grep "/global/.devices"
/global/.devices/node@2 on /dev/vx/dsk/rootdisk4vol
read/write/setuid/global/largefiles on Tue Nov 23 15:29:55 1999
/global/.devices/node@1 on /dev/vx/dsk/rootdisk4vol
read/write/setuid/global/largefiles on Tue Nov 23 15:29:55 1999
```

In this case, notice that the volume for node 1 (`/global/.devices/node@1`) and the volume for node 2 (`/global/.devices/node@2`) are identical (`rootdisk4vol`)

- To rename the volume, use the vxedit command:

```
# vxedit rename rootdisk4vol global-n1vol    (On Node 1)

# vxedit rename rootdisk4vol global-n2vol    (On Node 2)
```

- Make sure to update the `/etc/vfstab` file on each node to reflect the new volume name
- Reboot the cluster into cluster mode

```
# shutdown -y -g0
INIT New run level: 0
The system is coming down
...
<Various shutdown messages>
...
ok boot
```

Step 7 - Create and populate VxVM disk groups to house data service data

Use the `vxdiskadd` command or the `vxdiskadm` utility to initialize (or encapsulate) disks in the multi-hosted storage into disk groups for use by the cluster’s data services.

```
vxdiskadm

   OR

vxdiskadd <DiskPattern>

<DiskPattern> - portion of a disk address (c#t#d#)
                to add to a disk group.  For example:
                     c1 - will add all disks on controller 1
                     c1t0d0 - will add just c1t0d0
```

This step may also be done by using the Volume Manager Storage Administrator GUI.

Step 7 - Create and populate VxVM disk groups to house data service data

- Use `vxdiskadd` or `vxdiskadm` to add disks to existing or new disk groups for use by the cluster's data services:

```
# vxdiskadm
Volume Manager Support Operations
Menu: VolumeManager/Disk

 1      Add or initialize one or more disks
 2      Encapsulate one or more disks
 3      Remove a disk
 4      Remove a disk for replacement
 ...
<Other menu items>
...
 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform: 1

Add or initialize disks
Menu: VolumeManager/Disk/AddDisks
...
Select disk devices to add:
[<pattern-list>,all,list,q,?] c1t2d0

  Here is the disk selected.  Output format: [Device_Name]

  c1t2d0

Continue operation? [y,n,q,?] (default: y) y
...
Which disk group [<group>,none,list,q,?] (default: rootdg) scdg1

  There is no active disk group named scdg1.

Create a new group named scdg1? [y,n,q,?] (default: y) y

Use a default disk name for the disk? [y,n,q,?] (default: y) y

Add disk as a spare disk for scdg1? [y,n,q,?] (default: n) n

  A new disk group will be created named scdg1 and the selected disks
  will be added to the disk group with default disk names.

  c1t2d0

Continue with operation? [y,n,q,?] (default: y) y
...
Encapsulate this device? [y,n,q,?] (default: y) n

  c1t2d0

Instead of encapsulating, initialize? [y,n,q,?] (default: n) y

  Initializing device c1t2d0.

  Creating a new disk group named scdg1 containing the disk
  device c1t2d0 with the name scdg101.

Add or initialize other disks? [y,n,q,?] (default: n)
```

Step 8 - Create VxVM volumes in each disk group

Use the vxassist command or the VMSA GUI to create volumes in each disk group.

Make sure that volumes have the following characteristics:

- Volumes must be mirrored (in different enclosures)
- Create a Dirty Region Log for each volume (makes mirror resync operations faster)

```
vxassist -g <DiskGroup> make <VolumeName> <VolSize> <Attributes>

-g <DiskGroup> - Specifies the disk group to create the volume in
<VolumeName> - Name of volume to create
<VolSize> - Size of the volume to create, in blocks
            (use the k, m or g suffix to specify kilobytes,
            megabytes or gigabytes, respectively)
<Attributes> - Specifies attributes of the volume to create:
               layout=stripe-mirror (Create RAID 0+1 volume)
               layout=mirror-stripe (Create RAID 1+0 volume)
               layout=log           (Add Dirty Region Log)
               columns=N            (Specifies the number of
                                     stripe columns)
               stripeunit=N         (Specifies the blocks in each
                                     chunk)
```

Note: See the `vxassist` man page or the VxVM user documentation for more information on creating volumes

Step 8 - Create VxVM volumes in each disk group

- Use vxassist or the VMSA GUI to create volumes in each disk group:

```
# vxassist -g scdg1 make vol-04 500m layout=mirror-stripe,log column=2 stipeunit=128k
# vxprint -g scdg1 -v vol-04
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
v  vol-04       fsgen        ENABLED  1024000  -        ACTIVE   -       -
# newfs /dev/vx/rdsk/scdg1/vol-04
newfs: construct a new file system /dev/vx/rdsk/scdg1/vol-04: (y/n)? y
/dev/vx/rdsk/scdg1/vol-04:      1024000 sectors in 500 cylinders of 32 tracks, 64
sectors
        500.0MB in 32 cyl groups (16 c/g, 16.00MB/g, 7680 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 32864, 65696, 98528, 131360, 164192, 197024, 229856, 262688, 295520,
 328352, 361184, 394016, 426848, 459680, 492512, 525344, 558176, 591008,
 623840, 656672, 689504, 722336, 755168, 788000, 820832, 853664, 886496,
 919328, 952160, 984992, 1017824,
```

Step 9 - Register the disk group with the cluster framework

Once the disk group has been created and configured, it must be registered with the Sun Cluster 3.0 framework. Registration is required for disk groups except rootdg. Registration of the disk group can be done with the scsetup utility or the scconf command.

```
scsetup

  OR

scconf -a -D type=vxvm,name=<NameofDiskGroup>,nodelist=<NodeList>

       -a - indicates the “add” form of the scconf command
       type=vxvm - specifies a VxVM disk group

       name=<NameofDiskGroup> - specifies the VxVM disk group
                                to be registered

       nodelist=<NodeList> - Specifies a colon separated list of
                             nodes that can import this disk
                             group (based on the topology of
                             the cluster)
```

Step 9 - Register the disk group with the cluster framework

➤ Use `scsetup` or `scconf` to register the disk group with the cluster framework (must be done from the node currently importing the disk group):

```
# scsetup
  *** Main Menu ***

    Please select from one of the following options:

        1) Quorum
        2) Cluster interconnect
        3) Private hostnames
        4) Device groups
        5) New nodes
        6) Other cluster properties

        ?) Help with menu options
        e) Exit

    Option:  4

  *** Device Groups Menu ***

    Please select from one of the following options:

        1) Register a VxVM disk group as a device group
        2) Unregister a VxVM device group
        3) Add a node to a VxVM device group
        4) Remove a node from a VxVM device group
        5) Change key properties of a device group

        ?) Help
        e) Return to the Main Menu

    Option:   1

  >>> Register a VxVM Disk Group as a Device Group <<<

    VERITAS Volume Manager disk groups are always managed by the cluster
    as cluster device groups. This option is used to register a VxVM disk
    group with the cluster as a cluster device group.

    Is it okay to continue (yes/no) [yes]?  yes

    Name of the VxVM disk group you want to register?  scdg1

    Primary ownership of a device group is determined by either
    specifying or not specifying a preferred ordering of the nodes that
    can own the device group. If an order is specified, this will be the
    order in which nodes will attempt to establish ownership. If an order
    is not specified, the first node thet attempts to access a disk in
    the device group becomes the owner.

    Do you want to configure a preferred ordering (yes/no) [yes]?  no

    Are both nodes attached to all disks in this group (yes/no) [yes]?  yes

    Is it okay to proceed with the update (yes/no) [yes]?

scconf -a -D type=vxvm,name=scdg1,nodelist=venus:mars

    Command completed successfully.
```

Step 9 - Register the disk group with the cluster framework (Continued)

If the following error occurs when registering a disk group using scsetup or scconf, there is probably a minor number conflict between the nodes of the cluster:

```
scconf: Failed to add device group - in use
```

This error is caused when the cluster has many disk groups that are "managed" by different nodes of the cluster. When a disk group is created, VxVM chooses a random number (a random multiple of 1000) as that disk group's base minor number. When disk groups are created on different nodes of the cluster, there is a possibly that the nodes independently chose the same base minor number for the different disk groups. When this occurs, one of the conflicting disk groups must be *reminored* prior to registering the disk group with the cluster framework.

To reminor a disk group:

1. Determine which minor numbers are currently being used by the registered disk groups

   ```
   ls -l /dev/vx/dsk/*
   ```

2. Choose an unused base minor number and assign it to the new disk group

   ```
   vxdg reminor <DiskGroupName> <NewBaseMinorNumber>
   ```

3. Register the disk group with the cluster framework

   ```
   scsetup
      OR
   scconf -a -D type=vxvm,name=<DiskGroup>,nodelist=<NodeList>
   ```

Step 9 - Register the disk group with the cluster framework

- Determine which minor numbers are in use in the cluster:

```
# ls -l /dev/vx/dsk/*
/dev/vx/dsk/scdg1:
total 0
brw-------   1 root     root      65,16000 Nov 24 11:03 vol-01
brw-------   1 root     root      65,16001 Nov 24 11:03 vol-02
brw-------   1 root     root      65,16002 Nov 24 11:25 vol-03
brw-------   1 root     root      65,16003 Nov 24 11:25 vol-04

/dev/vx/dsk/scdg2:
total 0
brw-------   1 root     root      65,43000 Nov 23 19:12 vol01
brw-------   1 root     root      65,43001 Nov 23 19:12 vol02
brw-------   1 root     root      65,43002 Nov 23 19:12 vol03
```

The example above shows that base minor numbers 16000 and 43000 are in use. You may also want to check the local `/dev/vx/dsk/<DiskGroup>` directories on each node to see what base minor number are being used by disk groups that have **not** been registered with the cluster framework as well.

- After selecting a new base minor number to use for the disk group, use the vxdg command to reminor the disk group:

```
# vxdg reminor scdg3 17000
```

- Register the disk group with the cluster framework with `scsetup` or `scconf`:

```
# scconf -a -D type=vxvm,name=scdg3,nodelist=mars:venus
```

Guidelines for managing VxVM disk groups in Sun Cluster 3.0

When administering VxVM disk groups in Sun Cluster 3.0, the following guidelines must be followed:

- Once a disk group has been registered with the cluster framework, **do not** use the VxVM commands or GUI to manually import or deport the disk group on the nodes of the cluster. The cluster will automatically import or deport the disk group appropriately when the device group is migrated from one node to another (due to cluster events, such as membership changes or issuing of an `scswitch` command).
- All disk groups (except rootdg) must be registered with the cluster framework.
- Any VxVM administrative commands (such as `vxedit`, `vxassist`, and`vxdg`) or GUI operations must be performed from the node which currently "owns" the disk group. Use the `scstat -D` command to determine ownership of the disk group.

```
scstat -D

-D - specifies printing of device group status only
```

- Whenever making any configuration changes to a disk group (especially changes which affect the volumes within a disk group), the disk group should be re-registered with the cluster. This can be done with `scconf` or `scsetup`

```
scconf -c -D name=<DeviceGroupName>

-c - specifies the "change" form of the scconf command
-D - specifies that device group options follow
name=<DeviceGroupName> - specifies the name of the VxVM
                         device group to change
```

Guidelines for managing VxVM disk groups in Sun Cluster 3.0

- Once a disk group is registered with the cluster, DO NOT use the VxVM command (`vxdg`) or GUI to import or deport the group. The cluster framework automatically performs an import or deport of the disk group in response to cluster events.
- All disk groups (except rootdg) must be registered with the cluster framework.
- Any VxVM administrative commands should only be performed on the node which currently owns the disk group. Disk group ownership can be determined using the `scstat` command:

```
# scstat -D
...
  Device Group Name:                               scdg2
  Status:                                          Online
  Primary:                                         mars
  Secondary:                                       venus
  Spare:
  Inactive:
  Transition:
...
```

- Whenever configuration changes are performed on a disk group being managed by the cluster, make sure to re-register the disk group with the cluster to ensure that the VxVM namespace is kept consistent throughout the cluster:

```
# vxassist -g scdg1 make vol-05 500m layout=mirror-stripe,log columns=8
# vxassist -g scdg1 remove volume vol-04
# scconf -c -D name=scdg1
```

CHAPTER 5

Configuring Resource Groups

Objectives

Purpose

In this chapter we will learn how to configure Sun Cluster 3.0 resource groups

Prerequisites

Understanding of Sun Cluster 3.0 architecture and concepts

Objectives

Upon completion of this chapter, the participant will be able to:

- Configure scalable data services as a Sun Cluster resource group
- Configure failover data services as a Sun Cluster resource group

Objectives

- Learn how to configure scalable data services as a Sun Cluster resource group
- Learn how to configure failover data services as a Sun Cluster resource group

Resource Group Configuration Overview

Resource groups are the main “container” for data service applications (and their resources) that are to be managed by the cluster.

The steps required to configure resource groups are:

1. Install the data service applications
2. Register the appropriate resource types
3. Create and configure the resource groups
4. Enable the resources
5. Bring the resource group online

Resource Group Configuration Overview

1. Install the data service applications
2. Register the appropriate resource types
3. Create and configure the resource groups
4. Enable the resources
5. Bring the resource group online

NOTE: The GUI wizard now makes these steps much simpler.

Step 1 - Install data service applications

The first step in configuring resource groups is to install the data service applications (i.e. Oracle, Netscape, Apache, etc.) that are to be managed by the cluster. Prior to performing the actual installation, the following information should be determined:

- **The location for the application's binaries**
    - On a cluster file system:
        - ❏ Only need to perform installation procedure once
        - ❏ Only a single copy of the application needs to be maintained
        - ❏ Upgrading of the application may require downtime for the resource group
    - On a local file system:
        - ❏ Installation must be performed identically on multiple nodes (identical paths, parameters, etc.)
        - ❏ Multiple copies of the application must be maintained
        - ❏ Upgrades of the application can be performed on an "offline" copy without disturbing the "online" copy, thus minimizing any resource group downtime
- **The location for the application's data**
    - Should always be on a globally accessible device or cluster file system
- **The scalable or failover hostname/address that will be used to "host" this application on the cluster**
    - Make sure to have the hostname(s)/address(es) entered in the appropriate name services and/or host files

Install the applications as outlined in the application's installation procedures. Make sure to use the appropriate scalable/failover hostname or address whenever hostname or addressing information is required during the installation (do not use a node's actual address!).

Step 1 - Install data service applications

- Prior to installing the applications, gather the following information:
  - Location for the application binaries
    - Cluster file system
    - Local file system (file system local to each node)
  - Location for the application data
    - Should always be a global device or cluster file system
  - Scalable or failover hostname and address that will be used to “host” the application
    - Make sure to enter the scalable or failover hostname and address in the appropriate name services and/or host files
- Install the applications to be used on the cluster as outlined in the application’s documentation
- If the application requires configuration of a hostname or IP address, use the desired logical or scalable hostname/address

Step 2 - Register resource types

Register the resource types for the data services that will be used on the cluster:

```
scrgadm -a -t <Resource Type to Register>

-a specifies "add" of a resource type, resource or resource group
-t specifies a resource type
<Resource Type to Register> Name of the resource to register
```

The resource types are:

Resource Type Name	Description
SUNW.iws	iPlanet Web Server
SUNW.oracle_listener	Oracle8 Listener
SUNW.oracle_server	Oracle8 Server
SUNW.apache	Apache Web Server
SUNW.nfs	NFS Server
SUNW.dns	DNS Server
SUNW.nsldap	Netscape Directory Server
SUNW.LogicalHostname	Logical Host address resource type for failover applications (Automatically registered during SC installation)
SUNW.SharedAddress	Shared Address resource type for scalable applications (Automatically registered during SC installation)

Resource types need to be registered only once and registration can be done from any node in the cluster.

Step 2 - Register resource types

- Use the `scrgadm(1M)` command to register the resource types that will be used on the cluster:

```
# scrgadm -a -t SUNW.iws
# scrgadm -a -t SUNW.oracle_listener
# scrgadm -a -t SUNW.oracle_server
# scrgadm -a -t SUNW.apache
# scrgadm -a -t SUNW.nfs
# scrgadm -a -t SUNW.dns
# scrgadm -a -t SUNW.nsldap
```

- Registration of a resource type can be performed from any node of the cluster (perform on only one node)

- NOTE: SUNW.LogicalHostname and SUNW.SharedAddress are pre-installed. If they are accidentally removed, they can be re-registered like other resource types:

```
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.SharedAddress
```

Step 3 - Create and configure resource groups

Now that we have resource types registered in the cluster, we can instantiate the resource types as resources in resource groups. The `scrgadm(1M)` is used to add resources to a resource group. The steps required to create and configure a resource group vary between a logical host (failover) group and a scalable resource group:

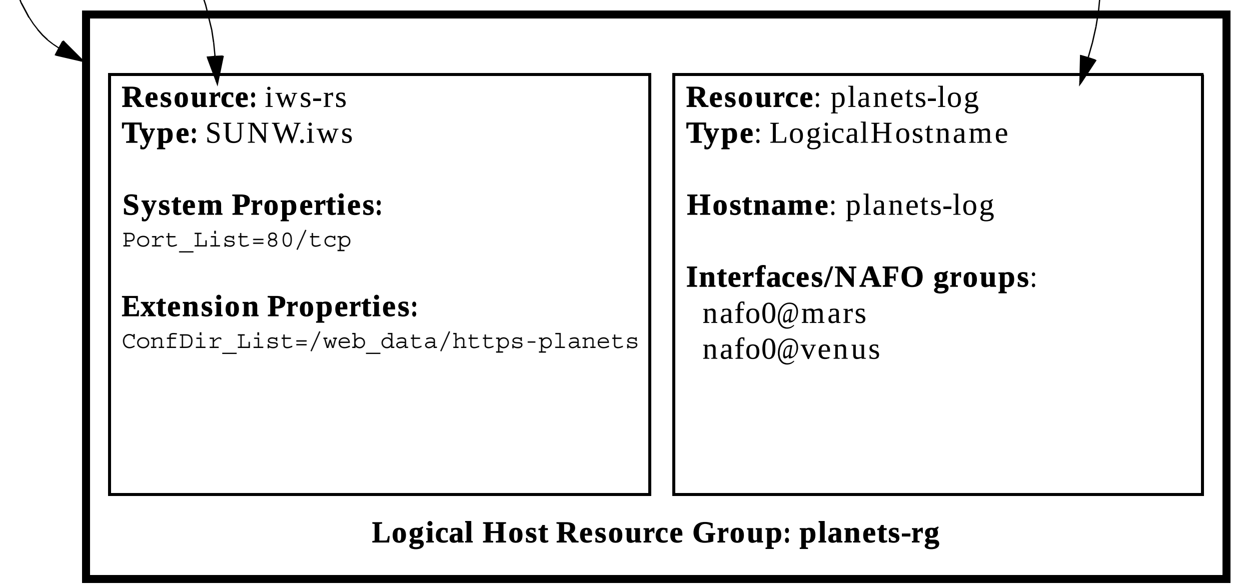
To create a Logical Host (failover) resource group:

```
Create the group:
scrgadm -a -g <group name> -h <nodelist>

-a Specifies “add” operation
-g <group name> Specifies the name of the logical host group
-h <nodelist> Comma separated list of nodes the group is
              eligible to run on. Defaults to all nodes

Add logical host addresses and adapters to the group:
scrgadm -a -L -g <group name> -l <lh-hostname> -n <iflist>

-a Specifies “add” operation
-L Specifies a Logical Hostname
-g <group name> Name of the resource group
-l <lh-hostname> Hostname of the for the logical host address
-n <iflist> Comma separated list of network interfaces or NAFO
            groups on each node on which the logical host will
            be configured.  If an interface is specified, a
            single adapter NAFO group will be created, if
            needed (Format: <Interface or NAFO group>@node).

Add the data service resources to the group:
scrgadm -a -j <resource name> -g <group name> -t <resource type>\
       -x <Resource extension Properties> ...                   \
        -y <Resource properties> ...

-a Specifies “add” operation
-j <resource name> Name of resource to add
-g <group name> Name of resource group to add resource to
-t <resource type> Type of resource being added
-x <Resource extension properties> Specifies the value for any
                                  extension properties for this
                                   resource
-y <Resource properties> Specifies the value for any resource
                         properties
Add shared addresses to the group:
scrgadm -a -S -j ....
```

Step 3 - Configure resource groups

- The scrgadm command is used to add resources to a resource group. The steps required vary depending if you are configuring a logical host (failover) resource group or a scalable resource group.
- To create a logical host (failover) resource group:

```
# scrgadm -a -g planets-rg -h mars,venus
# scrgadm -a -L -g planets-rg -l planets-log -n nafo0@mars,nafo0@venus
# scrgadm -a -j iws-rs -g planets-rg -t SUNW.iws \
        -x ConfDir_List=/web_data/https-planets -y Port_List=80/tcp
```



Resource: iws-rs
Type: SUNW.iws

System Properties:
Port_List=80/tcp

Extension Properties:
ConfDir_List=/web_data/https-planets

Resource: planets-log
Type: LogicalHostname

Hostname: planets-log

Interfaces/NAFO groups:
nafo0@mars
nafo0@venus

Logical Host Resource Group: planets-rg

Step 3 - Configure resource groups (Continued)

To configure a scalable service:

Create a failover group to contain the shared address:

```
scrgadm -a -g <group name> -h <nodelist>
-a Specifies "add" operation
-g <group name> Specifies the name of the resource group
-h <nodelist> Comma separated list of nodes the group is
              eligible to run on
```

Add address and adapters information to the group:

```
scrgadm -a -S -g <group name> -l <sa-hostname> -n <iflist>\
        -X <auxnodelist>
-a Specifies "add" operation
-S Specifies a Shared address
-g <group name> Name of the resource group
-l <sa-hostname> Hostname of the shared address
-n <iflist> Comma separated list of network interfaces or NAFO
            groups on each node on which the logical host will
            be configured.  If an interface is specified, a
            single adapter NAFO group will be created, if
            needed (Format: <Interface or NAFO group>@node).
-X <auxnodelist> List of nodes that can host the shared address
                 but not serve as the primary
```

Create a *separate* scalable resource group which is dependent on the shared address group:

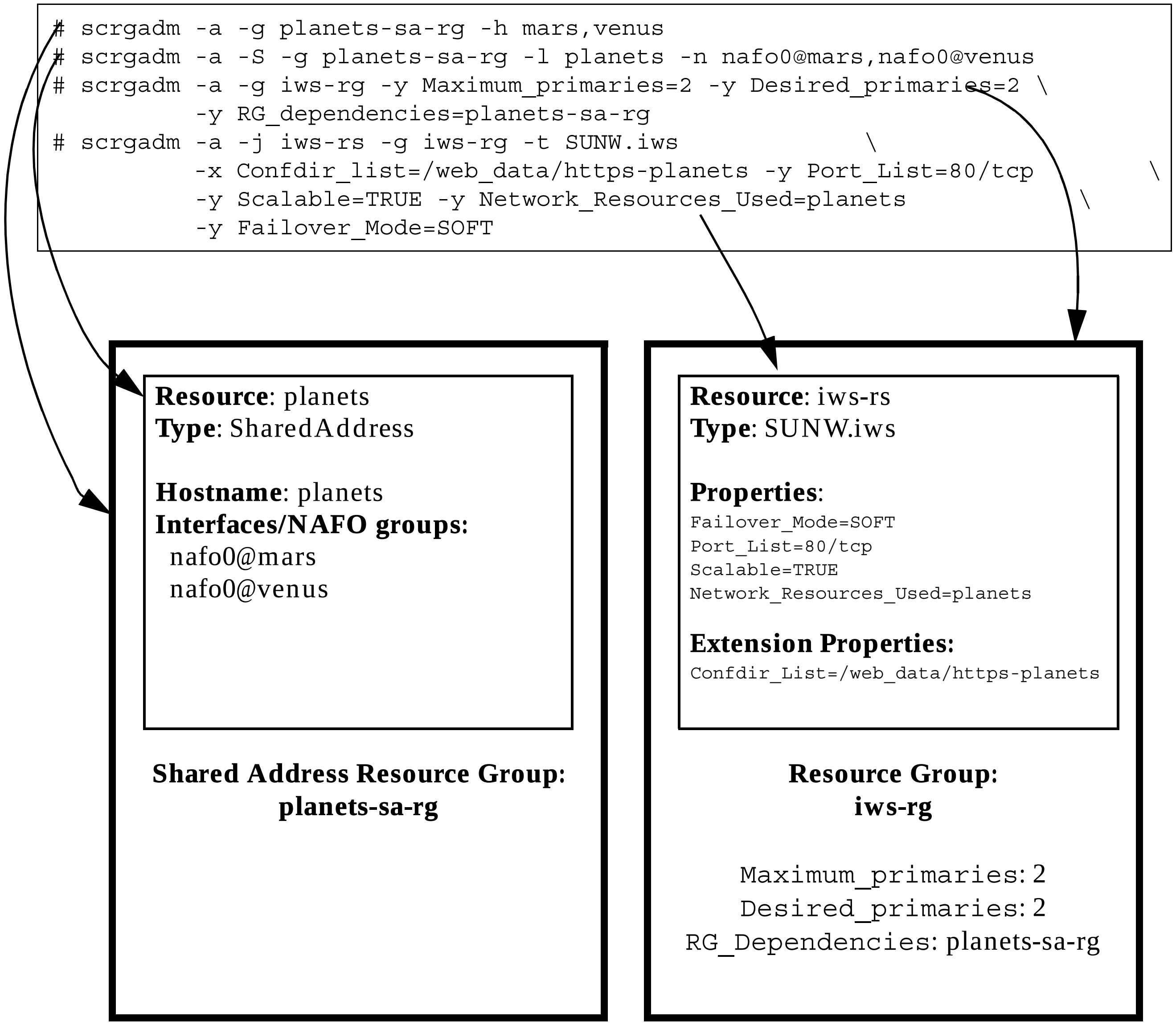
```
scrgadm -a -g <group name> -y <Properties> -y ...
-a Specfies "add" operation
-g Name of the resource group (different than shared
   address group)
-y <Properties> Sets properties for the resource group:
                Maximum_Primaries - max number of nodes for RG
                Desired_Primaries - desired number of nodes
                RG_dependencies  - set to shared address
                                   group created earlier
Setting Maximum_primaries >1 creates a scalable resource group.
Usually, Desired_primaries is set equal to Maximum_primaries.
```

Add scalable application resources to this dependent resource group:

