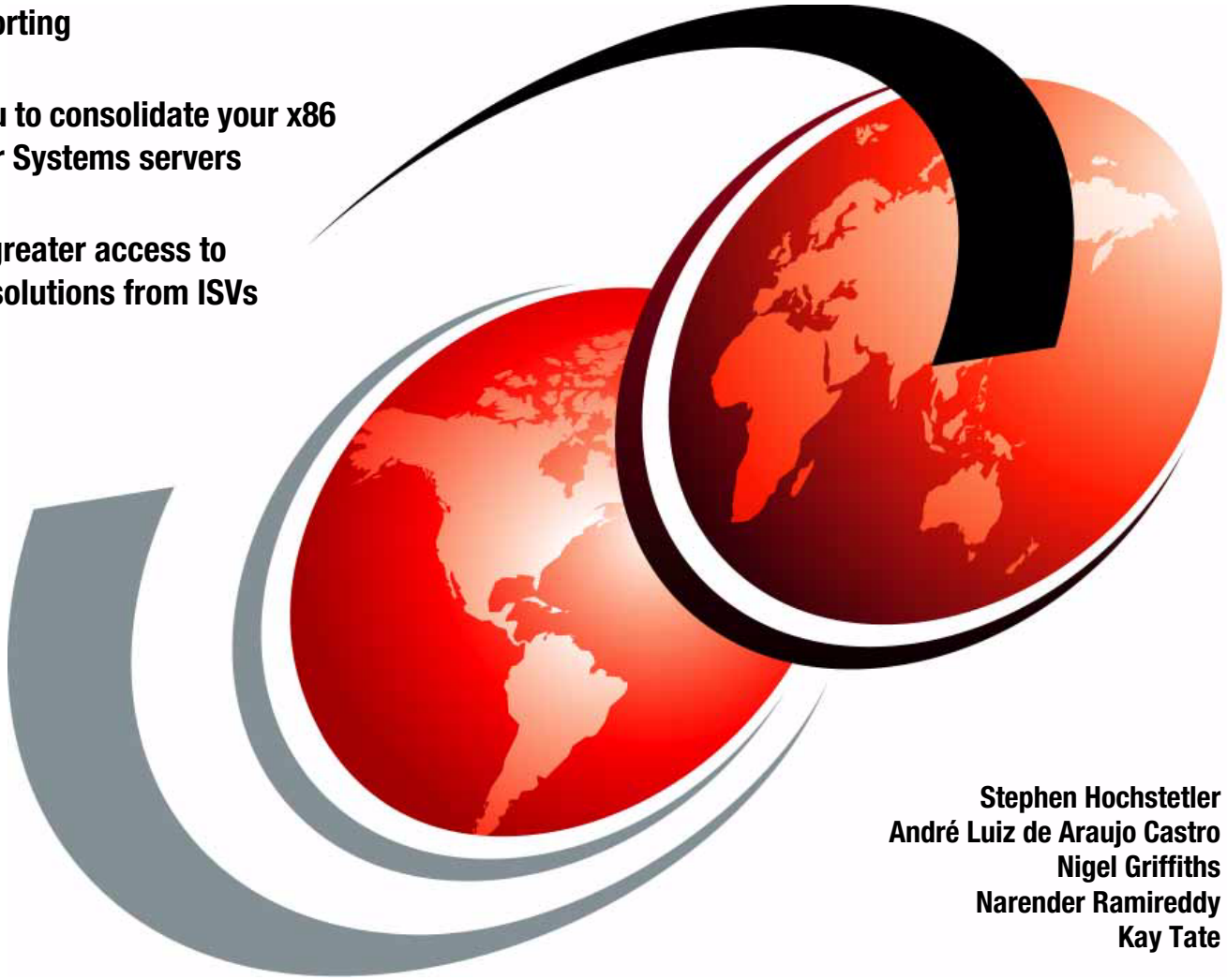
# Getting Started with PowerVM Lx86

- Runs x86 Linux binaries without porting
- Allows you to consolidate your x86 and Power Systems servers
- Provides greater access to software solutions from ISVs



Stephen Hochstetler
André Luiz de Araujo Castro
Nigel Griffiths
Narender Ramireddy
Kay Tate

Redpaper

ibm.com/redbooks

International Technical Support Organization

# Getting started with PowerVM Lx86

May 2008

REDP-4298-01

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

**Second Edition (May 2008)**

This edition applies to Version 1.2 of IBM PowerVM Lx86 (product number 5765-AVE).