```
scrgadm -a -j <resource name> -g <group name> -t <resource type>\
        -x <Resource extension Properties>                    \
        -y <Resource properties>

-a Specifies "add" operation
-j <resource name> Name of scalable resource to add
-g <group name> Name of resource group to add resource to
-t <resource type> Type of resource being added
```

Step 3 - Configure resource groups (Continued)

➤ To configure a scalable service:



```
# scrgadm -a -g planets-sa-rg -h mars,venus
# scrgadm -a -S -g planets-sa-rg -l planets -n nafo0@mars,nafo0@venus
# scrgadm -a -g iws-rg -y Maximum_primaries=2 -y Desired_primaries=2 \
          -y RG_dependencies=planets-sa-rg
# scrgadm -a -j iws-rs -g iws-rg -t SUNW.iws             \
          -x Confdir_list=/web_data/https-planets -y Port_List=80/tcp        \
          -y Scalable=TRUE -y Network_Resources_Used=planets            \
          -y Failover_Mode=SOFT
```

Resource: planets
Type: SharedAddress

Hostname: planets
Interfaces/NAFO groups:
nafo0@mars
nafo0@venus

Shared Address Resource Group:
planets-sa-rg

Resource: iws-rs
Type: SUNW.iws

Properties:
`Failover_Mode=SOFT`
`Port_List=80/tcp`
`Scalable=TRUE`
`Network_Resources_Used=planets`

Extension Properties:
`Confdir_List=/web_data/https-planets`

Resource Group:
iws-rg

`Maximum_primaries`: 2
`Desired_primaries`: 2
`RG_Dependencies`: planets-sa-rg

Step 4 - Enable the resources

Finally, all resources configured in the resource groups must be enabled, together with their fault monitors. This is done using the `scswitch(1M)` command:

```
Enable the resources:
scswitch -e -j <ResourceName>

-e specifies an “enable” operation
-j <ResourceName> Resource to be enabled

Enable the resources monitors:
scswitch -e -M -j <ResourceName>

-e Specifies an “enable” operation
-M Specifies resource monitor
-j <ResourceName> Resource to have the monitors enabled for
```

Step 4 - Enable the resources

➤ Enable the resources and resource monitors:

```
# scswitch -e -j planets-log
# scswitch -e -M -j planets-log
# scswitch -e -j iws-rs
# scswitch -e -M -j iws-rs

# scswitch -e -j planets
# scswitch -e -M -j planets
# scswitch -e -j iws-rs
# scswitch -e -M -j iws-rs
```

Step 5 - Bring the resource group online

Bring the resource group under RGM control ("managed") and place it online on a node (or nodes):

```
Make the resource group managed:
scswitch -o -g <Resource Group Name>

-o Specifies that the resource group is to be placed in a
   managed state
-g <Resource Group Name> Resource group to bring into a
                         a managed state

Bring the resource group online on a node:
scswitch -z -g <Resource Group Name> -h <NodeName>

-z Specifies that a mastery change operation
-g <Resource Group Name> Specifies the group to change the
                         mastery of
-h <NodeName> Name of the node to master the given resource group
```

Note: Steps 4 and 5 tell you to execute many `scswitch` commands. Here is a shortcut: Instead of invoking many `scswitch` commands specifying the `-e`, `-o`, and `-z` options separately, you can use a single `scswitch` command with the single `-Z` option.

For each resource group specified by the `-g` option, `scswitch -Z` enables all resources and their monitors, moves the resource group into managed state, and brings the resource group online on all the default primaries. Without the `-g` option, `scswitch` attempts to bring all resource groups online.

Step 5 - Bring the resource group online

➤ Make the resource group managed and bring it online:

```
# scswitch -o -g planets-rg
# scswitch -z -g planets-rg -h venus

# scswitch -o -g planets-sa-rg
# scswitch -o -g iws-rg
# scswitch -z -g planets-sa-rg -h mars
# scswitch -z -g iws-rg -h mars,venus
```

➤ Shortcut for Step 4 and Step 5: `scswitch -Z`

```
# scswitch -Z -g planets-rg

# scswitch -Z -g planets-sa-rg
# scswitch -Z -g iws-rg
```

CHAPTER 6

Configuring the Data Services

Objectives

Purpose

In this chapter we will cover the data services supported by Sun Cluster 3.0. The properties, fault monitoring and installation guidelines for each supported data service will be covered.

Prerequisites

Understanding of Sun Cluster architecture and concepts

Understanding of Sun Cluster installation and configuration

Understanding of Sun Cluster RGM configuration

Objectives

Upon completion of this chapter, the participant will be able to:

- Describe resource, extension and resource group properties
- Describe the resource and extension properties for each supported data service
- Install and configure each of the supported data services on a cluster

Objectives

- ➤ Describe resource, extension and resource group properties
- ➤ Describe the properties of each supported data service
- ➤ Illustrate how to install and configure the supported data services with Sun Cluster 3.0

Sun Cluster 3.0 Supported Data Services

Sun Cluster 3.0 supports a number of data service applications. The table on the following page lists the supported data service applications.

Sun Cluster 3.0 Supported Data Services

Application	Description	Versions	Cluster Support
iPlanet Web Server	HTTP Server	4.1	Scalable Failover
Apache	HTTP Server	1.3.9	Scalable Failover
NFS	Network File Server	2,3	Failover
DNS	Domain Name Server	BIND 8	Failover
iPlanet Directory Server	LDAP Server	4.11	Failover
Oracle8i	Database Server	8.1.6	Failover

Resource Properties

When a resource is placed into a resource group, a number of *properties* may be set which affect how the resource is managed by the RGM. Resources have two types of properties that may be set: Resource properties and extension properties.

Resource Properties

Resource properties are a set of system-defined properties which all Sun Cluster 3.0 resources have, regardless of their resource type. Depending on the implementation of the resource type, not all resource properties for a particular resource may be utilized. The table on the following page lists the settable resource properties.

Resource properties can be set or changed using the -y argument of the `scrgadm` command.

Resource Properties

- ➤ Resource properties are a set of system-defined properties which all resources have. The configurable resource properties are:

Resource Property	Description	Default
Resource_dependencies	A comma-separated list of other resources (in the same group) on which this resource is dependent	Empty
Failover_mode	Controls the response of the RGM to resource start or stop failure or timeout. Can be set to NONE, SOFT or HARD: NONE - Sets the resource state to a failure state and performs no other action SOFT - For START failure, the RGM relocates the resource's group to another node. For STOP failure, the RGM leaves the resource in STOP_ FAILED state and doesn't relocate the group. HARD - If the resource startup fails, the resource's group will be relocated to another node, if the resource stop fails, the node will be aborted	Resource Type Dependent
Network_resources_used	List of network address resources (logical host and shared address) on which this resource is dependent.	Empty
Cheap_probe_interval	The number of seconds between invocations of a quick fault probe for the resource	Resource Type Dependent
Thorough_probe_interval	The number of seconds between invocations of a high-overhead fault probe for the resource	Resource Type Dependent
Retry_count	The number of times a fault monitor should attempt to restart a failed resource before giving up	Resource Type Dependent
Retry_interval	The number of seconds over which to count attempts (Retry_count) to restart a failed resource	Resource Type Dependent
<*METHOD*>_TIMEOUT where <*METHOD*> is: START, STOP, PRENET_START, POSTNET_STOP, INIT, FINI, BOOT, VALIDATE, UPDATE, MONITOR_START, MONITOR_STOP, MONITOR_CHECK	The time, in seconds, that the RGM will allow a particular method to complete before considering the method invocation a failure	3600
Scalable	Optional. Whether this resource is a scalable resource or not. Values: TRUE/FALSE	Resource Type Dependent

Resource Properties (Continued)

The following page continues the listing of resource system properties

Resource Properties (Continued)

Resource Property	Description	Default

`Load_balancing_weights`	Optional configuration string array that is used by the chosen load balancing policy (see the Load_balancing_policy property)
	For the weighted policy, format is
	`weight@nodeid,weight@nodeid...`
	or
	`weight@nodename,weight@nodename...`
	where `nodeid` or `nodename` is a node in the cluster and `weight` is an integer reflecting a percentage of load (percentage is the weight divided by the sum of all weights of active nodes).
	Empty string indicates that a uniform distribution should be used.
		Empty

`Load_balancing_policy`	The type of load balancing that this scalable resource will be using. Valid policies are LB_WEIGHTED, LB_STICKY, and LB_STICKY_WILD.	LB_WEIGHTED
`Port_list`	List of ports / protocols configured for the resource. Valid protocols for scalable resources are “tcp” and “udp”; non-scalable resources can use other protocols as well.	Resource type Dependent

Resource Properties (Continued)

Extension Properties

Extension properties are a set of properties which are defined by the resource type implementor. Extension properties provide a way to add support for application-specific properties which need to be managed by the RGM. Application paths, ports, application logins or configuration file names are examples of information which can be managed using extension properties.

Extension properties can be set or changed using the `-x` argument of the `scrgadm` command.

Extension Properties

- Defined by the resource type implementor
- Provides a way to add application-specific properties to the set of system-defined properties which are managed by the RGM
- Examples of extension properties:
  - File system paths for required application files
  - Configuration file names
  - Application login ids to be used by the fault monitor
- Extension properties are set or changed using the `-x` argument to the `scrgadm` command

Resource Group Properties

Resource Groups also have a set of configurable properties which can affect how the RGM manages the resource group.

The properties are set or changed using the -y argument to the `scrgadm` command.

Resource Group Properties

➤ Like resources, resource groups have properties which can be configured using the -y option of the `scrgadm` command. Configurable properties are:

RG Property	Description	Default
Maximum_primaries	The maximum number of nodes on which the group can be online at once	1
Desired_primaries	The number of nodes where the group is desired to be online at once	1
Failback	TRUE or FALSE value indicating whether the set of primaries for this RG will be recomputed when a node joins the cluster. A RG may be migrated to a more preferred node as a result.	FALSE
RG_dependencies	Optional list of resource groups on which this resource group is dependent	Empty
Global_resources_ used	Optional list (comma separated) of global file system pathnames used by this RG	All Resources
RG_mode	FAILOVER or SCALABLE	FAILOVER: If Maximum_primaries = 1 SCALABLE: If Maximum_primaries > 1
Implicit_network_dependencies	The RGM will enforce implicit strong dependencies of non-network-address resources on network-address resources within the group. "Network address resources" are the LogicalHostname and SharedAddress resource types. All other resource types are non-network-address.	TRUE
Nodelist	List of nodes which are eligible to host this resource group	All Nodes
PathPrefix (optional)	Specifies a directory where resources in a RG can write essential files	None
Pingpong_Interval	Used to determine when to bring a RG online and to relocate a resource and RG	3600 seconds

iPlanet Web Server

Sun Cluster 3.0 can support the installation of a iPlanet Web Server as either a scalable or failover data service.

Application Installation

The high level steps to installing iPlanet Web server are:

1. Determine where you want to install iPlanet Web Server. It requires at least 115MB (not counting any data files, such HTML docs, CGI scripts, etc.) of free space. Also, enter the logical or scalable address and hostname into the appropriate name services as well as the `/etc/hosts` files on the cluster ndoes. Finally, create a user and group entry to serve as the owner of the NS server (the default is `nobody/nobody`). Make sure to create this user and group on each node of the cluster.

2. From one node of the cluster, locate the CD-ROM image of iPlanet Web Server and run `setup`.

   a. When prompted for the Install Location, supply the appropriate path.

   b. When prompted for the Computer Name, provide the fully qualified name of the *logical* or *scalable* host you want to use for the iPlanet Web Server, **not** the *physical* hostname of the node.

   c. When prompted for the user and group to run the server as, enter the appropriate user and group configured in step 1.

   d. When prompted for an administration port, you may use the default, however, ***make sure to note down what the assigned port number is*** (you will need this number to complete the configuration of the server)!

   The answers to the rest of the prompts will vary, depending on the specific details of the installation environment.

iPlanet Web Server

➤ Application Installation

1. Determine and configure the installation path and the logical/scalable hostname/address to be used for the web server

2. Use `setup` to install the iPlanet binaries:

```
# ./setup
                             Sun Netscape Alliance
                 iPlanet Web Server Installation/Uninstallation
--------------------------------------------------------------------------------


Welcome to the iPlanet Web Server installation program

...
  <Various installation messages>
  <You will be asked to configure the type of installaion and options to be
   installed>
...
Install Location [/usr/netscape/server4]: /global/iws

Computer name [venus]: planets.sun.com

System User [nobody]: nobody
System Group [nobody]: nobody
...
Run iWS Administration Server as [root]:
...
iWS Admin Server User Name [admin]: admin
iWS Admin Server Password:
iWS Admin Server Password (again):
...
iWS Admin Server Port [8888]:
...
Web Server Port [80]:
...
Do you want to register this with an existing Directory Server [No]:
...
Web Server Content Root [/global/iws/docs]: /global/iws/docs
...
   < The setup program will copy the appropriate files to the install location >
...
# cd /global/iws
# ./startconsole
```

iPlanet Web Server (Continued)

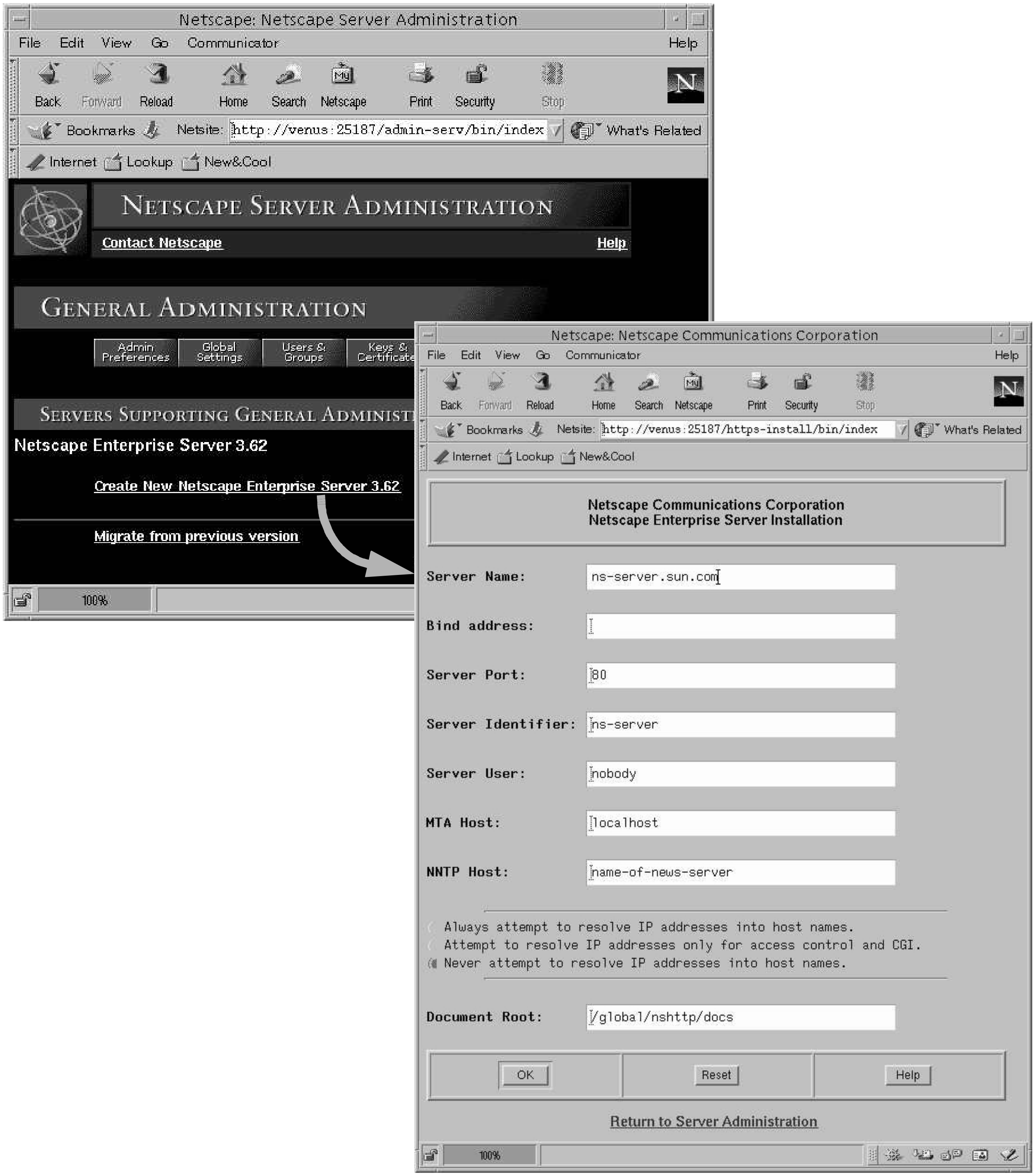
3. Use a browser to connect to the administration server. Use the *physical hostname* of the node on which you are performing the installation on (since the logical/scalable address will not exist until a resource group is configured and placed on line), together with the port number assigned (or chosen) during the `setup` session (e.g. `http://venus.sun.com:25431`, where venus is the physical node name and 25431 is the port number assigned to the admin server during `setup`). The browser is used to set up iPlanet Web Server instances:
   a. After typing the login and password (configured during setup), click on the “Create New iPlanet Web Server ...” link
   b. Fill out the presented form accordingly. Make sure that the document root directory resides on a global file system

      NOTE: the screen shots on the next few pages are from the Netscape Enterprise Server 3.6.1 Console, not the iPlanet 4.1 product.

(Continued ...)

iPlanet Web Server (Continued)

3. Connect to the admin server to create the web server



Netscape: Netscape Server Administration
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://venus:25187/admin-serv/bin/index What's Related
Internet Lookup New&Cool
NETSCAPE SERVER ADMINISTRATION
Contact Netscape
Help
GENERAL ADMINISTRATION
Admin Preferences
Global Settings
Users & Groups
Keys & Certificate
SERVERS SUPPORTING GENERAL ADMINIST
Netscape Enterprise Server 3.62
Create New Netscape Enterprise Server 3.62
Migrate from previous version
100%
Netscape: Netscape Communications Corporation
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://venus:25187/https-install/bin/index What's Related
Internet Lookup New&Cool
OK
Reset
Help
Return to Server Administration
100%

Netscape Communications Corporation
Netscape Enterprise Server Installation

Server Name:	ns-server.sun.com
Bind address:	
Server Port:	80
Server Identifier:	ns-server
Server User:	nobody
MTA Host:	localhost
NNTP Host:	name-of-news-server

○ Always attempt to resolve IP addresses into host names.
○ Attempt to resolve IP addresses only for access control and CGI.
● Never attempt to resolve IP addresses into host names.

Document Root:	/global/nshttp/docs

iPlanet Web Server (Continued)

4. Before using the server in a cluster environment, there is some manual configuration which must be done. Each node of the cluster requires that a local copy of the logs directory be maintained. To set up separate, per node logging:

   a. Decide on a location for the local iPlanet Web Server logs. This path should be on a local (not global) file system

   b. If required, create the path on **each** node of the cluster. The owner and group for the directory should be set to the owner and group assigned to the iPlanet Web Server during `setup`.

   c. On the node where the installation was performed, edit the `magnus.conf` file, which is located in *`<Server Root>`*`/https-`*`<Server Name>`*`/config` (`<Server Root`> and `<Server Name>` were configured during the `setup` session):

      Change the entries for `ErrorLog` and `PidLog` to place the `errors` file and the `pid` file into the directory created in step 4b.

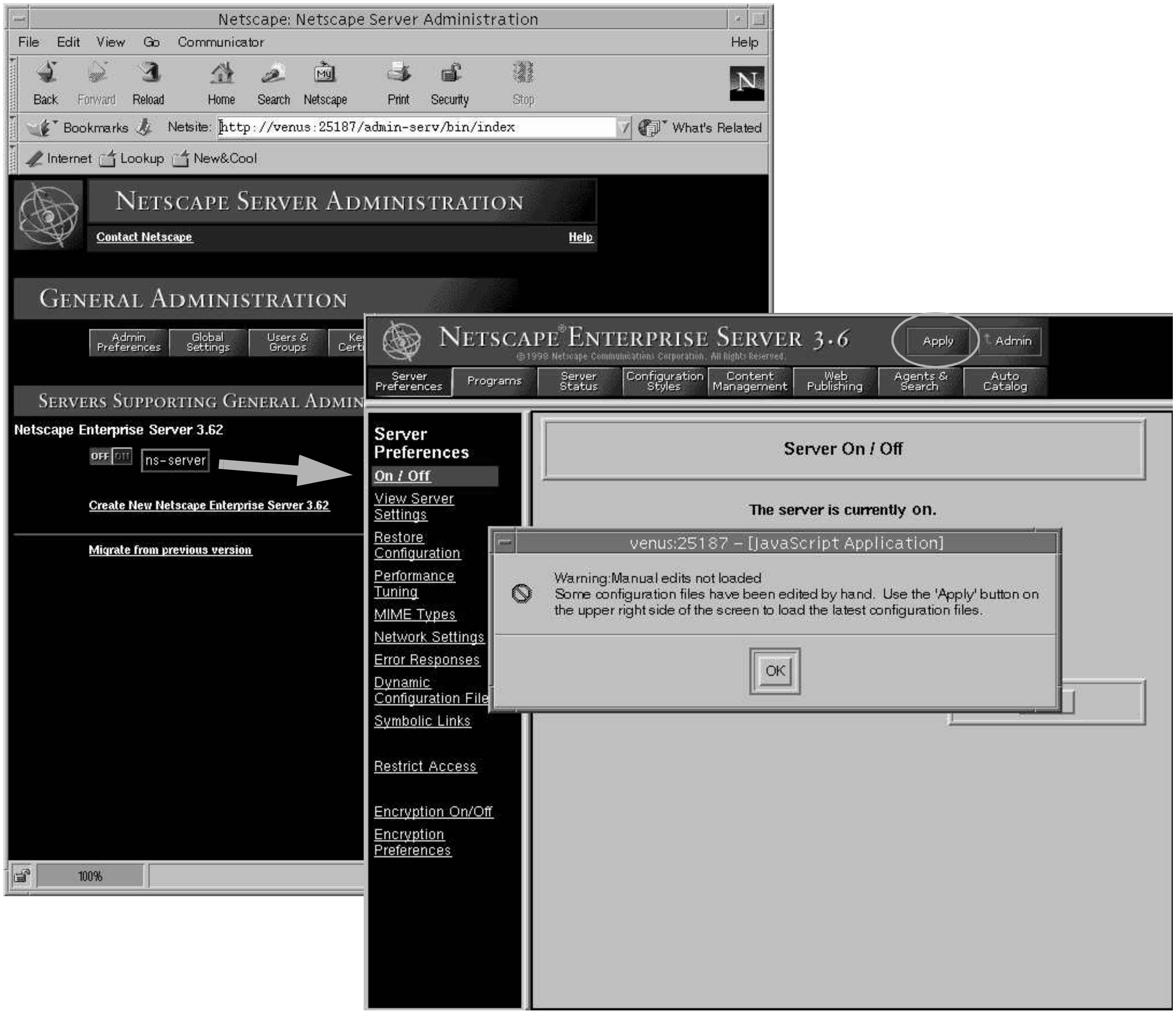
   d. From the administration server home page, click on the button next to the Off-On switch which has the name of the server on it (in the “Servers Supporting General Administration:” section of the page). The server will warn you that manual edits have been made, click on the Apply button in the upper right hand corner of the web page and choose to read the configuration changes from disk.

(Continued ...)

iPlanet Web Server (Continued)

4. Configure per-node server logging

```
# mkdir -p /var/cluster/nshttp/https-ns-server/logs
# chown nobody:nobody /var/cluster/nshttp/https-ns-server/logs
# vi /global/nshttp/https-ns-server/config/magnus.conf
...<Change ErrorLog and PidLog entries>...
# cat /global/nshttp/https-ns-server/config/magnus.conf
...
ErrorLog /var/cluster/nshttp/https-ns-server/logs
PidLog /var/cluster/nshttp/https-ns-server/pid
...
```


Netscape: Netscape Server Administration
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://venus:25187/admin-serv/bin/index What's Related
Internet Lookup New&Cool
Netscape Server Administration
Contact Netscape Help
General Administration
Admin Preferences Global Settings Users & Groups Ke Certi
Servers Supporting General Admin
Netscape Enterprise Server 3.62
OFF ns-server
Create New Netscape Enterprise Server 3.62
Migrate from previous version
100%
Netscape Enterprise Server 3.6
Apply Admin
Server Preferences Programs Server Status Configuration Styles Content Management Web Publishing Agents & Search Auto Catalog
Server Preferences
On / Off
View Server Settings
Restore Configuration
Performance Tuning
MIME Types
Network Settings
Error Responses
Dynamic Configuration File
Symbolic Links
Restrict Access
Encryption On/Off
Encryption Preferences
Server On / Off
The server is currently on.
venus:25187 – [JavaScript Application]
OK

Warning:Manual edits not loaded
Some configuration files have been edited by hand. Use the 'Apply' button on the upper right side of the screen to load the latest configuration files.

iPlanet Web Server (Continued)

4. Configure per-node logging (continued):

   e. Click on the “View Server Settings” link (on the right hand side of the page). Near the bottom of the page, click on the “Access log” link to set the Log File location to a file that’s in the path created in step 4b. Save and apply the changes.

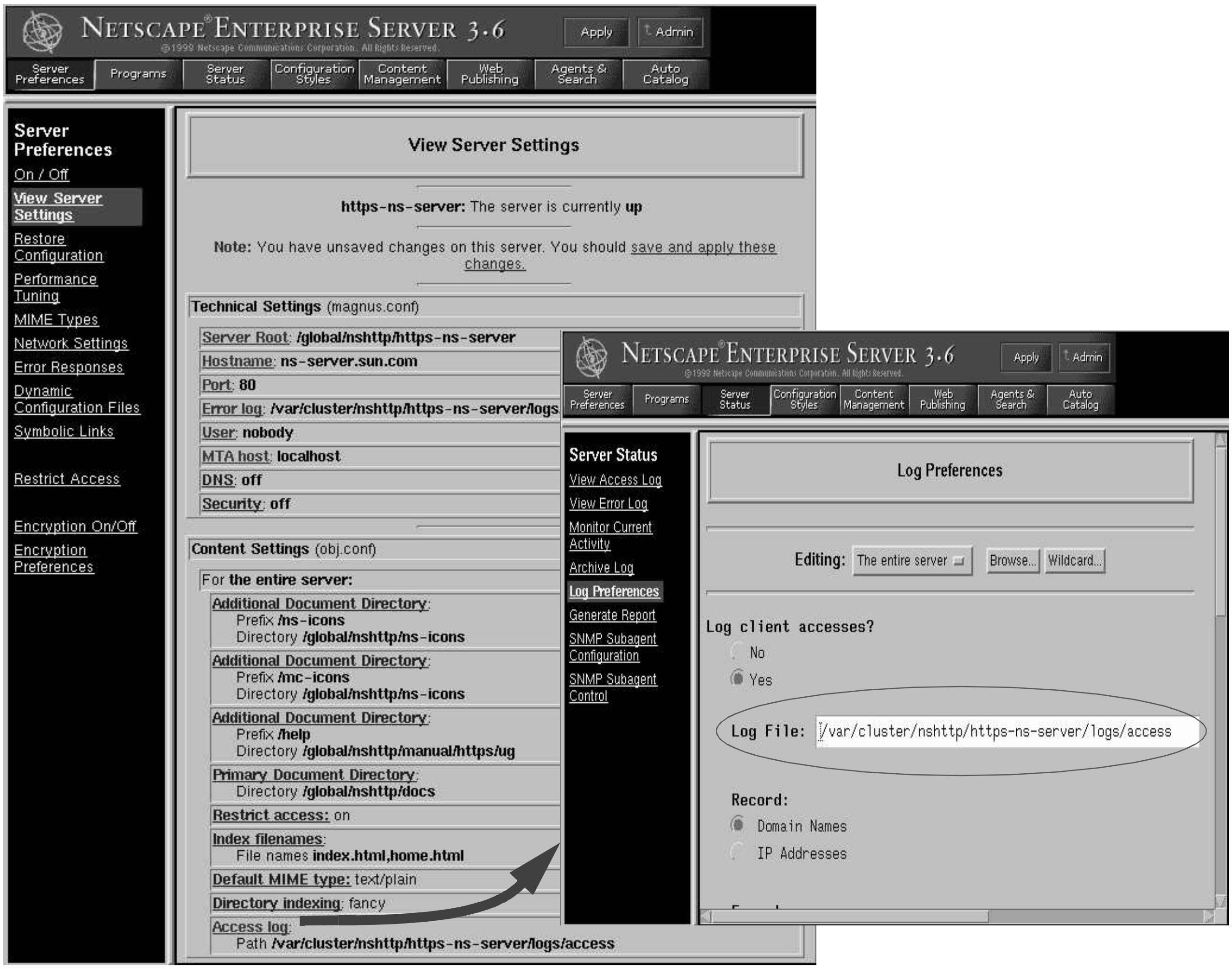
5. Perform some testing of the iPlanet Web Server.

   a. Make sure that the server can be started up and shut down properly (using the administrative interface and/or using the `start` and `stop` scripts located in *`<Server Root>`*`/https-`*`<Server Name>`*.

   b. Make sure you can connect to the server using a client browser. For the time being, you will have to use the node’s physical hostname or address to connect to the server (since the logical/scalable address has not been configured yet).

iPlanet Web Server (Continued)

4. Configure per-node server logging (continued)



View Server Settings

https-ns-server: The server is currently **up**

Note: You have unsaved changes on this server. You should save and apply these changes.

Technical Settings (magnus.conf)

- Server Root: **/global/nshttp/https-ns-server**
- Hostname: **ns-server.sun.com**
- Port: **80**
- Error log: **/var/cluster/nshttp/https-ns-server/logs**
- User: **nobody**
- MTA host: **localhost**
- DNS: **off**
- Security: **off**

Content Settings (obj.conf)

For **the entire server:**

- Additional Document Directory: Prefix **/ns-icons** Directory **/global/nshttp/ns-icons**
- Additional Document Directory: Prefix **/mc-icons** Directory **/global/nshttp/ns-icons**
- Additional Document Directory: Prefix **/help** Directory **/global/nshttp/manual/https/ug**
- Primary Document Directory: Directory **/global/nshttp/docs**
- Restrict access: on
- Index filenames: File names **index.html,home.html**
- Default MIME type: text/plain
- Directory indexing: fancy
- Access log: Path **/var/cluster/nshttp/https-ns-server/logs/access**

Log Preferences

Editing: The entire server

Log client accesses?
○ No
● Yes

Log File: /var/cluster/nshttp/https-ns-server/logs/access

Record:
● Domain Names
○ IP Addresses

5. Test the server
   - ➤ Start the server
   - ➤ Connect to the server using a browser
   - ➤ Shut down the server

iPlanet Web Server (Continued)

RGM Configuration

Extension Properties

The extension properties defined for the nshttp resource type are listed on the following page.

iPlanet Web Server

Extension Property	Description	Default
`Confdir_list`	A list containing the installation paths of one or more instances of the iPlanet Web Server (i.e. where the start and stop scripts are located for this server -- `<Server Root>/https-<Server Name>).`	
`Monitor_retry_count`	Number of times to set attempt to restart the fault monitor	4
`Monitor_retry_interval`	The time window (in minutes) to measure fault monitor restarts (See the Monitor_retry_count extension property)	2
`Probe_timeout`	Time out value (in seconds) used for the fault monitor probe	30

iPlanet Web Server

Resource Properties

The following page lists the default values of the resource properties for the `nshttp` resource type.

iPlanet Web Server (Continued)

- Resource property default values for the `nshttp` resource type:

Resource Property	Default (nshttp resource type)
<*METHOD*>_timeout Where <METHOD> is Start, Stop, Validate, Update, Init, Fini, Boot, Monitor_Start, Monitor_Stop, Monitor_Check	300 (Seconds)
Thorough_Probe_Interval	60 (Seconds)
Retry_Count	2
Retry_Interval	300 (Seconds)
Failover_Mode	SOFT
Network_resources_used	Empty
Scalable	FALSE
Load_balancing_policy	LB_WEIGHTED
Load_balancing_weights	Empty
Port_list	80/tcp

iPlanet Web Server (Continued)

Resource Group Properties

When configuring the web server as a scalable data service, the following resource group properties should be set properly:

`Maximum_primaries` - Set to the maximum number of nodes you may wish to configure the scalable resource group to run on simultaneously using administrative commands. The RGM will never attempt to bring the resource group online on more than this number of nodes at any one time.

`Desired_primaries` - Set to the number of nodes on which you desire to have the resource group running simultaneously under normal operating conditions. The RGM attempts to maintain this number of active primaries for the resource group. If the number of nodes on which the resource group is online falls below this setting due to hardware or software failures, the RGM will attempt to start the resource group on additional nodes until this setting is satisfied.

`RG_dependencies` - In the group containing the web server resource (i.e. the resource based on the `nshttp` resource type), set this property to the name of the resource group containing the associated SharedAddress resource.

iPlanet Web Server (Continued)

- Make sure to set the `Maximum_primaries`, `Desired_Primaries` and `RG_Dependencies` resource group properties properly (especially if configuring the web server as a scalable data service)
- Scalable data service example:

Creates 2 resource groups - `ns-server-sa-rg` and `iws-rg` using the scalable address `ns-server`. Note: if you omit the `-n` option in the third `scrgadm` command, below, the appropriate NAFO groups will be discovered automatically.

```
# scrgadm -a -t SUNW.nshttp
# scrgadm -a -g web-server-sa-rg -h mars,venus
# scrgadm -a -S -g web-server-sa-rg -l ns-server \
          -n nafo0@venus,nafo0@mars
# scrgadm -a -g iws-rg      \
          -y Maximum_primaries=2 \
          -y Desired_Primaries=2 \
          -y RG_dependencies=ns-server-sa
# scrgadm -a -j iws-rs -g iws-rg -t SUNW.iws \
          -x Confdir_list=/global/nshttp/https-ns-server   \
          -y Scalable=TRUE                                 \
          -y Network_Resources_Used=ns-server

# scswitch -e -j ns-server
# scswitch -e -j iws-rs
# scswitch -o -g ns-server-sa-rg
# scswitch -o -g iws-rg
# scswitch -z -g web-server-sa -h venus
# scswitch -z -g iws-rg -h venus,mars
```

A shortcut to substitute for the 6 `scswitch` commands listed in the box above:

```
# scswitch -Z -g web-server-sa-rg
# scswitch -Z -g iws-rg
```

iPlanet Web Server (Continued)

The following page illustrates an example of configuring an iPlanet Web Server as a failover data service.

iPlanet Web Server

➤ Failover data service example:

Creates a resource group - ns-server-lh using the logical host address/name of ns-server.
Note: if you omit the -n option in the third scrgadm command, below, the appropriate NAFO groups will be discovered automatically.

```
# scrgadm -a -t SUNW.nshttp
# scrgadm -a -g web-server-rg -h venus,mars
# scrgadm -a -L -g web-server-rg -l ns-server \
          -n nafo0@venus,nafo0@mars
# scrgadm -a -j iws-rs -g web-server-rg -t SUNW.iws \
          -x ConfDir_list=/global/nshttp/https-ns-server

# scswitch -e -j ns-server
# scswitch -e -j iws-rs
# scswitch -o -g web-server-rg
# scswitch -z -g web-server-rg -h venus
```

A shortcut to substitute for the 4 scswitch commands listed in the box above:

```
# scswitch -Z -g web-server-rg
```

Apache Web Server

Sun Cluster 3.0 supports the installation of the Apache Web Server as either a scalable or failover application.

Application Installation

Since Apache is open-source software, installation procedures may vary, depending on the nature of the specific distribution of Apache you are installing. The source for Apache may be downloaded from www.apache.org, precompiled versions of Apache can be obtained from www.sunfreeware.com. These installation instructions assume a local installation of Apache on each node of the cluster.

1. Determine where you want to install the Apache Web Server. It requires 3-4 MB (not counting any data files, such HTML docs, CGI scripts) of free space. The default location for the installation is `/usr/local/apache`. Also, enter the logical or scalable address and hostname into the appropriate name services as well as the `/etc/hosts` files on the cluster nodes.

2. Install the Apache directory tree to you target location (if you have compiled the source, you will need to copy your source tree to your target directory)

3. Configure the server by editing the `httpd.conf` file. By default this file is located in the `conf` directory of your installation directory (for example, `/usr/local/apache/conf/httpd.conf`). If performing a local installation, make sure that all nodes have an identical httpd.conf file. At the minimum, you will need to configure the following variables:

    a. **ServerName** - set to the logical or scalable host name you want the web server to be associated with

    b. **DocumentRoot** - set to a location (on a global file system) where you are going to store the html documents

    c. **Port** - set to the TCP port number you want to use for the web server

Apache Web Server

➤ Application Installation

1. Determine the installation location and logical/scalable address/hostname which will be used for the web server

2. Install the Apache web server files to the desired installation location:

```
# mkdir -p /usr/local/apache
# cd /usr/local/apache
# tar xvf /net/export/share/apache_1.3.9.tar
x bin, 0 bytes, 0 tape blocks
x bin/httpd, 803164 bytes, 1569 tape blocks
...
<Files extracted by tar>
...
```

3. Configure the web server by editing the httpd.conf file:

```
# vi /usr/local/apache/httpd.conf
...
<Edit http.conf file, uncomment ServerName and set to match the Logical/Scalable
  hostname, DocumentRoot should be changed to a global file system>
<MAKE SURE TO DO ON ALL NODES>
...
# cat /usr/local/apache/httpd.conf
...
Port 80
...
ServerName ap-server
...
DocumentRoot /global/ap-data/htdocs
...
<Directory /global/ap-data/htdocs
...
```

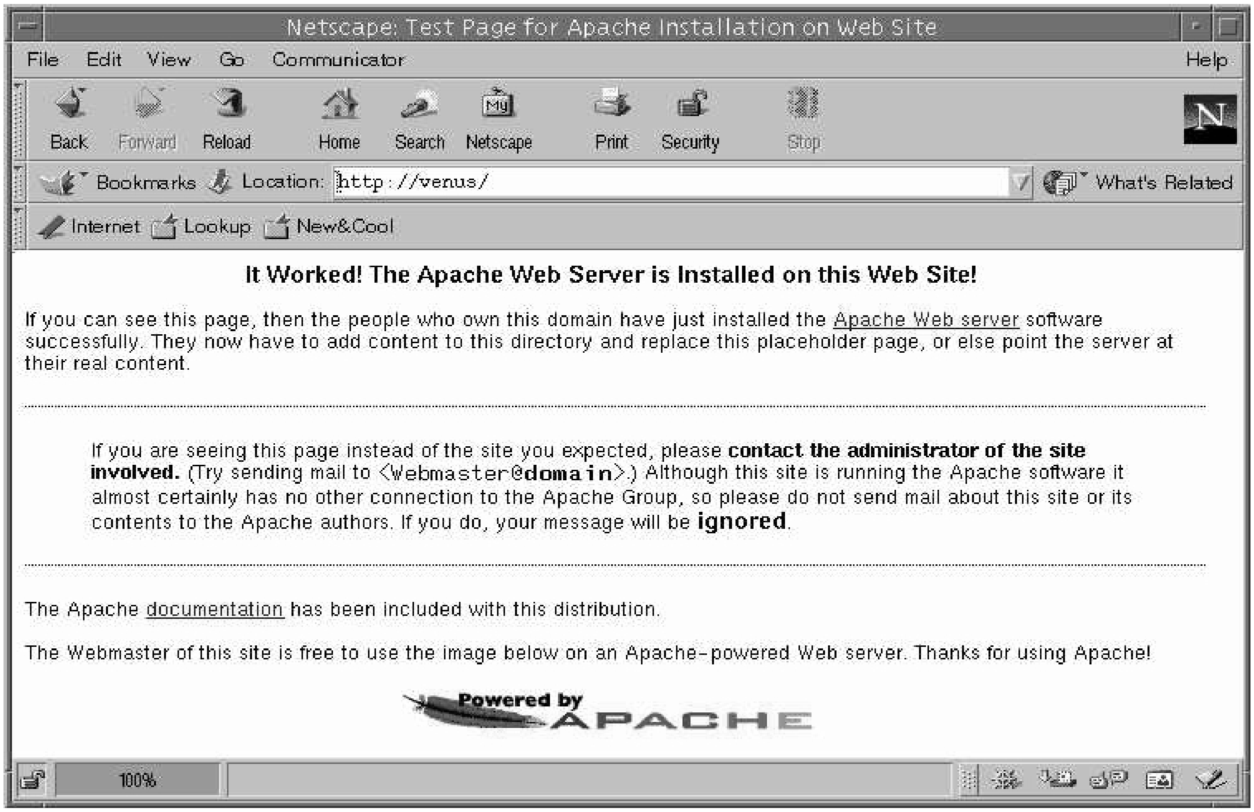
Apache Web Server (Continued)

4. Perform testing of the server prior to RGM configuration for the data service
   a. Start the server
   b. Connect to the server via a client web browser
   c. Shut down the server

Apache Web Server

4. Test the installation

```
# cd /usr/local/apache/bin
# ./apachectl start
# ps -ef | grep httpd
  nobody  3218  3216  0 16:22:19 ?        0:00 ./httpd
    root  3222   369  0 16:22:24 console  0:00 grep httpd
  nobody  3217  3216  0 16:22:19 ?        0:00 ./httpd
    root  3216     1  0 16:22:19 ?        0:00 ./httpd
  nobody  3221  3216  0 16:22:19 ?        0:00 ./httpd
  nobody  3220  3216  0 16:22:19 ?        0:00 ./httpd
  nobody  3219  3216  0 16:22:19 ?        0:00 ./httpd
# ./apachectl stop
# ps -ef | grep httpd
    root  3689   369  0 16:32:41 console  0:00 grep httpd
```



Location: http://venus/

It Worked! The Apache Web Server is Installed on this Web Site!

If you can see this page, then the people who own this domain have just installed the Apache Web server software successfully. They now have to add content to this directory and replace this placeholder page, or else point the server at their real content.

If you are seeing this page instead of the site you expected, please **contact the administrator of the site involved.** (Try sending mail to <Webmaster@**domain**>.) Although this site is running the Apache software it almost certainly has no other connection to the Apache Group, so please do not send mail about this site or its contents to the Apache authors. If you do, your message will be **ignored**.

The Apache documentation has been included with this distribution.

The Webmaster of this site is free to use the image below on an Apache-powered Web server. Thanks for using Apache!

Powered by APACHE

Apache Web Server (Continued)

RGM Configuration

Extension Properties

The extension properties defined for the apache resource type are listed on the following page.

Apache Web Server

Extension Property	Description	Default
`Confdir_list`	A list containing the installation paths of the one or more instances of Apache Web server (i.e. where the httpd.conf file is located)	
`Bin_dir`	Location of the Apache Binary Directory	
`Monitor_retry_count`	Number of times to set attempt to restart the fault monitor	4
`Monitor_retry_interval`	The time window (in minutes) to measure fault monitor restarts (See the Monitor_retry_count extension property)	2
`Probe_timeout`	Time out value (in seconds) used for the fault monitor probe	30

Apache Web Server

Resource System Properties

The following page lists the default values of the resource system properties for the `apache` resource type.

Apache Web Server (Continued)

- Resource property default values for the `apache` resource type:

Resource Property	Default (nshttp resource type)
<*METHOD*>_timeout Where <METHOD> is Start, Stop, Validate, Update, Init, Fini, Boot, Monitor_Start, Monitor_Stop, Monitor_Check	300 (Seconds)
Cheap_Probe_Interval	60
Thorough_Probe_Interval	60 (Seconds)
Retry_Count	2
Retry_Interval	300 (Seconds)
Failover_Mode	SOFT
Network_resources_used	Empty
Scalable	FALSE
Load_balancing_policy	LB_WEIGHTED
Load_balancing_weights	Empty
Port_list	80/tcp

Apache Web Server, continued

Resource Group Properties

When configuring the web server as a scalable data service, the following resource group properties should be set properly:

`Maximum_primaries` - Set to the maximum number of nodes you may wish to configure the scalable resource group to run on simultaneously using administrative commands. The RGM will never attempt to bring the resource group online on more than this number of nodes at any one time.

`Desired_primaries` - Set to the number of nodes on which you desire to have the resource group running simultaneously under normal operating conditions. The RGM attempts to maintain this number of active primaries for the resource group. If the number of nodes on which the resource group is online falls below this setting due to hardware or software failures, the RGM will attempt to start the resource group on additional nodes until this setting is satisfied.

`RG_dependencies` - In the group containing the web server resource (i.e. the resource based on the `nshttp` resource type), set this property to the name of the resource group containing the associated SharedAddress resource.

Apache Web Server (Continued)

- Make sure to set the `Maximum_primaries`, `Desired_Primaries` and `RG_Dependencies` resource group properties properly (especially if configuring the web server as a scalable data service)

- Scalable Apache data service example:

Creates 2 resource groups - `ap-server-sa-rg` and `apache-rg` using the scalable address `ap-server`. Note: if you omit the `-n` option in the third `scrgadm` command, below, the appropriate NAFO groups will be discovered automatically.

```
# scrgadm -a -t SUNW.apache
# scrgadm -a -g ap-server-sa-rg -h mars,venus
# scrgadm -a -S -g ap-server-sa-rg -l ap-server \
          -n nafo0@venus,nafo0@mars
# scrgadm -a -g apache-rg        \
          -y Maximum_primaries=2 \
          -y Desired_Primaries=2 \
          -y RG_dependencies=ap-server-sa
# scrgadm -a -j apache-rs -g apache-rg -t SUNW.apache \
          -x Confdir_list=/usr/local/apache/conf      \
          -x Bin_dir=/usr/local/apache/bin            \
          -y Scalable=TRUE                            \
          -y Network_Resources_Used=ap-server

# scswitch -e -j ap-server
# scswitch -e -j apache-rs
# scswitch -o -g ap-server-sa-rg
# scswitch -o -g apache-rg
# scswitch -z -g ap-server-sa-rg -h venus
# scswitch -z -g apache-rg -h venus,mars
```

A shortcut to substitute for the 6 `scswitch` commands listed in the box above:

```
# scswitch -Z -g ap-server-sa-rg
# scswitch -Z -g apache-rg
```

Apache Web Server (Continued)

The following page illustrates an example of configuring an Apache Web Server as a failover data service.

Apache Web Server

➤ Failover data service example:

Creates a resource group, ap-server-lh, containing the logical host address/name of ap-server and an apache resource called apache-res, installed in /usr/local/apache (on all nodes) .
Note: if you omit the -n option in the third scrgadm command, below, the appropriate NAFO groups will be discovered automatically.

```
# scrgadm -a -t SUNW.apache
# scrgadm -a -g ap-server-rg -h venus,mars
# scrgadm -a -L -g ap-server-rg -l ap-server \
          -n nafo0@venus,nafo0@mars
# scrgadm -a -j apache-rs -g ap-server-rg -t SUNW.apache   \
          -x ConfDir_list=/usr/local/apache/conf           \
          -x Bin_Dir=/usr/local/apache/bin

# scswitch -e -j ap-server
# scswitch -e -j apache-rs
# scswitch -e -M -j ap-server,apache-rs
# scswitch -o -g ap-server-rg
# scswitch -z -g ap-server-rg -h venus
```

A shortcut to substitute for the 5 scswitch commands listed in the box above:

```
# scswitch -Z -g ap-server-rg
```

NFS

Application Installation

The NFS server is part of the Solaris Operating Environment, thus there is no explcit application installation required.

1. Place all directories to be shared on cluster file systems. Decide on a logical host address and hostname to be used for the NFS server. Make sure to configure the appropriate name services and cluster node's `/etc/hosts` files with this logical host address.

2. Do not place entries for the directory to be shared via the Sun Cluster 3.0 NFS data service in the `/etc/dfs/dfstab` file on the nodes

3. On a cluster file system, create a directory path to hold the NFS state information (required for NFS lock recovery in case of a node crash) and dfstab file information. This directory will be configured as the `Pathprefix` resource group property.

4. Within the `Pathprefix` directory created in step 3, create a subdirectory called `SUNW.nfs`. This directory will hold the `dfstab` file for the NFS resource. Create a `dfstab.<ResourceName>` file (with the same format as a regular `/etc/dfs/dfstab` file) with the appropriate entries to share the cluster file system directories configured on step 1. The suffix of the `dfstab.<ResourceName>` file should be the name of the NFS resource you are going to create.

   To enable full fault monitoring, the cluster nodes should have full read-write access to the shared file systems. Without read-write access, the fault monitor still functions, but not as thoroughly.

NFS

➤ Application Installation

1. Make sure all directories that will be shared are located on a cluster file system. Configure a logical host address and hostname to be used for the NFS server.

2. Do **not** place entries in the `/etc/dfs/dfstab` file for the directories to be shared via Sun Cluster 3.0’s NFS data service

3. Create a directory path on a cluster file system to hold NFS state information and `dfstab` files for the NFS resource. This directory path will be configured as the `Pathprefix` resource group property

```
# mkdir /global/nfs-data/nfs-admin
```

4. In the directory created in step 3, create a sub-directory named “`SUNW.nfs`”. Create `dfstab.<ResourceName>` files with entries to share the appropriate directories.

```
# mkdir /global/nfs-data/nfs-admin/SUNW.nfs
# vi /global/nfs-data/nfs-admin/SUNW.nfs/dfstab.nfs-server-rs
...
<Add appropriate entries to share cluster file system based directories>
...
# cat /global/nfs-data/nfs-admin/SUNW.nfs/dfstab.nfs-server-rs
share -F nfs -o rw -d”Home Dirs” /global/nfs-data/export/home
share -F nfs -o rw -d”Engineering Data”/global/nfs-data/export/eng-dat
```

NFS (Continued)

RGM Configuration

Extension Properties

The following page lists the extension properties for the `nfs` resource type.

Resource Properties

The following page lists the default values for the `nfs` resource system properties.

Resource Group Properties

In addition to the resource properties and extension properties, the resource group containing an nfs resource must have the `Pathprefix` property set to the directory containing the `SUN.nfs` directory (where the `dfstab.<ResourceName>` files are located).

NFS

- Extension properties:

Extension Property	Description	Default
`Monitor_retry_count`	Number of times to set attempt to restart the fault monitor	4
`Monitor_retry_interval`	The time window (in minutes) to measure fault monitor restarts (See the Monitor_retry_count extension property)	2
`Rpcbind_nullrpc_timeout`	Timeout (seconds) to use when probing rpcbind	120
`Nfsd_nullrpc_timeout`	Timeout (seconds) to use when probing nfsd	120
`Mountd_nullrpc_timeout`	Timeout (seconds) to use when probing mountd	120
`Statd_nullrpc_timeout`	Timeout (seconds) to use when probing statd	120
`Lockd_nullrpc_timeout`	Timeout (seconds) to use when probing lockd	120
`Rpcbind_nullrpc_reboot`	Reboot system when null rpc call on rpcbind fails?	TRUE
`Nfsd_nullrpc_restart`	Restart nfsd when null rpc call fails?	TRUE
`Mountd_nullrpc_restart`	Restart mountd when null rpc call fails?	TRUE

- Resource property defaults for the `nfs` resource type

Resource Property	Default (nshttp resource type)
<*METHOD*>_timeout Where <METHOD> is Start, Stop, Validate, Update, Monitor_Start, Monitor_Stop, Monitor_Check, Prenet_start	300 (Seconds)
Cheap_Probe_Interval	20 (Seconds)
Thorough_Probe_Interval	120 (Seconds)
Retry_Count	2
Retry_Interval	300 (Seconds)
Failover_Mode	HARD
Network_resources_used	Empty

- Make sure to set the `Pathprefix` resource group property to the directory path that contains the `SUNW.nfs` sub-directory.

NFS (Continued)

Example Failover Data Service Configuration

When configuring the resource group, make sure the resource name matches the trailer placed on the `dfstab.<ResourceName>` file placed in the `<Pathprefix>/SUNW.nfs` directory.

The following page lists an example of configuring a resource group containing an `nfs` resource.

Changing Share Options

To change any share options for any NFS shares under SC 3.0 control, make sure to disable the fault monitoring (using `scswitch -n -M -j <NFS resource>` before issuing any share commands. After the share options have been changed, reenable fault monitoring using `scswitch -e -M -j <NFS resource>`.

NFS

➤ Example failover data service configuration

Creates a resource group, `nfs-server-rg`, with two resources:
`nfs-server-lh-rs` is a network address resource with the address `nfs-server`;
`nfs-resource-res` is an nfs resource.

```
# scrgadm -a -t SUNW.nfs
# scrgadm -a -g nfs-server-rg -h venus,mars \
          -y Pathprefix=/global/nfs-data/nfs-admin
# scrgadm -a -L -j nfs-server-lh-rs -g nfs-server-rg -l nfs-server
# scrgadm -a -j nfs-server-rs -g nfs-server-rg -t SUNW.nfs

# scswitch -Z -g nfs-server-rg
```

DNS

Sun Cluster 3.0 is compatible with the version of DNS which is included with Solaris (based on BIND 8). The DNS server (`in.named`) can be integrated with the cluster as a failover data service.

Application Installation

The DNS server application is part of the Solaris Operating Environment, thus no explicit application installation is required.

1. Choose a location on a cluster file system to place the DNS configuration (`named.conf`) and zone files.

2. Determine the logical host address which will be hosting the DNS server. Make sure to enter the logical host address and hostname in the appropriate name services as well as the host files of the cluster nodes.

3. Create a `named.conf` file and any appropriate zone files and place them in the location choosen in step 1.

4. Test the name server by manually starting up and shutting down the server on each node which is eligible to host the name server.

5. Configure the client systems by referencing the *logical* host address in their DNS client setup (e.g. `/etc/resolv.conf`, etc.). The cluster nodes should be configured to try the logical host address first and then the physical node addresses.

DNS

➤ Application installation

1. Choose a location on a cluster file system to place the DNS configuration and zone files

2. Choose a logical host name and address to use for the DNS server

3. Create a valid `named.conf` and appropriate zone files. Place them on the cluster file system location chosen in step 1

```
# /global/dns-data/named.conf
options {
      directory "/global/dns_data/named";
};
zone "." in {
      type hint; file "db.cache";
};
zone "0.0.127.in-addr.arpa" in {
      type master; notify no; file "db.127.0.0";
};
...
# cat /global/dns-data/named/db.127.0.0
@ IN SOA eng.sun.com. root.eng.sun.com (
            1999062501 ; serial number (YYYYMMDD##)
            10800 ; refresh every 3 hours
            1800 ; retry every 3 hours
            604800 ; expire after a week
            86400 ) ; TTL of 1 day
IN NS dns-server.eng.sun.com.
1 IN PTR localhost.
...
```

4. Test the name server independent of the cluster by manually starting up and querying the the name server on each node of the cluster. Make sure to watch `/var/adm/messages` for error messages.

```
# in.named -c /global/dns-data/named.conf
# nslookup localhost
```

5. Configure the client systems to use the logical host name as the primary resolver (don't forget to check the `nsswitch.conf` file also).

```
# cat /etc/resolv.conf
domain eng.sun.com
nameserver dns-server.eng.sun.com
```

DNS (Continued)

RGM Configuration

Extension Properties

The extension properties for the `dns` resource type are listed on the following page

Resource Properties

The resource property defaults for the `dns` resource type are listed on the following page

Resource Group Configuration Example

The following page illustrates the RGM configuration of an example failover data service using the `dns` resource type

DNS (Continued)

➤ Extension properties:

Extension Property	Description	Default
`Confdir_list`	The directory where the `named.conf` file is located	
`Monitor_retry_count`	Number of times to set attempt to restart the fault monitor	4
`Monitor_retry_interval`	The time window (in minutes) to measure fault monitor restarts (See the Monitor_retry_count extension property)	2
`Probe_timeout`	Time out value (in seconds) used for the fault monitor probe	30
`DNS_mode`	Configuration file (named.conf or named.boot) to use	conf

➤ Resource property defaults for the `dns` resource type

Resource Property	Default (nshttp resource type)
<*METHOD*>_timeout Where <METHOD> is Start, Stop, Validate, Update, Init, Fini, Boot, Monitor_Start, Monitor_Stop, Monitor_Check	300 (Seconds)
Thorough_Probe_Interval	60 (Seconds)
Retry_Count	2
Retry_Interval	300 (Seconds)
Failover_Mode	SOFT
Network_resources_used	Empty
Port_list	53/udp

➤ Example failover configuration

Creates a resource group, `dns-server-lh`, containing the logical host address/name of `dns-server`

```
# scrgadm -a -t SUNW.dns
# scrgadm -a -g dns-server-rg -h venus,mars
# scrgadm -a -L -j dns-server-lhrs -g dns-server-rg -l dns-server
# scrgadm -a -j dns-rs -g dns-server-rg -t SUNW.dns   \
          -x ConfDir_list=/global/dns-data
# scswitch -Z -g dns-server-rg
```

Netscape Directory Server (LDAP)

Netscape Directory Server is an LDAP server which can be configured as a failover data service on Sun Cluster 3.0.

Application Installation

1. Determine where you want to install Netscape Directory Server and which logical hostname and address you want to use to host the server. the LDAP server may be should be installed on a cluster file system.
2. Prior to beginning installation, configure and bring online a resource group containing the logical host resource based on the logical host address and hostname determined in step 1. The installation will fail unless the logical host address is active on the node. We add the nsldap resource to this group after the Netscape LDAP server installation is complete.
3. Run the setup program to install the Netscape Directory Server in the desired location. Make sure to use the configured logical hostname when prompted for computer name information during the installation.
4. Further configuration of the LDAP server can be performed from the Netscape Server Console
5. Test to make sure that the server can be started and shut down manually on the nodes which will host the LDAP server

Netscape Directory Server (LDAP)

➤ Application Installation

1. Determine the installation path and the logical hostname/address to be used for the LDAP server. Make sure that the logical hostname and address are entered into the appropriate name services. The installation should be performed on a cluster file system.

2. Configure a resource group containing a logical host resource based on the logical hostname/address from step 1. Bring this resource group online on one of the cluster nodes.

```
# scrgadm -a -g ldap-rg -h venus,mars
# scrgadm -a -L -j ldap-server-lhrs -g ldap-rg -l ldap-server
# scswitch -Z -g ldap-rg
```

3. From the node hosting the resource group, locate the Netscape Directory Server CD image, begin the installation using the `setup` program. When prompted for “Install location”, make sure to enter the location determined in step 1. When prompted for Computer Name, make sure to use the logical hostname, not the physical hostname of the node.

```
# cd <Location of Netscape Directory Server 4.1 CDROM image>
# ./setup
...
Install location [/usr/netscape/server4]: /global/ldap-data
...
Computer name [venus.eng.sun.com]: ldap-server.eng.sun.com
...
```

4. Configure the LDAP server using the Netscape Server Console utility

5. Test to make sure that the server can be shut down and started manually on all nodes on which it may be running

```
# cd <LDAP install location>/slapd-<ServerName>
# ./start-slapd
# ./stop-slapd
```

Netscape Directory Server (LDAP) (Continued)

RGM Configuration

Once the application has been configured, add a resource of type `nsldap` to the existing resource group.

Extension Properties

The extension properties for the nsldap resource type are listed on the following page

Resource Properties

The resource property defaults for the nsldap resource type are listed on the following page

Resource Group Configuration Example

The following page illustrates an example of configuring a Netscape Directory Server as a failover data service.

Netscape Directory Server (LDAP)

➤ Extension properties

Extension Property	Description	Default
`Confdir_list`	The directory containing the `start-slapd` and `stop-slapd` scripts for the LDAP server (`<ServerRoot>/slapd-<ServerName>`)	`None`
`Monitor_retry_count`	Number of times to set attempt to restart the fault monitor	4
`Monitor_retry_interval`	The time window (in minutes) to measure fault monitor restarts (See the Monitor_retry_count extension property)	2
`Probe_timeout`	Time out value (in seconds) used for the fault monitor probe	30

➤ Resource property defaults for `nsldap` resource type

Resource Property	Default (nshttp resource type)
<*METHOD*>_timeout Where <METHOD> is Start, Stop, Validate, Update, Monitor_Start, Monitor_Stop, Monitor_Check	300 (Seconds)
Thorough_Probe_Interval	60 (Seconds)
Retry_Count	2
Retry_Interval	300 (Seconds)
Failover_Mode	SOFT
Network_resources_used	Empty
Port_List	389/tcp

➤ Failover data service example

```
Register the resource type
# scrgadm -a -t SUNW.nsldap
Add an LDAP resource, ldap-rs, to the resource group, ldap-rg, which was created
and brought online earlier
# scrgadm -a -j ldap-rs -g ldap-rg -t SUNW.nsldap \
          -x Confdir_list=/global/ldap-data/slapd-ldap-server
Bring the newly created LDAP resource online
# scswitch -Z -g ldap-rg
```

Oracle8i

Oracle8i can be installed Sun Cluster 3.0 as either a failover application only or with the Parallel Server option (OPS). This section will cover the installation of Oracle8i as a failover data service.

Application Installation

1. Determine the location for the following Oracle8i components:
   a. Application binaries (can be global or local)
   b. Database data files - Control Files, Tablespaces, etc (should be placed on a global device(s) or a global file system(s) only)

2. Determine the logical host address and hostname to be used for the database. Make sure to enter this hostname and address in the appropriate name services and cluster host files.

3. Create a UNIX group called "`dba`", together with a UNIX user (as a member of the `dba` group) to serve as the Oracle database administrator user. This user and group must be created identically on each node of the cluster which will run the Oracle data service.

4. Install Oracle using the `Installer` utility in the `root` directory of the Oracle8i CD-ROM image (this must be done as the UNIX user created in step 3, not as root). The ORACLE_HOME and ORACLE_BASE parameters should be set based on the decisions made in step 1.

   Note: This step, as well as the rest of the Oracle8i configuration steps (through Step 7) should be performed on one node of the cluster only

Oracle8i

➤ Application Installation

1. Determine where you want to install the Oracle application binaries and data files. The application binaries can be installed either locally or globally. The database data files must be installed on a globally accessible device(s) (if using raw data files) or a cluster file system(s)

2. Determine the logical host address and hostname to be used for the Oracle server. Make sure to enter this hostname/address in the appropriate name services and in the cluster node's host files.

3. Create a UNIX group called "`dba`". Also create a user (whose primary group is `dba`) to serve as the Oracle Database Administrator user. Make sure to create the user and group identically on each node of the cluster.

```
# hostname
mars
# cat /etc/passwd
...
oracle:x:100:125:Oracle DBA:/global/oracle:/sbin/sh
...
# cat /etc/groups
...
dba::125:oracle
...
```

```
# hostname
venus
# cat /etc/passwd
...
oracle:x:100:125:Oracle DBA:/global/oracle:/sbin/sh
...
# cat /etc/groups
...
dba::125:oracle
...
```

4. Install Oracle to the desired location (as chosen in step 1) using `orainst`.

```
# su - oracle
$ cd <Location of Oracle8i CD>
$ ./Installer
```

Oracle8i (Continued)

5. Configure the database (Create databases, load data, etc.) as outlined in the Oracle8i documentation and per local requirements.

6. Sun Cluster 3.0's Oracle8i fault monitor periodically logs in to the monitored database to determine if the database is healthy. The database user to be used for this purpose must have the following privleges:

   a. CREATE SESSION

   b. CREATE TABLE

   c. SELECT priviliges on sys.`v_$sysstat`

   You may also want to assign a default tablespace with a 1M quota (the fault monitor will periodically create, insert into and drop a table to determine the responsiveness of the database).

7. Configure the Net8 Listener component appropriately. Make sure to provide the Logical Host name and not the physical host name of the node when configuring the `listener.ora` file.

8. If $ORACLE_HOME is not located on a cluster file system (i.e. you installed Oracle locally), copy the entire $ORACLE_HOME directory tree to the same location on the other node(s) in the cluster.

9. Replicate the contents of `/var/opt/oracle` to the other nodes of the cluster (this must be done even if you installed Oracle on a cluster file system)

10. Configure the client systems - make sure to reference the Logical Host in the `tnsnames.ora` file

Oracle8i

5. Create the Oracle database. Data files should be placed on global devices and/or cluster file systems.

6. Configure an Oracle user to for the Sun Cluster 3.0 Oracle fault monitor.

```
$ svrmgrl

Oracle Server Manager Release 3.1.6.0.0 - Production

(c) Copyright 1997, Oracle Corporation.  All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
PL/SQL Release 8.1.6.0.0 - Production

SVRMGR> connect internal;
Connected.
SVRMGR> create user sc_fm identified by sc_fm;
Statement processed.
SVRMGR> grant create session, create table to sc_fm;
Statement processed.
SVRMGR> grant select on v_$sysstat to sc_fm;
Statement processed.
SVRMGR> alter user sc_fm default tablespace users quota 1m on users;
Statement processed.
```

7. Configure the Net8 Listener (`listener.ora` file)

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS= (PROTOCOL= TCP)(Host= ora-server)(Port= 1521))
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME= /global/oracle)
      (SID_NAME = ORCL)
    )
  )
```

8. If Oracle was installed locally, copy the entire ORACLE_HOME directory tree to the other nodes in the cluster, otherwise, continue to step 9

```
# cd <Location of Oracle Installation ($ORACLE_HOME)>
# tar cvf - . | (rsh mars "cd <Location of Oracle Installation>;tar xf -")
```

Oracle 8i (Continued)

9. Replicate the contents of `/var/opt/oracle` to the other nodes of the cluster (this must be done even if you installed Oracle on a cluster file system)

10. Configure the client systems - make sure to reference the Logical Host in the `tnsnames.ora` file

11. Test the installation, make sure that the database can be started up and shut down on all nodes of the cluster (or at least the nodes that will be configured to host the Oracle data service). Also test that you are able to connect to the database using the login and password you created (or modified) for the database fault monitor.

Oracle8i (Continued)

9. Copy `/var/opt/oracle/*` to the other nodes of the cluster

```
# tar cvf /var/opt/oracle/* | (rsh mars “tar xf -”)
```

10. Configure the client system’s `tnsnames.ora` file

```
#
# Installation Generated Net8 Configuration
# Version Date: Oct-27-97
# Filename: Tnsnames.ora
#
orcl =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL= TCP)(Host= ora-server)(Port= 1521))
    (CONNECT_DATA = (SID = ORCL))
  )
```

11. Test the installation

```
# su - oracle
$ svrmgrl
...
SVRMGR> connect internal
Connected.
SVRMGR> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SVRMGR> startup
ORACLE instance started.
Total System Global Area                        4775440 bytes
Fixed Size                                        48656 bytes
Variable Size                                   4235264 bytes
Database Buffers                                 409600 bytes
Redo Buffers                                      81920 bytes
Database mounted.
Database opened.
SVRMGR> exit;
Server Manager complete.

$ sqlplus sc_fm/sc_fm
...
SQL> select * from sys.v_$sysstat;
...
```

Oracle8i

RGM Configuration

The Sun Cluster HA for Oracle data service is made up of two resource types, the oracle_server resource type and the oracle_listener resource type. The extension and resource properties as well as example RGM configurations for both resource types are outlined in this section.

Extension Properties

The extension properties for the oracle_server and oracle_listener resource types are listed on the following page.

Oracle 8

➤ Oracle_server extension properties

Extension Property	Description	Default
Alert_log_file	The full path to the alert_<SID>.log file for the instance	None
Connect_cycle	The number of probe cycles to stay connected to the database before the fault monitor disconnects	5
Connect_string	The Oracle connect string which the fault monitor should use to connect to the database. Format is <user>/<password> (where <user>/<password> should be the login and password configured for the fault monitor). A value of "/" indicates that Oracle is using OS authentication.	None
ORACLE_HOME	The Oracle home directory	None
ORACLE_SID	Oracle Instance Name	None
Parameter_file	The full path to the Oracle instance's parameter file (pfile), usually named init<SID>.ora	Oracle default parameter file
Probe_timeout	The time, in seconds, after which a running database probe is aborted by the fault monitor and a timeout condition is set	60
User_env	A fully qualified path name to a file containing environment variables to be set before performing database startup or shut down	None
Wait_for_online	Boolean variable which, if TRUE, waits in the start method for the database to come online	TRUE
Debug_level	Debug level for logging messages	1

➤ Oracle_listener extension properties

Extension Property	Description	Default
ORACLE_HOME	The Oracle home directory	None
Listener_name	The name of the listener resource (as defined in the listener.ora file)	None
User_env	A fully qualified path name to a file containing environment variables to be set before performing database startup or shut down	None
Debug_level	Debug level for logging messages	1

Oracle8i

Resource Properties

The resource property defaults for the `oracle_server` and `oracle_listener` resource type are listed on the following page

Oracle8i

➤ Oracle_server resource property defaults

Resource Property	Default
Start_timeout, Stop_timeout	300 (Seconds)
Validate_timeout, Update_timeout, Monitor_start_timeout, Monitor_stop_timeout	120 (Seconds)
Init_timeout, Fini_timeout, Boot_timeout	30
Thorough_Probe_Interval	30 (Seconds)
Retry_Count	2
Retry_Interval	600 (Seconds)
Failover_Mode	SOFT

➤ Oracle_listener resource property defaults

Resource Property	Default
Start_timeout, Stop_timeout, Validate_timeout, Update_timeout	6 (0Seconds)
Init_timeout, Fini_timeout, Boot_timeout, Monitor_start_timeout, Monitor_stop_timeout	30
Thorough_Probe_Interval	30 (Seconds)
Retry_Count	-1
Retry_Interval	600 (Seconds)
Failover_Mode	NONE

Oracle8i

The following page illustrates an example of configuring an Oracle8i database as a failover data service.

Oracle 8

➤ RGM Configuration example

Register the resource types

```
# scrgadm -a -t SUNW.oracle_server
# scrgadm -a -t SUNW.oracle_listener
```

Create the resource group

```
# scrgadm -a -g orcl-rg -h venus,mars
```

Add the resources

```
# scrgadm -a -L -j ora-server-lhrs -g orcl-rg -l ora-server
# scrgadm -a -j orcl-db-rs -g orcl-rg -t SUNW.oracle_server \
          -x Oracle_sid=ORCL -x Oracle_home=/global/oracle \
          -x Alert_log_file=/global/oracle/admin/orcl/bdump/alert_orcl.log \
          -x Parameter_file=/global/oracle/admin/orcl/pfile/initorcl.ora   \
          -x Connect_string=sc_fm/sc_fm
# scrgadm -a -j orcl-listener-rs -g orcl-rg -t SUNW.oracle_listener \
          -x Oracle_home=/global/oracle -x Listener_name=LISTENER
```

Enable the resources

```
# scswitch -e -j ora-server-lhrs
# scswitch -e -j orcl-db-rs,orcl-listener-rs
```

Bring the resource group online

```
# scswitch -o -g orcl-rg
# scswitch -z -g orcl-rg -h venus
```

Shortcut: Enable the resources, make the resource group managed, and bring the resource group online by executing one `scswitch` command instead of the 4 `scswitch` commands in the box above:

```
# scswitch -Z -g orcl-rg
```

CHAPTER 7

Sun Cluster 3.0 Administration

Objectives

Purpose

Once the cluster is installed and configured, there are some day-to-day maintenance tasks which must be performed. This chapter will outline the administrative tools available and how to perform basic cluster administrative tasks.

Prerequisites

Understanding of Sun Cluster architecture and concepts

Understanding of Sun Cluster installation and configuration

Understanding of Sun Cluster RGM and Data Service configuration

Solaris System Administration

Objectives

Upon completion of this chapter, the participant will be able to:

- Describe the available cluster administration commands and utilities
- Perform basic cluster administration tasks
- Install, configure and use Sun Management Center-based cluster monitoring

Objectives

- Describe the available administration commands and utilities
- Perform basic cluster administration tasks
- Install, configure and use the Sun Management Center-based cluster monitoring

Administrative Commands and Utilities

Sun Cluster 3.0 comes with a number of commands and utilities which can be used to administer the cluster. Refer the man pages for more detailed information on each of these commands and utilities.

The SC 3.0 administrative commands and utilities are:

- **`scinstall`** - Installs cluster software and initializes cluster nodes
- **`scconf`** - Updates the Sun Cluster software configuration
- **`scsetup`** - Interactive Sun Cluster configuration tool
- **`sccheck`** - checks and validates Sun Cluster configuration
- **`scstat`** - displays the current status of the Cluster
- **`scgdevs`** - Administers the global device namespace
- **`scdidadm`** - Disk ID configuration and administration utility
- **`scshutdown`** - Utility to shutdown a cluster node or cluster
- **`scrgadm`** - Manages registration and configuration of resource types, resources and resource groups
- **`scswitch`** - Performs ownership or state changes of Sun Cluster resource groups and disk device groups
- **`pnmset`** - Sets up and updates the configuration for Public Network Management (PNM)
- **`pnmstat`** - Report status for Network Adapter Failover (NAFO) groups managed by PNM
- **`pnmptor, pnmrtop`** - Maps pseudo adapter to real adapter name (`pnmptor`) or real adapter to pseudo adapter name (`pnmrtop`) in NAFO groups
- **`ccp`** - Cluster control panel (administrative console)
- **`cconsole, ctelnet, crlogin`** - Multi window, multi machine remote console, telnet or rlogin (administrative console)

Administrative Commands and Utilities

- ➤ scinstall
- ➤ scconf
- ➤ scsetup
- ➤ sccheck
- ➤ scstat
- ➤ scgdevs
- ➤ scdidadm
- ➤ scshutdown
- ➤ scrgadm
- ➤ scswitch
- ➤ pnmset
- ➤ pnmstat
- ➤ pnmrtop, pnmptor
- ➤ ccp
- ➤ cconsole, ctelnet, crlogin

Administrative Commands and Utilities

scinstall

Used to install Sun Cluster software and initialize new cluster nodes. When run with no arguments, `scinstall(1M)` will run in an interactive fashion, presenting the user with menus and prompts to perform its tasks. scinstall can also be run in a non-interactive mode by supplying the proper command line arguments.

`scinstall` is located in the `SunCluster_3_0/Tools` directory of the Sun Cluster 3.0 CDROM, or in `/usr/cluster/bin` on a node where Sun Cluster has already been installed.

All forms of `scinstall` affect only the node it is run on.

Adminstrative Commands and Utilities

➤ scinstall

To run `scinstall` interactively:

```
scinstall
```

To install Sun Cluster software and/or initialize a node as a new Sun Cluster member:

```
scinstall -i [-k] [-d <cdimage_dir>] [-s <srvc>,...]
             [-N <clusternode>
                 [-C <clustername>] [-T <authentication_options>]
                 [-G {<special> | <filesystem>} ] [-A <adapter_options>]
                 [-B <junction_options>] [-m <cable_options>]
                 [-w [<netaddr_options>]
             ]
```

To upgrade a Sun Cluster node:

```
scinstall -u [-d <cdimage_dir>] [-s <srvc>,...]
             [-N <clusternode>
                 [-C <clustername>] [-G{<special> | <filesystem>]
                 [-T authenticaion_options]
             ]
```

To setup a Sun Cluster install server (copies the CD-ROM image to an install directory:

```
scinstall -a <install_dir> [-d <cdimage_dir>]
```

To establish the given nodename as an installation client of the install server:

```
scinstall -c <jumpstart_dir> -h <nodename> [-d <cdimage_dir>] [-s <srvc>,...]
             [-N <clusternode>
                 [-C <clustername>] [-G {<special> | <filesystem>}]
                 [-T <authentication_options>] [-A <adapter_options>]
                 [-B <junction_options>] [-m <cable_options>]
                 [-w <netaddr_options>]
```

To print the release and package version information:

```
scinstall -p [-v]
```

Administrative Command and Utilities (Continued)

scconf

scconf(1M) is used to manage the cluster software configuration. It is used to add new items to the configuration, change the properties of already configured items and remove items from the configuration. scconf can be run from any node which is a member of the cluster (and is usually only run on one node). Items which scconf manages include:

- ❏ Quorum options
- ❏ Disk device groups (SDS disksets, VxVM disk groups or global raw devices)
- ❏ The name of the cluster
- ❏ Add or remove cluster nodes
- ❏ Cluster transport adapters, junctions and cables
- ❏ Private hostnames for the nodes (hostname used over the cluster transport)
- ❏ Node authentication options

When used with the -p option, scconf will print out the current cluster configuration

scconf is located in /usr/cluster/bin.

Administrative Command and Utilities (Continued)

➤ scconf

To add or initialize a new item to the software configuration (e.g. A new node, transport adapter, junction or cable, quorum device, device group or authentication option)

```
scconf -a [-Hv] [-h <node_options>] [-A <adapter_options>] [-B <junction_options>]
          [-m <cable_options>] [-p <privatehostname_options>]
          [-q <quorum_options>] [-D <devicegroup_options>]
          [-T <authentication_options>]
```

To change the options for an exisiting item in the software configuration (e.g. The cluster name, a transport adapter, junction or cable, the private hostnames, quorum devices, device groups options and authentication options):

```
scconf -c [-Hv] [-c <cluster_options] [-A <adapter_options>]
          [-B <junction_options>] [-m <cable_options>]
          [-P <privatehostname_options>] [-q <quorum_options>]
          [-D <devicegroup_options>] [-T <authentication_options>]
```

To remove an item from the software configuration (e.g. A node, adappter, junction, cable, quorum device, device group or authentication):

```
scconf -r [-Hv] [-h <node_options>] [-A <adapter_options>] [-B <junction_options>]
          [-m <cable_options>] [-q <quorum_options>] [-D <devicegroup_options>]
          [-T <authentication_options>]
```

To print out the current configuration:

```
scconf -p [-Hv}
```

To print help information about the command options:

```
scconf [-H]
```

➤ General Notes:

- ➤ Each form of the command will accept a -H option. If present, this option will cause scconf to print out help information (specific to the form of the command used) and ignore any other options given.
- ➤ The suboptions (e.g. <node_options>, <adapter_options>, etc.) take the form of *attribute=value*, such as:

Adds a new adapter, hme3, on node venus

```
# scconf -a -A trtype=dlpi,name=hme3,node=venus
```

Administrative Command and Utilities (Continued)

scsetup

scsetup is an interactive, menu driven utility which can perform most of the post-installation cluster configuration tasks which are handled by scconf. scsetup should be run immediately after the cluster software has been installed and all nodes have joined the cluster (scsetup will automatically detect the new installation and prompt for the proper quorum configuration information automatically).

scsetup can be run from any node of the cluster.

Administrative Commands and Utilities

➤ scsetup

```
# scsetup
  *** Main Menu ***

    Please select from one of the following options:

        1) Quorum
        2) Cluster interconnect
        3) Private hostnames
        4) Device groups
        5) New nodes
        6) Other cluster properties

        ?) Help with menu options
        e) Exit

    Option:
```

Administrative Commands and Utilities (Continued)

sccheck

`sccheck` is a utility which, when run on a node of the cluster (can be run on any node currently in the cluster) checks the validity of the cluster configuration. It checks to make sure that the basic configuration of the cluster is correct and consistent across all nodes.

Options can be given to `sccheck` to invoke a brief check, print verbose messages, suppress warning messages or to perform the check on only certain nodes of the cluster.

Administrative Commands and Utilities (Continued)

➤ sccheck

```
sccheck [-bvW] [-h <hostlist>
   -b : perform a brief check
   -v : verbose mode
   -W : disable warnings
   -h : Run check on specific hosts
```

```
# sccheck -v
vfstab-check: CHECKED - Check for node id
vfstab-check: CHECKED - Check for node id
vfstab-check: CHECKED - Check for /global/.devices/node@<id>
vfstab-check: CHECKED - Check for mount point
vfstab-check: CHECKED - Check for identical global entries
vfstab-check: CHECKED - Check for option ‘syncdir’
vfstab-check: CHECKED - Check for physical connectivity
vfstab-check: CHECKED - Check for option ‘logging’ for raw device
vfstab-check: CHECKED - vfstab check completed.
```

Administrative Commands and Utilities (Continued)

scstat

scstat (1M) is a utility which prints out the current status of various cluster components. It can be used to display the following information:

- ❑ The cluster name
- ❑ List of cluster members
- ❑ Status of each cluster member
- ❑ Status of resource groups and resources
- ❑ Status of every path in the cluster interconnect
- ❑ Status of every disk device group
- ❑ Status of every quorum device

Administrative Commands and Utilities (Continued)

➤ scstat

```
scstat -[-DWgnpq] [-h node]
        -D - Disk group status, -W - interconnect status,
        -g - resource group status, -n node status, -p - all components status,
        -q - quorum device status
```

```
# scstat -g
Resource Group
  Resource Group Name:                             netscape-rg
  Status
    Node Name:                                     venus
    Resource Group State:                          Online

    Node Name:                                     mars
    Resource Group State:                          Offline

  Resource
    Resource Name:                                 netscape-server
    Status
      Node Name:                                   venus
      Resource Monitor Status/Message:             Online - SharedAddress online
      Resource State:                              Online

      Node Name:                                   mars
      Resource Monitor Status/Message:             Offline - SharedAddress offline
      Resource State:                              Offline

  Resource Group Name:                             netscape-rg-2
  Status
    Node Name:                                     venus
    Resource Group State:                          Online

    Node Name:                                     mars
    Resource Group State:                          Online

  Resource
    Resource Name:                                 netscape-res
    Status
      Node Name:                                   venus
     Resource Monitor Status/Message:            Online - Successfully started Netscape Web
Server for resource <netscape-res>.
      Resource State:                              Online

      Node Name:                                   mars
     Resource Monitor Status/Message:            Online - Successfully started Netscape Web
Server for resource <netscape-res>.
      Resource State:                              Online
```

Administrative Commands and Utilities (Continued)

scgdevs

scgdevs is used to manage the global devices namespace. The global devices namespace is mounted under /global and consists of a set of symbolic links to physical device files.

By calling scgdevs, an administrator can attach new global devices (such as a tape drive, CD-ROM drive or disk drive) to the global devices namespace without requiring a system reboot. The drvconfig(1M) and disks(1M), tapes(1M) or devlinks(1M) commands must be run prior to running scgdevs. Also run devfsadm(1M) before running scgdevs.

This command should be run on the node (the node must be a current cluster member) where the new device is being installed.

Administrative Commands and Utilities (Continued)

- ➤ scgdevs

```
# drvconfig
# disks
# devfsadm
# scgdevs
Configuring DID devices
Configuring the /dev/global directory (global devices)
obtaining access to all attached disks
reservation program successfully exiting
```

Administrative Commands and Utilities (Commands)

scdidadm

The scdidadm command is used to administer the disk ID (DID) pseudo device driver. It can create driver configuration files, modify entries in the configuration file, load the current configuration files into the kernel and listing the mapping between DID devices and the physical devices.

The scdidadm command is run during cluster startup to initialize the DID driver. It is also used by the scgdevs(1M) command to update the DID driver. The primary use of the scdidadm command by administrator will be to list the current DID device mappings.

The scdidadm command can be run from any node of the cluster.

Administrative Commands and Utilities (Continued)

➤ scdidadm

To perform a consistency check against the kernel representation of the devices and the physical devices:

```
scdidadm -c
```

To remove all DID references to underlying devices which have been detached from the current node (Use after running the normal Solaris device commands to remove references to non-existent devices):

```
scdidadm -C
```

To print out the DID device mappings:

```
scdidadm -l | -L [-h] [-o <fmt>,...] [<path> | <DID_instance>]
   fmt can be instance, path, fullpath, host, name, fullname, diskid or asciidiskid
```

To reconfigure the DID database to add any new devices (this is perfromed by `scgdevs`):

```
scdidadm -r
```

To replace a disk device in the DID database:

```
scdidadm -R <path> | <DID_instance>
```

To run `scgdevs(1M)` on each member of the cluster:

```
scdidadm -S
```

To initialize and load the DID configuration into the kernel:

```
scdidadm -ui
```

To print the verson number of this program:

```
scdidadm -v
```

```
# scdidadm -hlo instance,host,path,name
Instance Host       Physical Path        Pseudo Path
1        venus      /dev/rdsk/c0t0d0     d1
2        venus      /dev/rdsk/c1t2d0     d2
3        venus      /dev/rdsk/c1t3d0     d3
4        venus      /dev/rdsk/c1t4d0     d4
5        venus      /dev/rdsk/c1t5d0     d5
6        venus      /dev/rdsk/c2t2d0     d6
7        venus      /dev/rdsk/c2t3d0     d7
8        venus      /dev/rdsk/c2t4d0     d8
9        venus      /dev/rdsk/c2t5d0     d9
```

Administrative Commands and Utilities (Continued)

scswitch

The `scswitch` command is used to perform the following tasks:

- ❏ Switch resource groups or disk device groups to new primary nodes (`scswitch -z ...`)
- ❏ Bring resource groups or disk device groups online or offline (`scswitch -z ...` or `scswitch -m ...`)
- ❏ Restart a resource group on a node (`scswitch -R ...`)
- ❏ Enable or disable resources and resource monitors (`scswitch -e|-n ...`)
- ❏ Switch resource groups to or from an "unmanaged" state (`scswitch -o|-u ...`)
- ❏ Clear error flags on resources (`scswitch -c ...`)
- ❏ Bring resource group offline on all nodes: `scswitch -F -g ....`
- ❏ Enable all resources, make resource group managed, and bring resource group online on default master(s): `scswitch -Z -g [optional]...`

The `scswitch` command can be run on any node of the cluster.

Admistrative Commands and Utilities (Continued)

➤ scswitch

To switch the primary for a resource group (or bring the resource group online if it is not online on any node):

```
scswitch -z -g <resource_grp>[,<resource_grp>...] -h <node>[,<node>...]
```

To switch the primary for a disk device group (or bring the disk device group online if it is currently offline):

```
scswitch -z -D <device_group_name>[,<device_group_name>...] -h <node>[,<node>...]
```

To place a resource group offline:

```
scswitch -z -g <resource_grp>[,<resource_grp>...] -h “”
```

To place a disk device group offline (places the disk device group into “maintenance mode”):

```
scswitch -m -D <device_group_name>[,<device_group_name>...]
```

To restart a resource group on a node:

```
scswitch -R -g <resource_group>[,<resource_group>...] -h <node>[,<node>...]
```

To enable a resource or resource monitor:

```
scswitch -e [-M] -j <resource>[,<resource>...]
```

To disable a resource or resource monitor:

```
scswitch -n [-M] -j <resource>[,<resource>...]
```

To make a resource group “managed” (i.e. bring the resource group under cluster control):

```
scswitch -o -g <resource_grp>[,<resource_grp>...]
```

To make a resource group “unmanaged” (i.e. take the resource group away from cluster control):

```
scswitch -u -g <resource_grp>[,<resource_grp>...]
```

To clear a resource’s error flags:

```
scswitch -c -h <node>[,<node>...] -j <resource>[,<resource>...] -f <flag_name>
```

flag_name can be: BOOT_FAILED, UPDATE_FAILED, INIT_FAILED, FINI_FAILED or STOP_FAILED
(before clearing a STOP_FAILED flag, make sure that the data service is actually down)

NOTE: Only STOP_FAILED is currently implemented of all of these “_FAILED” flags.

Administrative Commands and Utilities

scshutdown

The `scshutdown` command will shutdown the entire cluster.

When shutting down the entire cluster, `scshutdown` performs the following tasks:

1. Places all of the functioning resource groups on the cluster into an offline state. If any of the transitions fail, `scshutdown` will be aborted.
2. Unmounts all of the cluster file systems. If any of the unmounts fail, `scshutdown` will be aborted.
3. Shutdown all of the active device services. If any of the transitions fail, `scshutdown` will be aborted.
4. Runs `/usr/sbin/init 0` on all nodes

The `scshutdown` command should be run on only one node.

The normal Solaris `shutdown` command can be used to shutdown an individual node in the cluster.

Administrative Commands and Utilities (Continued)

➤ scshutdown

To shutdown all nodes in the cluster:

```
scshutdown [-g <grace_period>] [-y] [-f] [<message>]
    -y suppresses the confirmation messages, so the command can be run
       without user intervention
    -f forces the shutdown of the node, even if the switchover of resource
       groups or disk device groups fail
    <message> Optional message to be sent following the standard shutdown message
              (e.g. the message that states “The system will be shut down in ...”)
```

Administrative Commands and Utilities (Continued)

scrgadm

The `scrgadm` command is used for the following tasks:

- ❑ Add, change or remove resource types
- ❑ Create, change the properties of or remove resource groups
- ❑ Add, change the properties of or remove resources within resource groups, including logical hostname or shared address resources
- ❑ Print the properties of resource groups and their resources

The `scrgadm` command can be run on any node which is a member of the cluster.

Administrative Commands and Utilities

➤ scrgadm

To register a resource type:

```
scrgadm -a -t <resource_type_name> [-h <RT_installed_node_list]
        [-f <registration_file_path>]
```

To de-register a resource type:

```
scrgadm -r -t <resource_type_name>
```

To create a new resource group:

```
scrgadm -a -g <RG_name> [-h <nodelist>] [-y <property=value> [...]]
    Use -y Maximum_primaries=n (n>1) to create a scalable resource group.
```

To add a logical hostname or shared address resource to a resource group:

```
scrgadm -a -g <RG_name> -l <hostnamelist> [-n <netiflist>]
```

To add a resource to a resource group:

```
scrgadm -a -j <resource_name> -t <resource_type_name> -g <RG_name>
        [-y <property=value> [...]] [-x <extension_property=value> [...]]
```

To change the properties of a resource group:

```
scrgadm -c -g <RG_name> -y <property=value> [-y <property=value>]
```

To change the properties of a resource:

```
scrgadm -c -j <resource_name> [-y <property=value> [...]]
        [-x <extension_property=value> [...]]
```

To remove a resource from a resource group:

```
scrgadm -r -j <resource_name>
    A resource must be disabled (using scswitch -n) before it can be removed
```

To remove a resource group:

```
scrgadm -r [-L|-S] -g <RG_name>
    Before removing a resource group, perform the following steps:
      1. Place the resource group offline (scswitch -z -g <RG_name> -h “”)
      2. Disable all resources (scswitch -n -j <resource_name>)
      3. Remove all resources (scrgadm -r -j <resource_name>
      4. Make the resource group unmanaged (scswitch -u -g <RG_name>)
```

To print out the resource types, resource groups and resources (and their properties) in the cluster:

```
scrgadm -p[v[v]]
    The additional -v flags will provide more verbose output
```

Administrative Commands and Utilities (Continued)

pnmset

The pnmset utility is used to configure Network Adapter Failover (NAFO) groups on a node. It can be used to:

- ❑ Create, change or remove a NAFO group
- ❑ Migrate IP addresses from the active adapter to a configured standby adapter
- ❑ Print out the current NAFO group configuration

pnmset can be run interactively or, if backup groups have already been configured, non-interactively.

pnmset will only affect the node it is run on, thus must be run separately on each node of the cluster.

Administrative Commands and Utilities (Continued)

- ➤ pnmset

To create NAFO groups (initial setup):

```
pnmset [-f <filename>] [-n[-t]] [-v]
    Where:  -f <filename> indicates a filename to save or read the configuration
               to/from (see pnmconfig(4)).  Default is /etc/cluster/pnmconfig.
            -n Do not run interactively, instead read configuration file for
               NAFO group information (see pnmconfig(4) for file format)
            -t Do not run a test of the interfaces before configuring the
               NAFO groups (only valid with the -n option)
            -v Do not start or restart the pnmd daemon, verify only.  Any new
               groups will not be active until the daemon is restarted.
```

To reconfigure PNM (after the PNM service has already been started):

```
pnmset -c <NAFO_group> -o <subcommand> [<subcommand args> ...]
   Subcommands:  create [<adp1> <adp2> ...] - creates a new NAFO group
                 delete - deletes the NAFO group
                 add <adp> - adds the specified adapter to the NAFO group
                 remove <adp> - removes the specified adapter from the NAFO group
                 switch <adp> - moves the IP addresses from the current live
                                adapter to the specified adapter
```

To print out the current NAFO group configuration:

```
pnmset -p
```

Administrative Commands and Utilities (Continued)

pnmstat

pnmstat will report the current status of the NAFO groups configured on a node. It will report the following information:

1. Status of the NAFO groups:
   a. OK - The NAFO group(s) are working
   b. DOUBT - The NAFO group(s) are currently in a transition state. PNM has not determined if the group is healthy or down.
   c. DOWN - The NAFO group is down, no adapters in the group are capable of hosting the configured IP addresses.
2. Seconds since last failover
3. Currently active adapter

If run without any arguments, pnmstat will simply display the overall status of PNM on the node. When run with a specific NAFO group (-c) or with the -l option it will display all three statistics.

pnmstat will report only on the node it is run on unless the -h option is given, in which case it will report the NAFO group status on the specified host.

Administrative Commands and Utilities (Continued)

➤ pnmstat

To report the general status of PNM on a node:

```
pnmstat
```

To report the status of all the NAFO groups on a node:

```
pnmstat -l
```

To report the status of a particular NAFO group on a node:

```
pnmstat -c <NAFO_group>
```

To report the status of the NAFO groups on another node:

```
pnmstat -h <host> [-s] [-c <NAFO_group>] [-l]
   -s indicates that the cluster interconnect should be used instead of the public
      network
```

➤ Examples:

```
# pnmstat
OK
# pnmstat -c nafo0
OK
NEVER
hme0
# pnmstat -l
group   adapters        status  fo_time act_adp
nafo0   hme0            OK      NEVER   hme0
# pnmstat -h venus -c nafo0
OK
NEVER
hme0
```

Administrative Commands and Utilities (Continued)

pnmptor and pnmrtop

pnmrtop and pnmptor map NAFO group names (pseudo adapter names) to actual network adapter names and vice versa.

These commands will only report on NAFO groups which are configured on the local node.

Administrative Commands and Utilities (Continued)

➤ pnmptor

To convert a NAFO group name to the name of the currently active adapater:

```
pnmptor <NAFO_group>
```

```
# pnmptor nafo0
hme0
```

➤ pnmrtop

To display the NAFO group for a specified network adapter:

```
pnmptor <adp>
```

```
# pnmrtop hme0
nafo0
```

Monitoring Sun Cluster 3.0 using Sun Management Center

Sun Cluster 3.0 can be monitored using Sun Management Center 2.0.1 (aka SyMON)

Installation

Installation of Sun Management Center for use with Sun Cluster 3.0 consists of the following steps:

1. Identify and configure the Sun Management Center console, server and help server. Refer to the Sun Management Center documentation for installation and configuration instructions for each of these components. The Sun Management Center console, server and help server should **not** be any of the cluster nodes, they can, however reside on the cluster's administrative console system. Note the port configured for server-agent communication (the default is 161).

2. Install the Sun Management Center agent component on each of the cluster nodes.

3. Install the Sun Cluster 3.0 modules on the Sun Management Center server, help server and console. The Sun Cluster 3.0 modules for Sun Management Center are:

Package Name	Sun Management Center Component to be installed on
`SUNWscsam`	Agent (Cluster Nodes) - automatically installed by scinstall.
`SUNWscsal`	Agent (Cluster Nodes) - automatically installed by scinstall.
`SUNWscssv`	Server
`SUNWscshl`	Help Server
`SUNWscscn`	Console

Monitoring Sun Cluster 3.0 using Sun Management Center

- Installation

  - Identify and install the Sun Management Center server, help server and console components:

```
# cd <Location of Sun Management Center CD-ROM>
# ./es-inst
```

  - Install the Sun Management Center agent component on each of the cluster nodes:

```
# cd <Location of Sun Management Center CD-ROM>
# ./es-inst
...
Install SyMON Server Component? [y|n|q] n
Install SyMON Agent Component? [y|n|q] y
Install SyMON Console Component? [y|n|q] n
Install SyMON Help Documentation? [y|n|q] n
...
```

  - Install the Sun Cluster 3.0 Sun Management Center modules on the Sun Management Center server, help server and console:

```
On Sun Management Center Server
# cd <Location of Sun Cluster 3.0 CD-ROM>/Packages
# pkgadd -d . SUNWscssv

On Sun Management Center Help Server:
# cd <Location of Sun Cluster 3.0 CD-ROM>/Packages
# pkgadd -d . SUNWscshl

On Sun Management Center Console
# cd <Location of Sun Cluster 3.0 CD-ROM>/Packages
# pkgadd -d . SUNWscscn
```

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

Configuration

The Sun Management Center server must be configured to facilitate the monitoring of the cluster nodes. To configure Sun Management Center to monitor the cluster, perform the following steps:

1. On the Sun Management Center server system, add the following entries to `/etc/group`:

```
esadm::3701:<symon_admin_user_name>
esdomadm::3702::<symon_admin_user_name>
esops::3703:<symon_admin_user_name>
```

2. Start the server processes using the `es-start -A` command

3. On each of the cluster nodes, start the Sun Management Console agent processes using the `es-start -a` command.

4. On the Sun Management Center console, start the Sun Management Center console using the `es-start -c` command

(Continued ...)

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

- Configuration
  - Verify that the configured Sun Management Center user is a member of the esadm, esdomadm and the esops group:

```
esadm::3701:root
esdomadm::3702:root
esops::3703:root
```

  - Start up the Sun Management Center server:

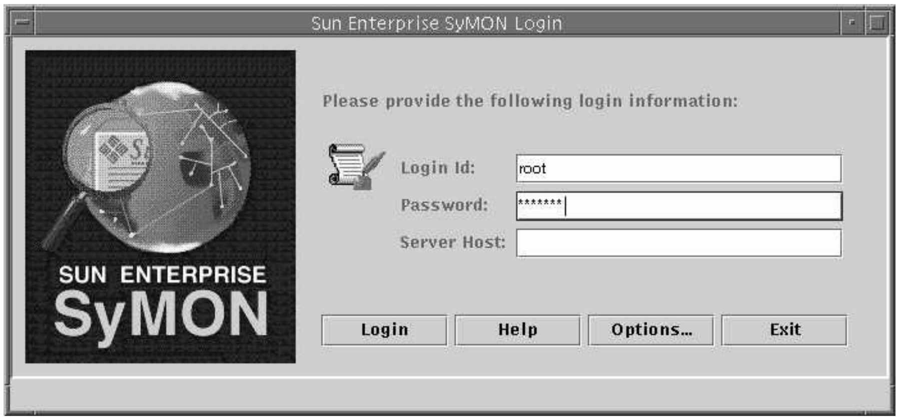
```
On Sun Management Center server system:
# /opt/SUNWsymon/sbin/es-start -A
```

  - Start up the Sun Management Center agents on the cluster nodes:

```
On each of the cluster nodes:
# /opt/SUNWsymon/sbin/es-start -a
```

  - Start up the Sun Management Center console:

```
On the Sun Management Center Console:
# /opt/SUNWsymon/sbin/es-start -c
```



Sun Enterprise SyMON Login

Please provide the following login information:

Login Id: root

Password: *******

Server Host:

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

5. From the Sun Management Center console, configure each of the cluster nodes:

   a. Highlight the default domain and select Create an Object ... from the Edit menu

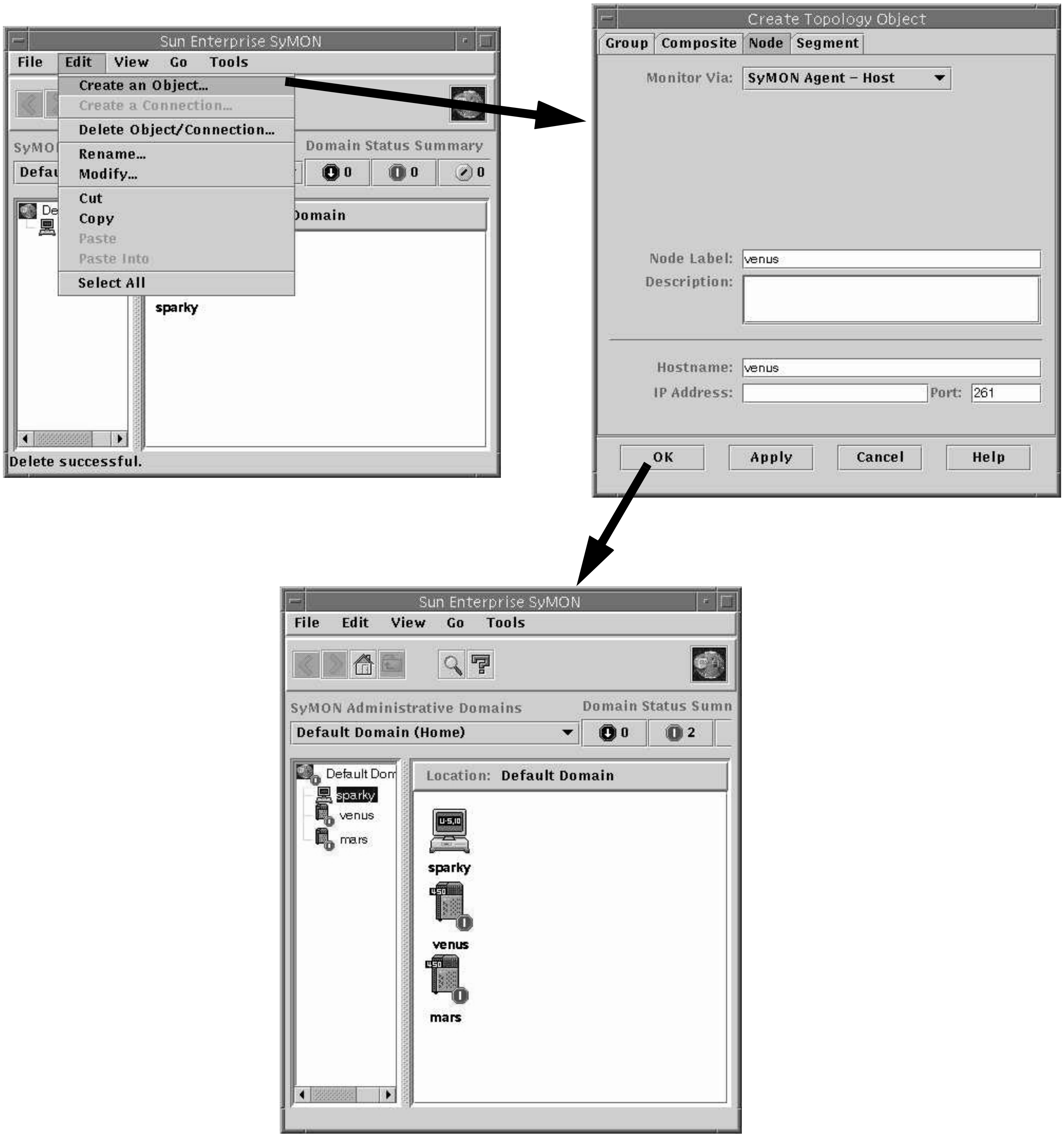
   b. In the Create Topology dialog box, choose the Node tab

   c. From the “Monitor via: “drop-down selection list at the top of the dialog box, select “SyMON Agent - Host”

   d. Use the physical node name as the Node Label and Hostname. Leave the IP field blank and make sure to fill in the proper port number (as configured during Sun Management Center installation)

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

- Configure each cluster node:



Delete successful.

Monitor Via: SyMON Agent – Host

Node Label: venus

Description:

Hostname: venus

IP Address:

Port: 261

Default Domain (Home)

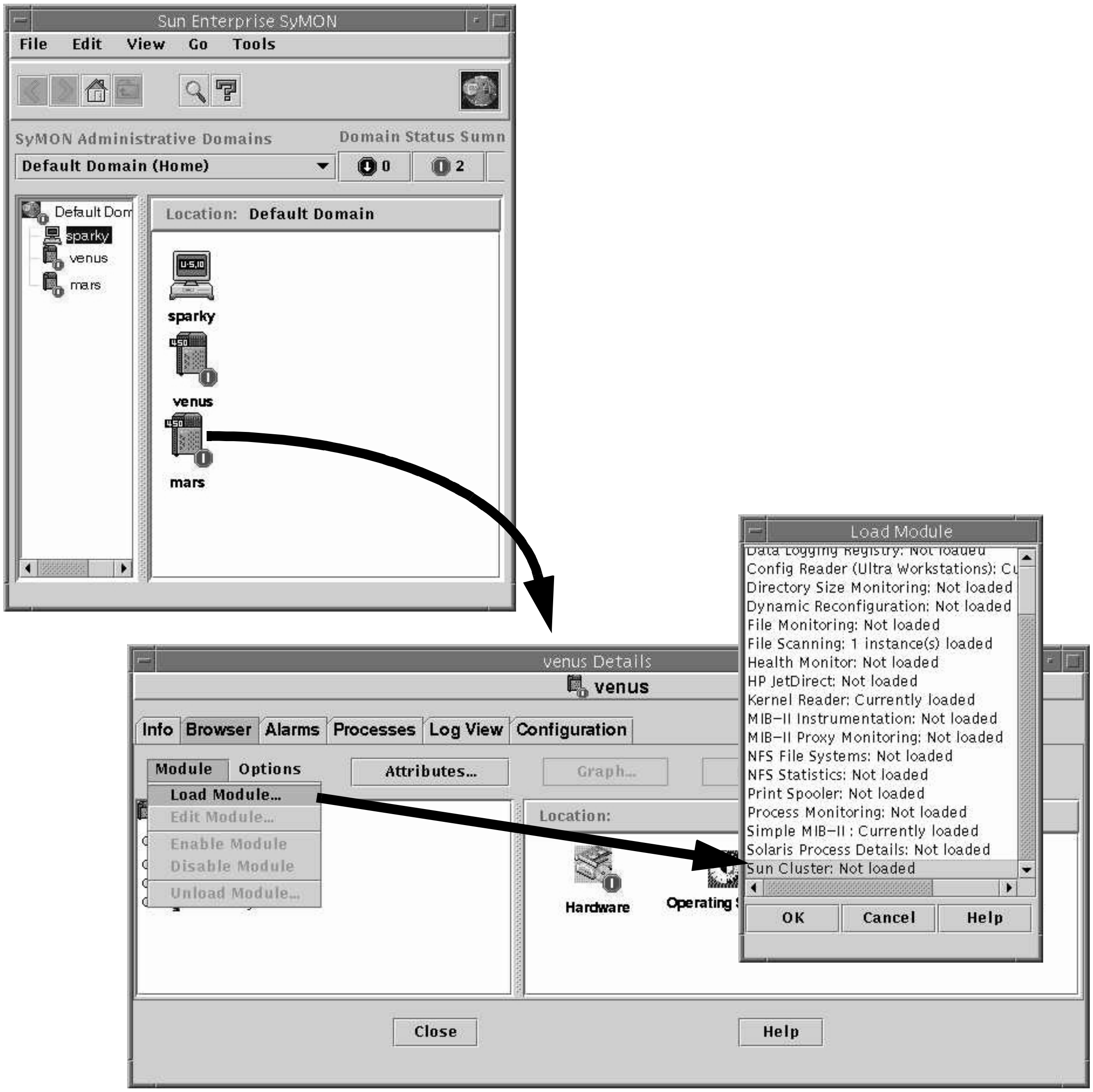
Location: Default Domain

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

e. Double-click on the newly created icon for the node to bring up the detail window for the node. Choose the “Load Module ...” item from the “Module” menu. From presented list, choose the “Sun Cluster” module

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

➤ Load the Sun Cluster module for each node:



Data Logging Registry: Not loaded
Config Reader (Ultra Workstations): Cu
Directory Size Monitoring: Not loaded
Dynamic Reconfiguration: Not loaded
File Monitoring: Not loaded
File Scanning: 1 instance(s) loaded
Health Monitor: Not loaded
HP JetDirect: Not loaded
Kernel Reader: Currently loaded
MIB-II Instrumentation: Not loaded
MIB-II Proxy Monitoring: Not loaded
NFS File Systems: Not loaded
NFS Statistics: Not loaded
Print Spooler: Not loaded
Process Monitoring: Not loaded
Simple MIB-II : Currently loaded
Solaris Process Details: Not loaded
Sun Cluster: Not loaded

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

Using Sun Management Center with Sun Cluster 3.0

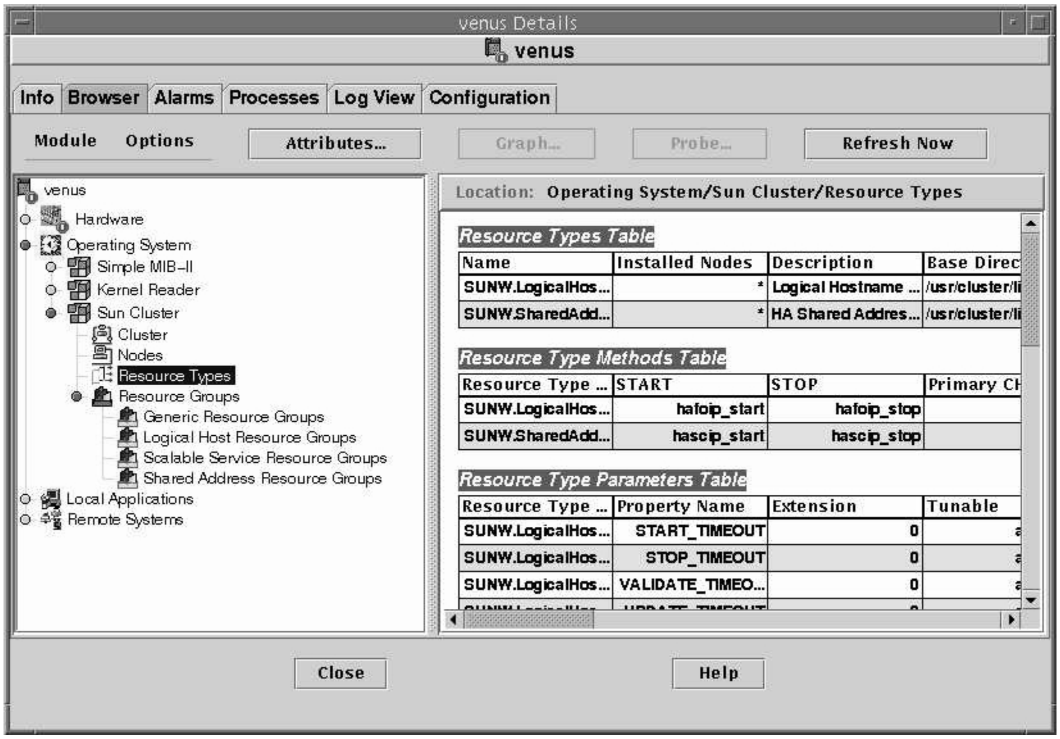
Once the Sun Cluster module has been loaded (this only needs to be done once, the next time you start the Sun Management Center console, it should automatically load itself), the Sun Cluster attributes are under the "Operating System" attribute of the node. The following cluster information can be monitored:

- ❏ Cluster status
- ❏ Node status
- ❏ Registered resource types and properties
- ❏ Resource group status and properties

By clicking on any of the cluster attributes, the appropriate status and property information will be displayed in a tabular format on the right hand side of the node's detail window.

Monitoring Sun Cluster 3.0 using Sun Management Center (Continued)

- Using Sun Management Center with Sun Cluster 3.0



Location: Operating System/Sun Cluster/Resource Types

Resource Types Table

Name	Installed Nodes	Description	Base Direc
SUNW.LogicalHos...	*	Logical Hostname ...	/usr/cluster/li
SUNW.SharedAdd...	*	HA Shared Addres...	/usr/cluster/li

Resource Type Methods Table

Resource Type ...	START	STOP	Primary Cl
SUNW.LogicalHos...	hafoip_start	hafoip_stop	
SUNW.SharedAdd...	hascip_start	hascip_stop	

Resource Type Parameters Table

Resource Type ...	Property Name	Extension	Tunable
SUNW.LogicalHos...	START_TIMEOUT	0	a
SUNW.LogicalHos...	STOP_TIMEOUT	0	a
SUNW.LogicalHos...	VALIDATE_TIMEO...	0	a

APPENDIX A

Configuration Worksheets

Configuration Worksheets

This section provides the following planning worksheets:

- Cluster and Node Names Worksheet
- Cluster Interconnect Worksheet
- Public Networks Worksheet
- Local Devices Worksheet
- Local File System Layout Worksheet
- Disk Device Group Configurations Worksheet
- Volume Manager Configurations Worksheet
- Metadevices Worksheet

You might need to make multiple copies of a worksheet to accommodate all the components in your cluster configuration.

Cluster and Node Names Worksheet

Cluster name ______________________________________

Private network IP address ____________________________ (default: `172.16.0.0`)

Private network mask ________________________________ (default: `255.255.0.0`)

Nodes

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Cluster Interconnect Worksheet

Adapters **Cabling** **Junctions**

Draw lines between cable endpoints

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Junction name ___________________
Junction type ___________________

Port Number	Description (optional)

Junction name ___________________
Junction type ___________________

Port Number	Description (optional)

Public Networks Worksheet

Node name ___________________________

Primary hostname _______________________

Network name __________________________

Adapter names _________________________

NAFO group number: `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number : `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number : `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number : `nafo`____

Node name ___________________________

Primary hostname _______________________

Network name __________________________

Adapter names _________________________

NAFO group number: `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number: `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number: `nafo`____

Secondary hostname_____________________

Network name __________________________

Adapter names _________________________

NAFO group number: `nafo`____

Local Devices Worksheet

Node name ______________________________

Local disks

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Other local devices

Device type _________ Name _____________ Device type _________ Name _____________

Device type _________ Name _____________ Device type _________ Name _____________

Node name ______________________________

Local disks

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Other local devices

Device type _________ Name _____________ Device type _________ Name _____________

Device type _________ Name _____________ Device type _________ Name _____________

Local File System Layout Worksheet

Node name ______________________________

Mirrored root

Volume Name	Component	Component	File System	Size
			/	
			/usr	
			/var	
			/opt	
			swap	
			/globaldevices	

Non-mirrored root

Device Name	File System	Size
	/	
	/usr	
	/var	
	/opt	
	swap	
	/globaldevices	

Disk Device Group Configurations Worksheet

Volume manager: ___________________________

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Volume Manager Configurations Worksheet

Volume manager: ___________________________

Name	Type	Component	Component

Metadevices Worksheet

		Metamirrors		Submirrors			Physical Device	
File System	Metatrans	(Data)	(Log)	(Data)	(Log)	Hot Spare Pool	(Data)	(Log)

Configuration Worksheet Examples

Note – The data used in these examples is intended as a guideline only, and does not represent a complete configuration of a functional cluster.

- Example: Cluster and Node Names
- Example: Cluster Interconnect
- Example: Public Networks
- Example: Local Devices
- Example: Local File System Layout—With Mirrored Root
- Example: Local File System Layout—Without Mirrored Root
- Example: Disk Device Group Configurations
- Example: Volume Manager Configurations
- Example: Metadevices

Example: Cluster and Node Names

Cluster name **sccluster**

Private network IP address ____________ (default: `172.16.0.0`)

Private network mask ____________ (default: `255.255.0.0`)

Nodes

Node name **phys-schost-1**

Private hostname **phys-schost-1-priv**

Node name **phys-schost-2**

Private hostname **phys-schost-2-priv**

Node name ____________

Private hostname ____________

Node name ____________

Private hostname ____________

Node name ____________

Private hostname ____________

Node name ____________

Private hostname ____________

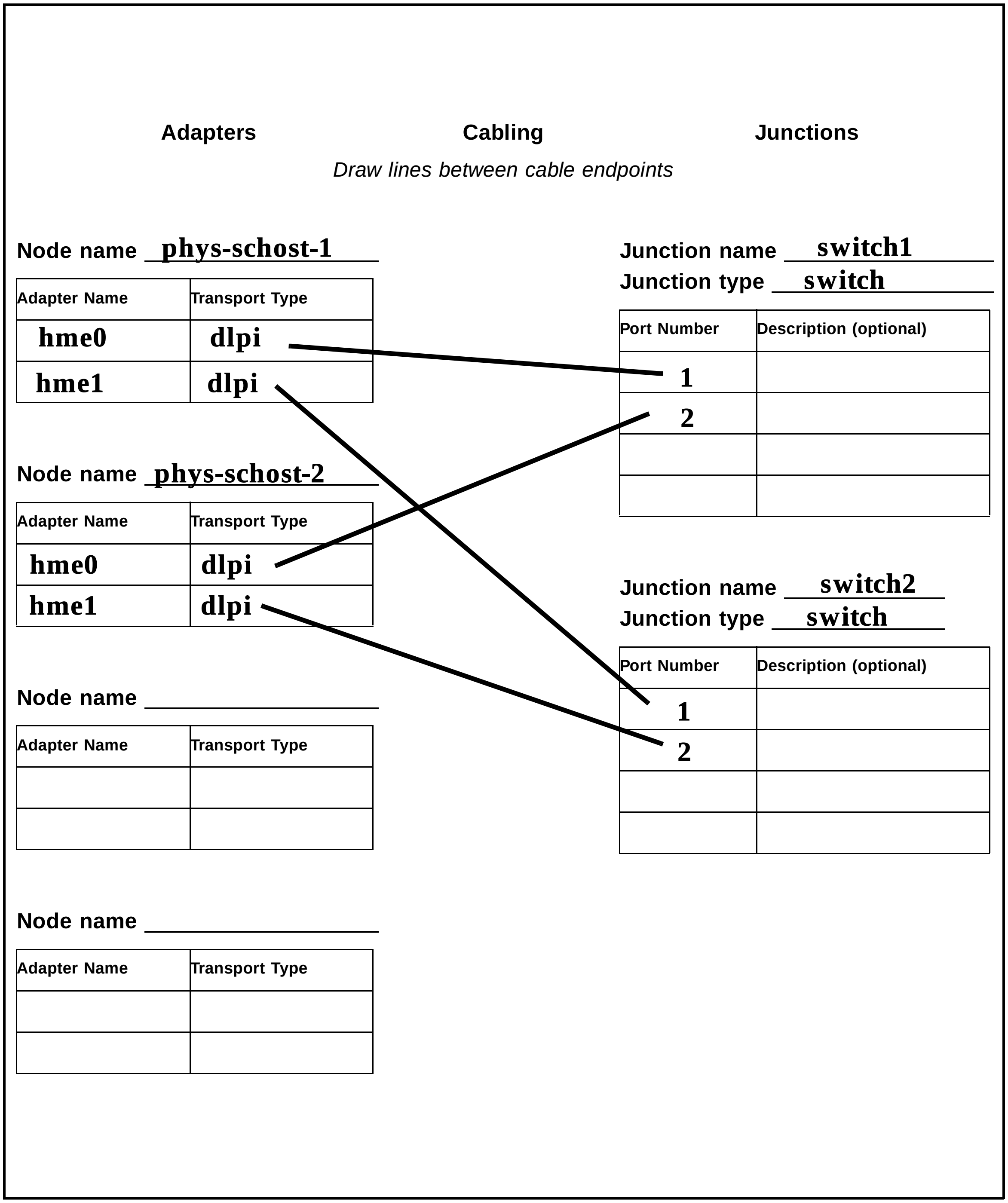
Node name ____________

Private hostname ____________

Node name ____________

Private hostname ____________

Example: Cluster Interconnect



Adapters **Cabling** **Junctions**

Draw lines between cable endpoints

Node name phys-schost-1

Adapter Name	Transport Type
hme0	dlpi
hme1	dlpi

Node name phys-schost-2

Adapter Name	Transport Type
hme0	dlpi
hme1	dlpi

Node name ____________________

Adapter Name	Transport Type

Node name ____________________

Adapter Name	Transport Type

Junction name switch1

Junction type switch

Port Number	Description (optional)
1	
2	

Junction name switch2

Junction type switch

Port Number	Description (optional)
1	
2	

Example: Public Networks

Node name phys-schost-1

Primary hostname phys-schost-1

Network name net-85

Adapter names hme0

NAFO group number: nafo 0

Secondary hostname phys-schost-1-86

Network name net-86

Adapter names hme3

NAFO group number : nafo 1

Secondary hostname____________________

Network name __________________________

Adapter names _________________________

NAFO group number : nafo____

Secondary hostname____________________

Network name __________________________

Adapter names _________________________

NAFO group number : nafo____

Node name phys-schost-2

Primary hostname phys-schost-2

Network name net-85

Adapter names hme0

NAFO group number: nafo 0

Secondary hostname phys-schost-2-86

Network name net-86

Adapter names hme3

NAFO group number: nafo 1

Secondary hostname____________________

Network name __________________________

Adapter names _________________________

NAFO group number: nafo____

Secondary hostname____________________

Network name __________________________

Adapter names _________________________

NAFO group number: nafo____

Example: Local Devices

Node name phys-schost-1

Local disks

Disk name c0t0d0 Size 2G Disk name ____________ Size ______

Disk name c0t1d0 Size 2G Disk name ____________ Size ______

Disk name c1t0d0 Size 2G Disk name ____________ Size ______

Disk name c1t1d0 Size 2G Disk name ____________ Size ______

Other local devices

Device type tape Name /dev/rmt/0 Device type ________ Name __________

Device type ________ Name __________ Device type ________ Name __________

Node name ____________________

Local disks

Disk name ____________ Size ______ Disk name ____________ Size ______

Disk name ____________ Size ______ Disk name ____________ Size ______

Disk name ____________ Size ______ Disk name ____________ Size ______

Disk name ____________ Size ______ Disk name ____________ Size ______

Other local devices

Device type ________ Name __________ Device type ________ Name __________

Device type ________ Name __________ Device type ________ Name __________

Example: Local File System Layout—With Mirrored Root

Node name _____phys-schost-1__________

Mirrored root

Volume Name	Component	Component	File System	Size
d1	c0t0d0s0	c1t0d0s0	/	200MB
d2	c0t0d0s1	c1t0d0s1	/var	200MB
d3	c0t0d0s3	c1t0d0s3	swap	750MB
d4	c0t0d0s4	c1t0d0s4	/globaldevices	100MB
d5	c0t0d0s5	c1t0d0s5	/opt	400MB
d6	c0t0d0s6	c1t0d0s6	/usr	900MB
d7	c0t0d0s7	c1t0d0s7	SDS replicas	10MB

Non-mirrored root

Device Name	File System	Size
	/	
	/usr	
	/var	
	/opt	
	swap	
	/globaldevices	

Example: Local File System Layout—Without Mirrored

Root

Node name phys-schost-1

Mirrored root

Volume Name	Component	Component	File System	Size
			/	
			/var	
			swap	
			/globaldevices	
			/opt	
			/usr	

Non-mirrored root

Device Name	File System	Size
c0t0d0s0	/	200MB
c0t0d0s1	/var	200MB
c0t0d0s3	swap	750MB
c0t0d0s4	/globaldevices	100MB
c0t0d0s5	/opt	400MB
c0t0d0s6	/usr	900MB
c0t0d0s7		10MB

Example: Disk Device Group Configurations

Volume manager: Solstice DiskSuite

Disk group/diskset name relo-sccluster

Node names (1) phys-schost-1 (2) phys-schost-2 (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ☒ Yes ❑ No

Maximum number of secondaries 0

Failback? ☒ Yes ❑ No

Disk group/diskset name ____________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❑ Yes ❑ No

Maximum number of secondaries ________

Failback? ❑ Yes ❑ No

Disk group/diskset name ____________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❑ Yes ❑ No

Maximum number of secondaries ________

Failback? ❑ Yes ❑ No

Example: Volume Manager Configurations

Volume manager: Solstice DiskSuite

Name	Type	Component	Component
schost-1/d0	trans	schost-1/d1	schost-1/d4
schost-1/d1	mirror	c0t0d0s4	c4t4d0s4
schost-1/d4	mirror	c0t0d2s5	c4t4d2s5

Example: Metadevices

		Metamirrors		Submirrors			Physical Device	
File System	Metatrans	(Data)	(Log)	(Data)	(Log)	Hot Spare Pool	(Data)	(Log)
A	d10	d11		d12		hsp000	c1t0d0s0	
				d13		hsp000	c2t0d1s0	
			d14		d15	hsp006		c1t0d1s6
					d16	hsp006		c2t1d0s6

APPENDIX B

Configuration GUI

Configuration GUI

Introduction

This section describes how to use the Resource Group Manager configuration GUI to configure the following data services:

- NFS - failover
- Apache Web Server - scalable

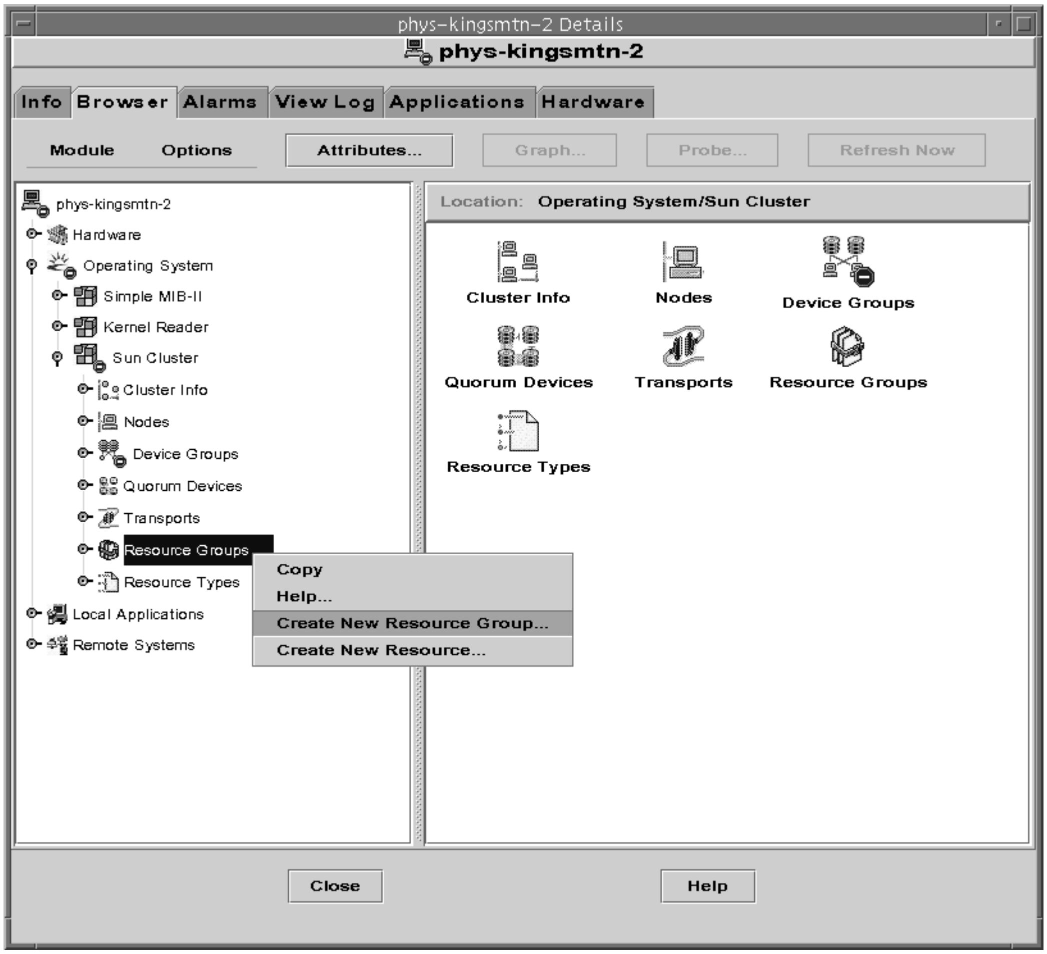
The full procedures for using command-line tools to accomplish these tasks are described in Chapter 6, as are descriptions of the properties being configured. This appendix uses cluster node names `phys-kingsmtn-1` and `phys-kingsmtn-2` rather than `mars` and `venus`, which are used in the Chapter 6 examples.

Before Starting

The Sun Cluster 3.0 Resource Group Manager configuration GUI runs under Sun Management Center 2.1. This document assumes that the Sun Management Center server and console are already configured and running,that Sun Management Center agents are configured, and the Sun Cluster module is loaded and running on those cluster nodes which will be used as viewports onto the cluster.

▼ Invoking the Configuration GUI

1. **From the Sun Management Center console double-click the node to be used as a viewport onto the cluster.**
2. **From the cluster node Details window, double click** Operating System**, then** Sun Cluster, **then** Resource Groups **to expand these tree-view nodes.**
3. **Right-click on** Resource Groups **in the tree-view to pop up the menu from which the** Create New Resource Group **and** Create New Resource **configuration GUIs may be launched.**



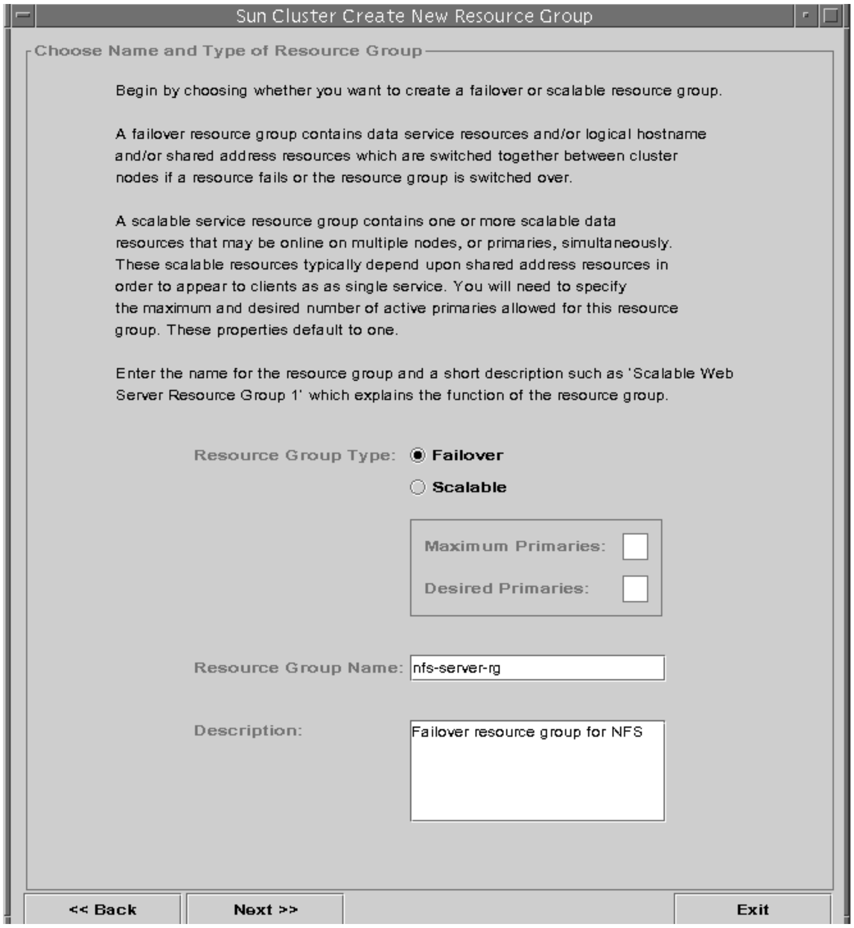
NFS - a Failover Data Service

▼ Overview:

1. **Install and prepare the NFS application as described in Chapter 6.**
2. **Follow the directions in Chapter 6 to register the resource type** SUNW.nfs, **if it is not already registered.**
3. **Use the configuration GUI to create a single resource group named** nfs-server-rg.
4. **Use the configuration GUI to create a logical hostname resource named** nfs-server-lhrs.
5. **Use the configuration GUI to create a NFS resource named** nfs-server-res**.**
6. **Follow the directions in Chapter 6 for using `scswitch(1M)` to manage the resource group and bring it online**

▼ Create the Resource Group

1. **Invoke the** Create New Resource Group **GUI.**
2. **Click** Next **to move past the introductory screen.**
3. **Leave** Failover **checked; enter** nfs-server-rg **as the resource group name; enter a description.**



Sun Cluster Create New Resource Group

Choose Name and Type of Resource Group

Begin by choosing whether you want to create a failover or scalable resource group.

A failover resource group contains data service resources and/or logical hostname and/or shared address resources which are switched together between cluster nodes if a resource fails or the resource group is switched over.

A scalable service resource group contains one or more scalable data resources that may be online on multiple nodes, or primaries, simultaneously. These scalable resources typically depend upon shared address resources in order to appear to clients as as single service. You will need to specify the maximum and desired number of active primaries allowed for this resource group. These properties default to one.

Enter the name for the resource group and a short description such as 'Scalable Web Server Resource Group 1' which explains the function of the resource group.

Resource Group Type: ● Failover ○ Scalable

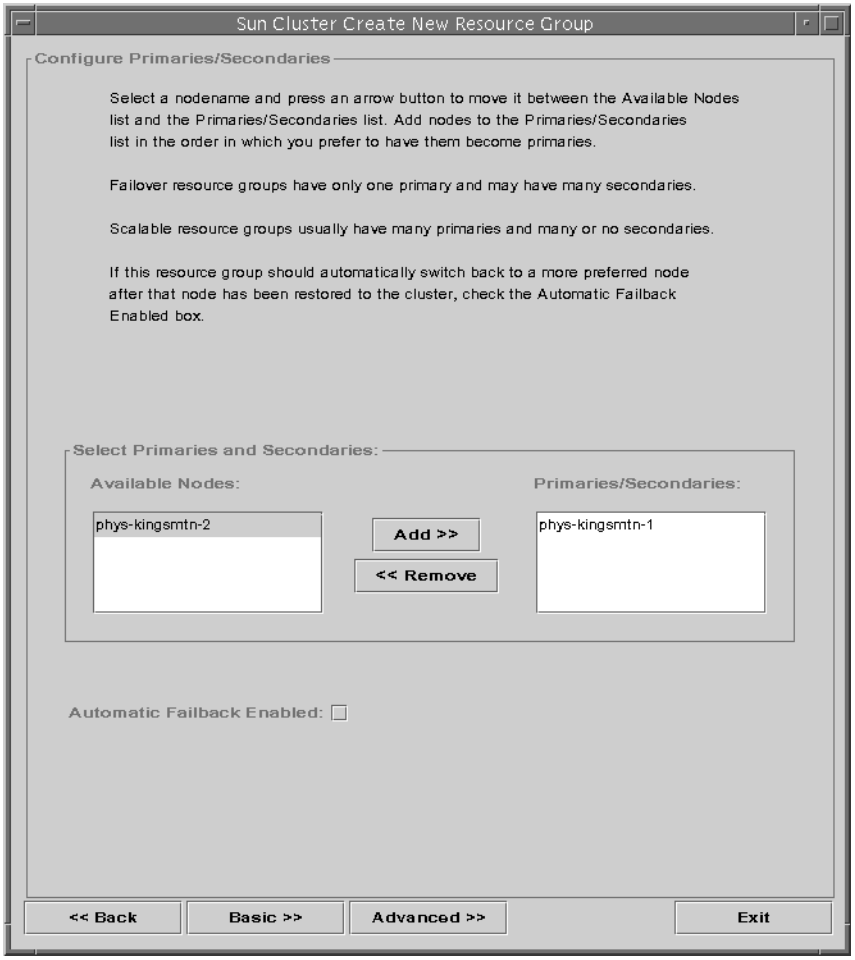
Maximum Primaries: □

Desired Primaries: □

Resource Group Name: nfs-server-rg

Description: Failover resource group for NFS

4. **Click** Next **to move ahead to the** Configure Primaries/Secondaries **screen.**
5. **In the left-hand list,** Available Nodes**, double-click on each node that may host this resource group.**



Sun Cluster Create New Resource Group

Configure Primaries/Secondaries

Select a nodename and press an arrow button to move it between the Available Nodes list and the Primaries/Secondaries list. Add nodes to the Primaries/Secondaries list in the order in which you prefer to have them become primaries.

Failover resource groups have only one primary and may have many secondaries.

Scalable resource groups usually have many primaries and many or no secondaries.

If this resource group should automatically switch back to a more preferred node after that node has been restored to the cluster, check the Automatic Failback Enabled box.

Select Primaries and Secondaries:

Available Nodes: phys-kingsmtn-2

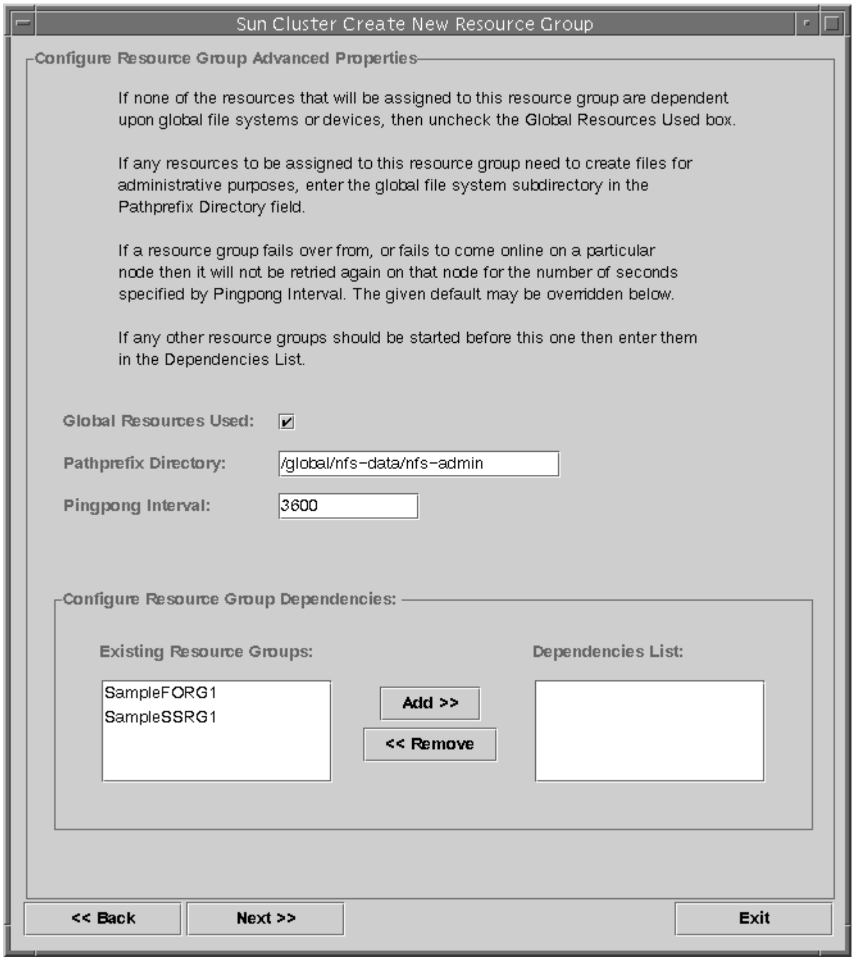
Primaries/Secondaries: phys-kingsmtn-1

Automatic Failback Enabled: □

6. **The** NFS **resource that will later be placed into this resource group will need to refer to the resource group's** PathPrefix **property, so click** Advanced **to move ahead to the** Advanced Properties **screen.**

7. **In the** Configure Resource Group Advanced Properties **screen, enter** /global/nfs-data/nfs-admin **in the** PathPrefix Directory **textfield.**

   This resource group will not have any dependencies on any other resource groups.



Sun Cluster Create New Resource Group

Configure Resource Group Advanced Properties

If none of the resources that will be assigned to this resource group are dependent upon global file systems or devices, then uncheck the Global Resources Used box.

If any resources to be assigned to this resource group need to create files for administrative purposes, enter the global file system subdirectory in the Pathprefix Directory field.

If a resource group fails over from, or fails to come online on a particular node then it will not be retried again on that node for the number of seconds specified by Pingpong Interval. The given default may be overridden below.

If any other resource groups should be started before this one then enter them in the Dependencies List.

Global Resources Used: ☑

Pathprefix Directory: /global/nfs-data/nfs-admin

Pingpong Interval: 3600

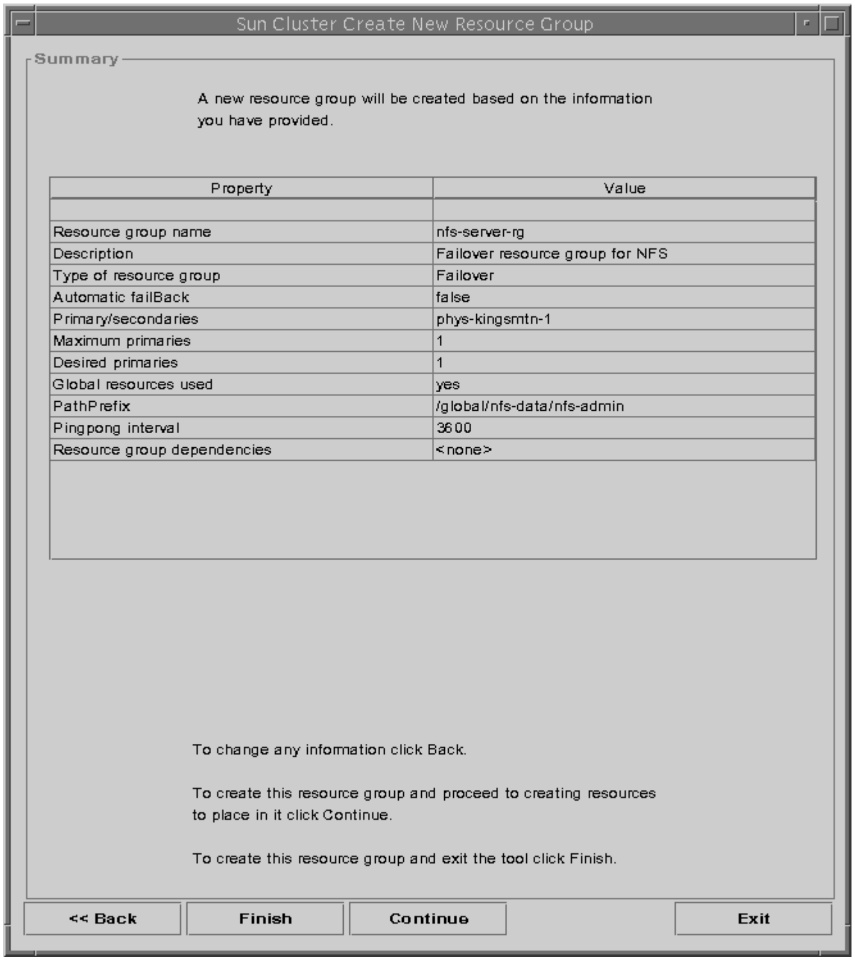
Configure Resource Group Dependencies:

Existing Resource Groups:

SampleFORG1
SampleSSRG1

Dependencies List:

8. **Click** Next **to move ahead to the** Summary **screen.**



A new resource group will be created based on the information you have provided.

Property	Value
Resource group name	nfs-server-rg
Description	Failover resource group for NFS
Type of resource group	Failover
Automatic failBack	false
Primary/secondaries	phys-kingsmtn-1
Maximum primaries	1
Desired primaries	1
Global resources used	yes
PathPrefix	/global/nfs-data/nfs-admin
Pingpong interval	3600
Resource group dependencies	<none>

To change any information click Back.

To create this resource group and proceed to creating resources to place in it click Continue.

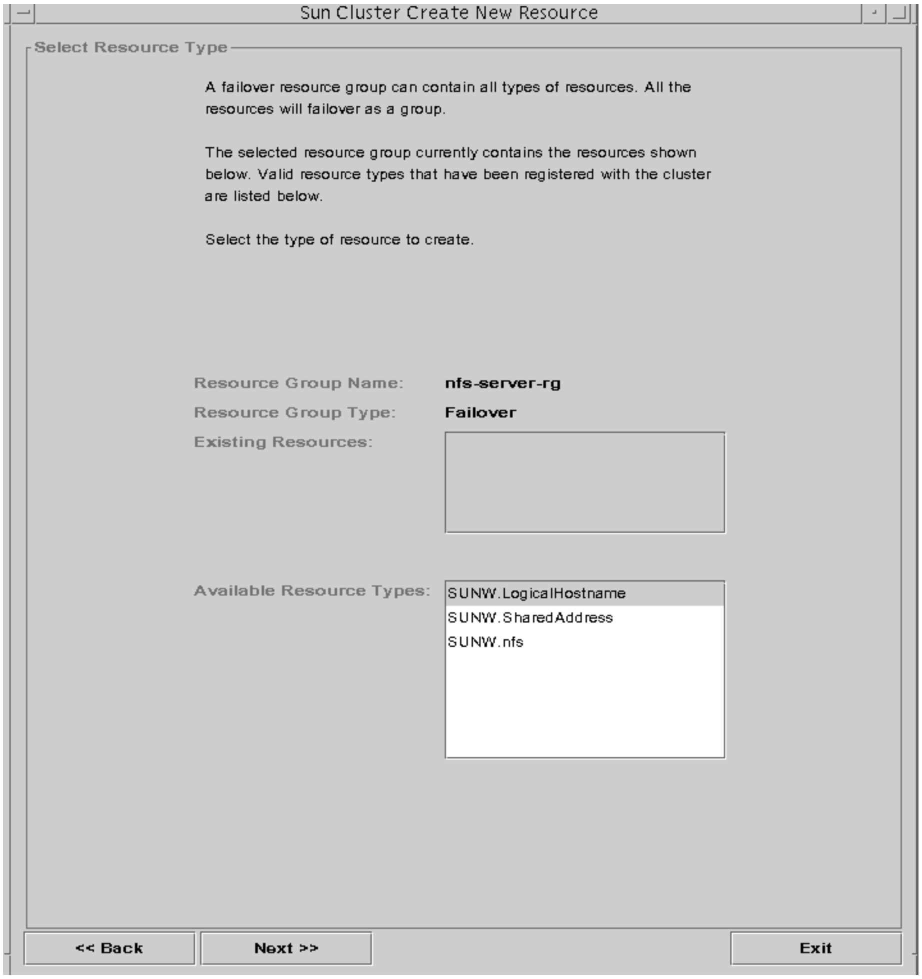
To create this resource group and exit the tool click Finish.

Review the Property/Value pairs. Let the mouse pointer linger over a long value to have that value appear as a tooltip.

9. **Click** Continue **to create this resource group on the cluster and automatically move ahead to the** Create New Resource **configuration GUI, with this new resource group already selected.**

▼ Create the Logical Hostname Resource

1. **Select** SUNW.LogicalHostname **as the resource type to instantiate as a resource.**



Sun Cluster Create New Resource

Select Resource Type

A failover resource group can contain all types of resources. All the resources will failover as a group.

The selected resource group currently contains the resources shown below. Valid resource types that have been registered with the cluster are listed below.

Select the type of resource to create.

Resource Group Name: **nfs-server-rg**

Resource Group Type: **Failover**

Existing Resources:

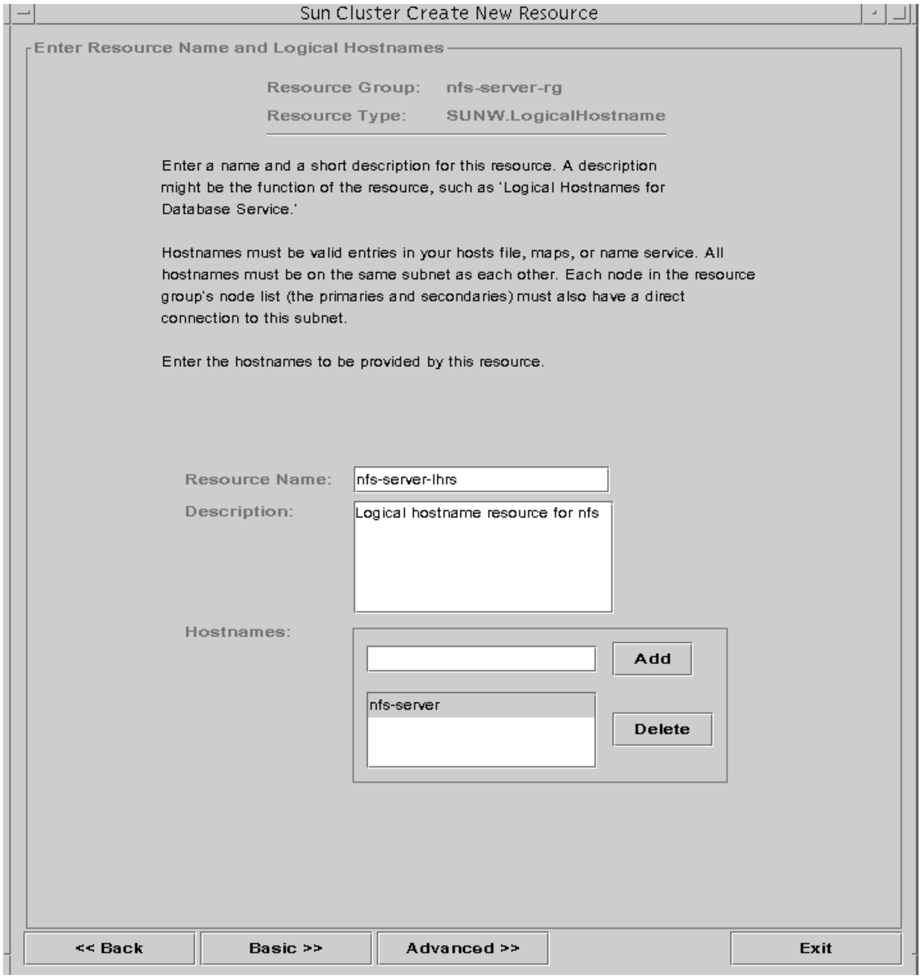
Available Resource Types:

SUNW.LogicalHostname
SUNW.SharedAddress
SUNW.nfs

2. **Click** Next **to move ahead to the naming screen.**

3. **Enter** nfs-server-lhrs **for the name of the logical hostname resource and enter a short description of the resource; enter** nfs-server **as the hostname to be provided by this resource.**

   Note that this hostname must be on the same public subnet as the cluster nodes that may host the resource group. This hostname must be a valid entry in the site host files or maps or name service. A logical hostname or shared address resource may contain more than one hostname.



Sun Cluster Create New Resource

Enter Resource Name and Logical Hostnames

Resource Group: nfs-server-rg

Resource Type: SUNW.LogicalHostname

Enter a name and a short description for this resource. A description might be the function of the resource, such as 'Logical Hostnames for Database Service.'

Hostnames must be valid entries in your hosts file, maps, or name service. All hostnames must be on the same subnet as each other. Each node in the resource group's node list (the primaries and secondaries) must also have a direct connection to this subnet.

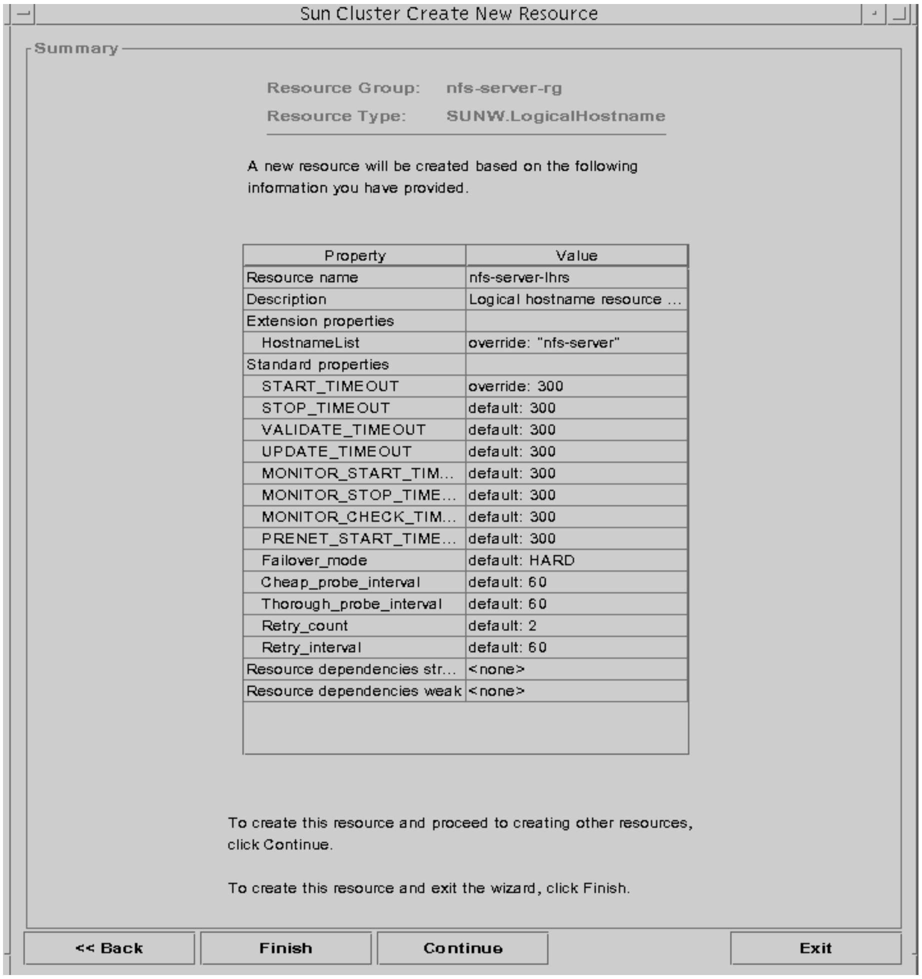
Enter the hostnames to be provided by this resource.

Resource Name: nfs-server-lhrs

Description: Logical hostname resource for nfs

Hostnames: nfs-server

4. **There is no need to adjust any advanced properties on this resource, so click** Basic **to move ahead to the** Summary **screen.**



Resource Group: nfs-server-rg

Resource Type: SUNW.LogicalHostname

A new resource will be created based on the following information you have provided.

Property	Value
Resource name	nfs-server-lhrs
Description	Logical hostname resource ...
Extension properties	
HostnameList	override: "nfs-server"
Standard properties	
START_TIMEOUT	override: 300
STOP_TIMEOUT	default: 300
VALIDATE_TIMEOUT	default: 300
UPDATE_TIMEOUT	default: 300
MONITOR_START_TIM...	default: 300
MONITOR_STOP_TIME...	default: 300
MONITOR_CHECK_TIM...	default: 300
PRENET_START_TIME...	default: 300
Failover_mode	default: HARD
Cheap_probe_interval	default: 60
Thorough_probe_interval	default: 60
Retry_count	default: 2
Retry_interval	default: 60
Resource dependencies str...	<none>
Resource dependencies weak	<none>

To create this resource and proceed to creating other resources, click Continue.

To create this resource and exit the wizard, click Finish.

5. **Click** Continue **in order to create this resource and continue creating more resources for this same resource group.**

▼ Create the NFS Resource

1. **Select** SUNW.nfs **as the resource type to instantiate. Click** Next**.**
2. **Enter** nfs-server-res **as the resource name and a enter description**; **click** Next**.**
3. **There are no adjustments required or recommended to any of the extension or standard properties, so click** Basic **to move directly to the** Summary **screen.**
4. **Click** Finish **to create this resource and exit the configuration GUI.**
5. **Refer to Chapter 6 for appropriate** `scswitch(1M)` **commands to manage and bring this resource group online.**

Apache Web Server- a Scalable Data Service

▼ Overview:

1. **Install and prepare the** apache **application as described in Chapter 6.**
2. **Follow the directions in Chapter 6 to register the resource type** SUNW.apache, **if it is not already registered.**
3. **Use the configuration GUI to create a failover resource group named** ap-server-sa **to hold the shared address resource**.
4. **Use the configuration GUI to create a shared address resource named** ap-server-sars**.**
5. **Use the configuration GUI to create a scalable resource group named** apache-sa **to hold the web server resource**.
6. **Use the configuration GUI to create a data service resource, an instance of the Apache Web Server, called** apache-res**.**
7. **Follow the directions in Chapter 6 for using `scswitch(1M)` to manage the resource groups and bring them online.**

▼ Create a Failover Resource Group for the Shared Address

1. **Invoke the** Create New Resource Group **GUI.**
2. **Click** Next **to move past the introductory screen.**
3. **Leave** Failover **checked; enter** ap-server-sa **as resource group name; enter a description.**
4. **Click** Next **to move ahead to the** Configure Primaries/Secondaries **screen.**
5. **In the left-hand list,** Available Nodes**, double-click on each node that may host this resource group.**
6. **Click** Basic **to move ahead directly to the** Summary **screen.**

   Review the Property/Value pairs. Let the mouse pointer linger over a long value to have that value appear as a tooltip.
7. **Click** Continue **to create this resource group on the cluster and automatically move ahead to the** Create New Resource **configuration GUI, with this new resource group already selected.**

▼ Create the Shared Address Resource

1. **Select** SUNW.SharedAddress **as the resource type to instantiate as a resource.**
2. **Click** Next **to move ahead to the naming screen.**
3. **Enter** ap-server-sars **for the name of the logical hostname resource and enter a short description of the resource.**
4. **Enter** ap-server **as the hostname to be served by this resource.**

   Note that this hostname must be on the same public subnet as the cluster nodes that may host the resource group. This hostname must be a valid entry in the site host files or maps or name service. A logical hostname or shared address resource may serve more than one hostname.

5. **There is no need to adjust any advanced properties on this resource, so click** Basic **to move ahead to the** Summary **screen.**

6. **Click** Finish **in order to create this resource and exit the configuration GUI.**

▼ Create a Scalable Resource Group for the Apache Web Server

1. **Invoke the Create New Resource Group GUI.**

2. **Click** Next **to move past the introductory screen.**

3. **Click** Scalable; **enter** 2 **for each of** Maximum **and** Desired Primaries; **enter** apache-sa **as resource group name; enter a description.**

4. **Click** Next **to move ahead to the** Configure Primaries/Secondaries **screen.**

5. **In the left-hand list,** Available Nodes**, double-click on each node that may host this resource group.**

6. **Click** Advanced **in order to get access to resource group dependencies.**

   In the left-hand list of existing resource groups, double-click `ap-server-sa`, the resource group containing the shared address resource that the web server will use.

7. **Click** Next **in order to move ahead to the** Summary **screen.**

   Review the Property/Value pairs. Let the mouse pointer linger over a long value in order to have that value appear as a tooltip.

8. **Click** Continue **to create this resource group on the cluster and automatically move ahead to the** Create New Resource **configuration GUI, with this new resource group already selected.**

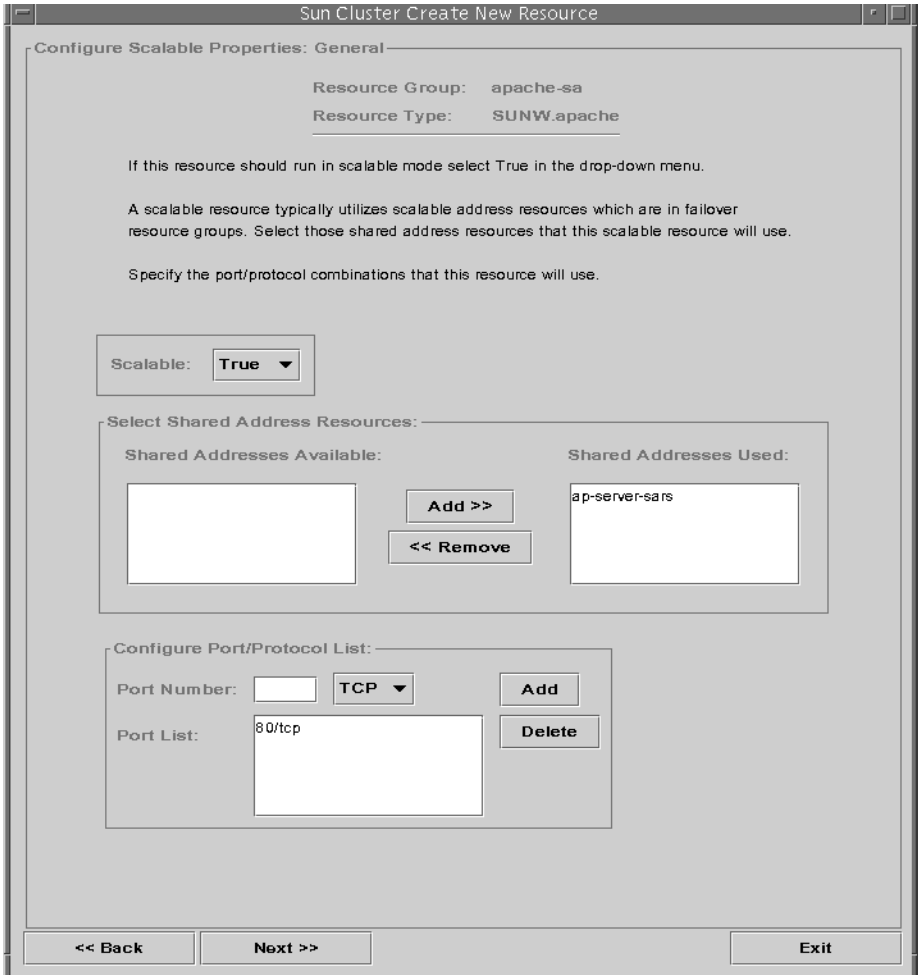
▼ Create the Scalable Apache Web Server Resource

1. **Select** SUNW.apache **as the resource type to instantiate as a resource.**

2. **Click** Next **to move ahead to the naming screen.**

3. **Enter** apache-res **for the name of the resource and enter a short description. Click** Next **to move ahead to the** General Scalable Properties **screen.**

4. **Change the** Scalable **type to** True **using the drop-down choice; select** ap-server-sars **from the list of available shared address resources; leave the** Port/Protocol List **with only the default of** 80/tcp**.**



Sun Cluster Create New Resource

Configure Scalable Properties: General

Resource Group: apache-sa

Resource Type: SUNW.apache

If this resource should run in scalable mode select True in the drop-down menu.

A scalable resource typically utilizes scalable address resources which are in failover resource groups. Select those shared address resources that this scalable resource will use.

Specify the port/protocol combinations that this resource will use.

Scalable: True

Select Shared Address Resources:

Shared Addresses Available:

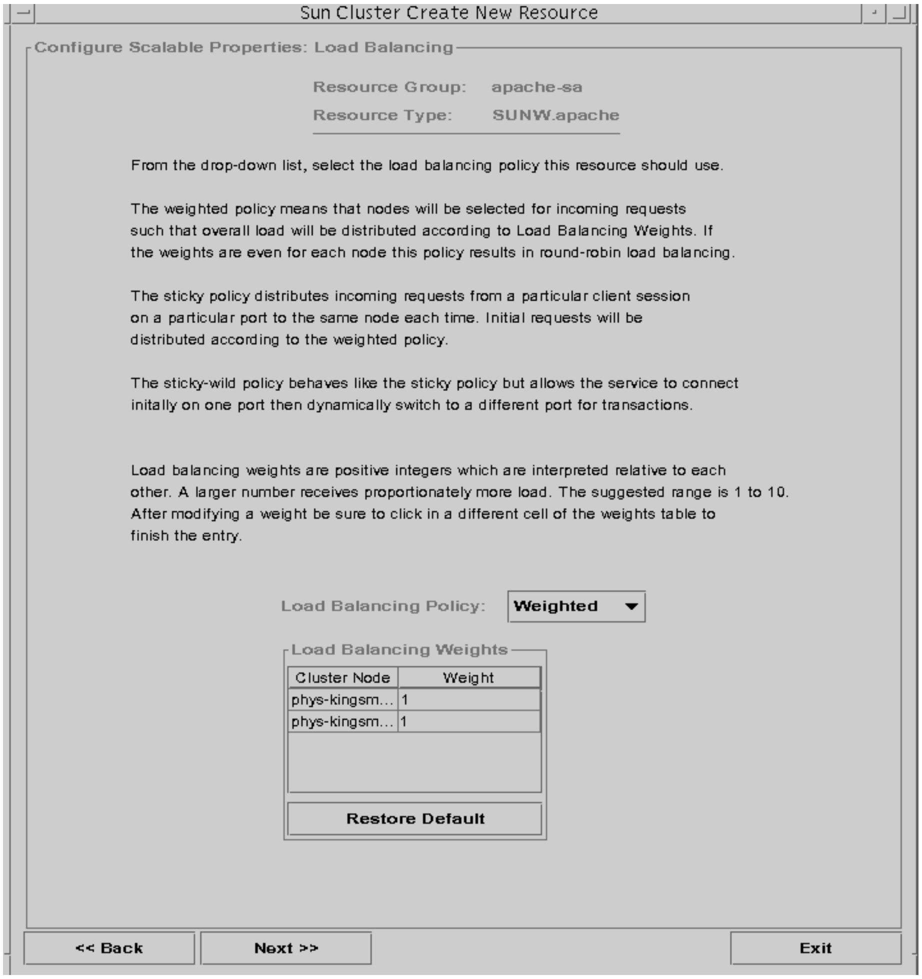
Shared Addresses Used: ap-server-sars

Configure Port/Protocol List:

Port Number: TCP

Port List: 80/tcp

5. **Click** Next **to move ahead to the** load balancing properties **screen. Leave the** policy **as** Weighted; **enter** 1 **in the** Weight **column for each node (after entering the last value, be sure to click in a different table cell to finish the entry).**



Sun Cluster Create New Resource

Configure Scalable Properties: Load Balancing

Resource Group: apache-sa
Resource Type: SUNW.apache

From the drop-down list, select the load balancing policy this resource should use.

The weighted policy means that nodes will be selected for incoming requests such that overall load will be distributed according to Load Balancing Weights. If the weights are even for each node this policy results in round-robin load balancing.

The sticky policy distributes incoming requests from a particular client session on a particular port to the same node each time. Initial requests will be distributed according to the weighted policy.

The sticky-wild policy behaves like the sticky policy but allows the service to connect initally on one port then dynamically switch to a different port for transactions.

Load balancing weights are positive integers which are interpreted relative to each other. A larger number receives proportionately more load. The suggested range is 1 to 10. After modifying a weight be sure to click in a different cell of the weights table to finish the entry.

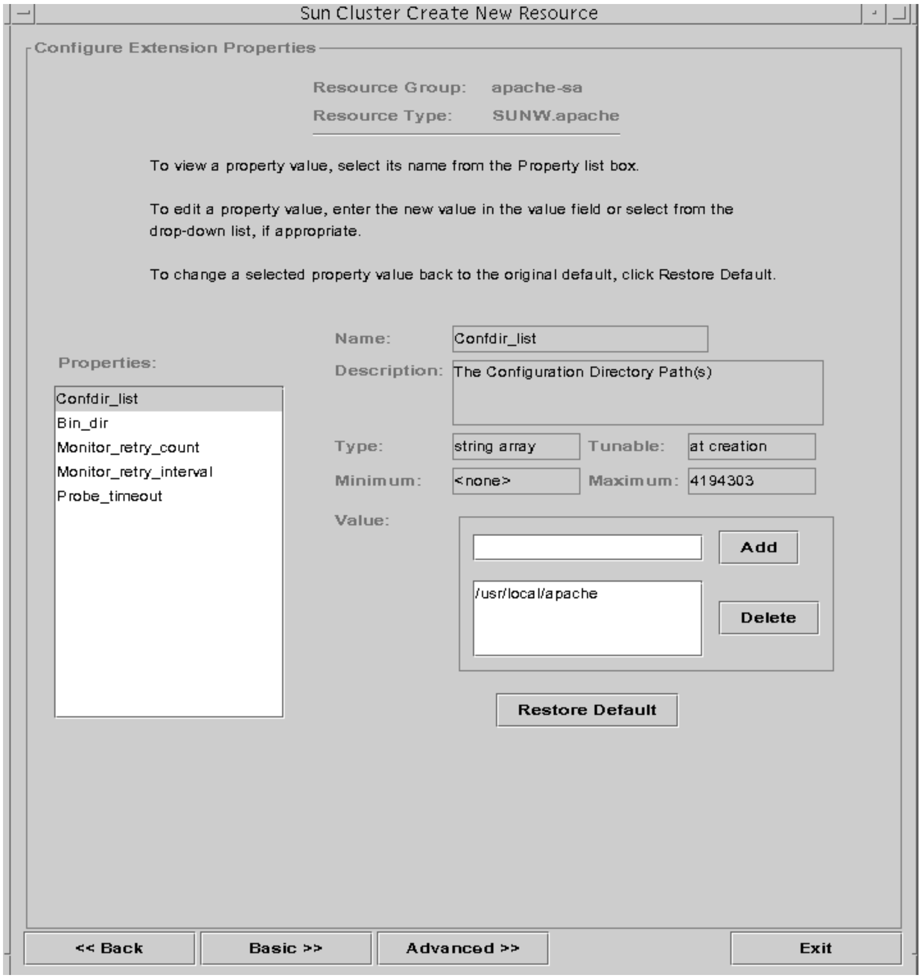
Load Balancing Policy: Weighted

Load Balancing Weights

Cluster Node	Weight
phys-kingsm...	1
phys-kingsm...	1

6. **Click** Next **to move ahead to the** Extension Properties **screen; set the following properties:**

Property	Value
Confdir_list	/usr/local/apache
Bin_dir	/usr/local/apache/bin



Sun Cluster Create New Resource

Configure Extension Properties

Resource Group: apache-sa
Resource Type: SUNW.apache

To view a property value, select its name from the Property list box.

To edit a property value, enter the new value in the value field or select from the drop-down list, if appropriate.

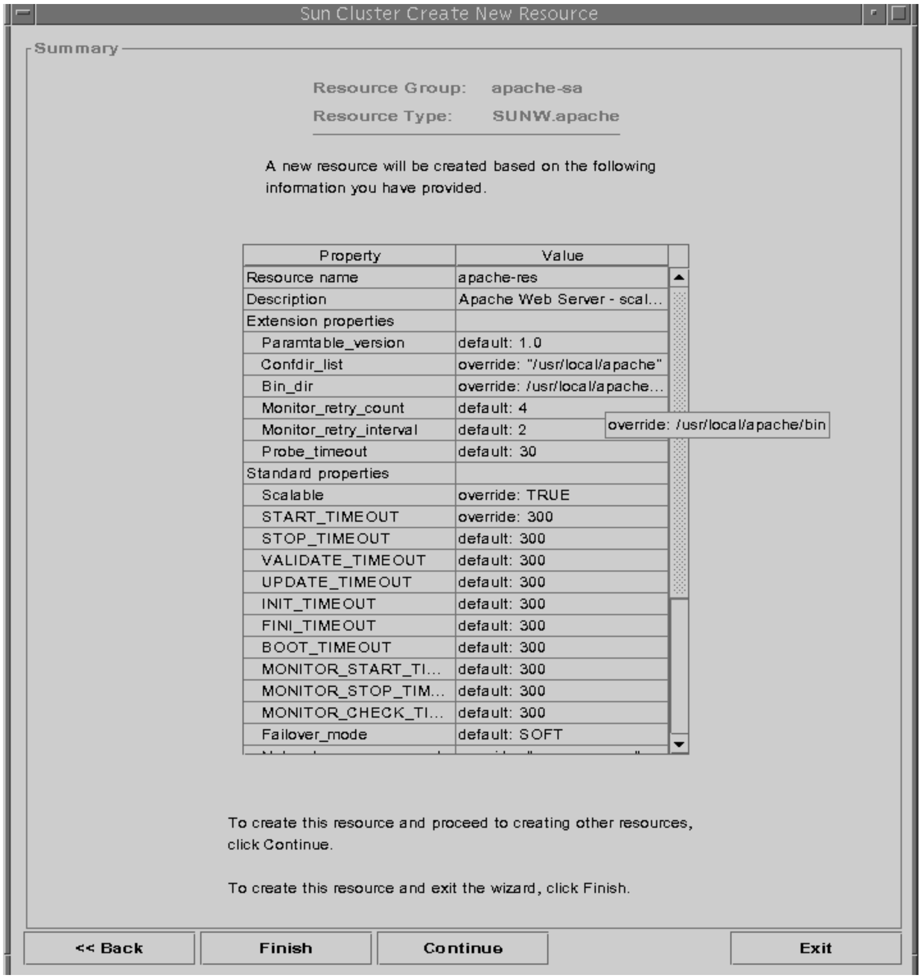
To change a selected property value back to the original default, click Restore Default.

Properties:
- Confdir_list
- Bin_dir
- Monitor_retry_count
- Monitor_retry_interval
- Probe_timeout

Name: Confdir_list
Description: The Configuration Directory Path(s)
Type: string array
Tunable: at creation
Minimum: <none>
Maximum: 4194303
Value: /usr/local/apache

Note that for the Apache Web Server the number of elements in the `Confdir_list` must be the same as the number of elements in the Port/Protocol List which were set (default was accepted) back on the General Scalable Properties screen.

7. **There is no need to adjust any advanced properties on this resource, so click** Basic **to move ahead to the** Summary **screen.**



Resource Group: apache-sa
Resource Type: SUNW.apache

A new resource will be created based on the following information you have provided.

Property	Value
Resource name	apache-res
Description	Apache Web Server - scal...
Extension properties	
Paramtable_version	default: 1.0
Confdir_list	override: "/usr/local/apache"
Bin_dir	override: /usr/local/apache...
Monitor_retry_count	default: 4
Monitor_retry_interval	default: 2
Probe_timeout	default: 30
Standard properties	
Scalable	override: TRUE
START_TIMEOUT	override: 300
STOP_TIMEOUT	default: 300
VALIDATE_TIMEOUT	default: 300
UPDATE_TIMEOUT	default: 300
INIT_TIMEOUT	default: 300
FINI_TIMEOUT	default: 300
BOOT_TIMEOUT	default: 300
MONITOR_START_TI...	default: 300
MONITOR_STOP_TIM...	default: 300
MONITOR_CHECK_TI...	default: 300
Failover_mode	default: SOFT

override: /usr/local/apache/bin

To create this resource and proceed to creating other resources, click Continue.

To create this resource and exit the wizard, click Finish.

8. **Click** Finish **in order to create this resource and exit the configuration GUI.**

9. **Refer to Chapter 6 for appropriate** `scswitch(1M)` **commands to manage and bring these resource groups online.**

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No.: TBD
Course No.: TBD
Revision 1.1, September 2000