This document created or updated on July 1, 2010.

# Contents



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
BladeCenter®
Chiphopper™
DB2®
developerWorks®
Domino®
IBM®
POWER Hypervisor™
Power Systems™
POWER5™
POWER5+™
POWER6®
PowerVM™
POWER®
Redbooks®
Redpaper™
Redbooks (logo)  ®
System p5®
System p®
System z®
Tivoli®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

IBM® PowerVM™ Lx86 brings new benefits to clients and application providers who want the reliability and flexibility of IBM Power Systems™ in their Linux® environment yet use applications that have not yet been ported to the platform. IBM PowerVM Lx86 allows most Linux x86 binaries to run unchanged on IBM POWER® processor-based systems using translation software that translates x86 instructions dynamically to POWER instructions and transforms x86 Linux system calls into calls to the POWER Linux kernel. It helps the consolidation effort by allowing Linux x86 applications to be consolidated onto Linux for Power partitions along side AIX® partitions.

This IBM Redpaper™ publication can help you understand how to optimize your IT environment by running x86 Linux applications within a Linux for Power environment. If you are running Linux for Power environments and x86 Linux environments, you can lower total cost of ownership by consolidating servers using IBM PowerVM virtualization. This paper shows the planning and installation choices as well as hints and tips to bring the most value to the IBM PowerVM Lx86 solution.

This paper is intended for independent software vendors (ISVs) and their clients to assist them in understanding and installing IBM PowerVM Lx86. Managers can use this paper as they set server strategies. System administrators can also use this paper as they install IBM PowerVM Lx86 with applications.

**Note:** During the open beta for this product, it was known as *System p® Application Virtual Environment* (System p AVE). Therefore, you might see that name in this paper. However, its formal product name is *PowerVM Lx86* with an announcement from IBM on 29 January 2008.

## The team that wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Austin Center.

**Stephen Hochstetler** is an IT Specialist. He writes extensively and teaches IBM classes worldwide on all areas of Linux, Power Systems, and Tivoli®. Before joining the ITSO, Stephen worked in Tivoli Services as a network management specialist.

**André Luiz de Araujo Castro** is a Senior Field Technical Sales Support (FTSS) for IBM Brazil working for the Power Systems sales team. He has 19 years of experience in the IT related environment. He has worked at IBM for nine years, and as an FTSS for the last seven years, supporting the Power Systems sales team in technical presales activities related to Linux, AIX, Power Systems, and BladeCenter®.

**Nigel Griffiths** is a Linux for Power Technical Advocate based in the U. K. but in a European role. He has 27 years of experience in the UNIX® performance, database, kernel, and development field. He has worked at IBM for 14 years in UNIX services, benchmarking, and product introduction and is a Technical Focus Group leader. He has developed the free and popular **`nmon`** performance monitor for AIX and Linux. He has written extensively on performance tuning, PowerVM, and Linux for Power.

**Narender Ramireddy** is an Advisory Software Engineer in the IBM Developer Relations organization. A key technical resource in the Chiphopper™ Dallas Technical Solution Center team, he currently works with ISV's in porting Linux applications to IBM platform.

**Kay Tate** is a Senior Program Manager in the ISV Business Strategy and Solutions Enablement organization. She has 37 years of experience in supercomputing, standards, and business application fields. She has written extensively on standards, Linux, and cross-platform Linux support for business applications. She is also a member of the Linux Standard Base Steering Committee.

Thank you also to the following people for their contributions to this project:

Robert Friske
IBM Southfield

Eduardo Matos Paccini
Paulo Pontes Vieira de Aragão
Ramon Sergio de Almeida Campos
IBM Brazil

## Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  **ibm.com**/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

# Introduction to PowerVM Lx86

IBM PowerVM Lx86 brings new benefits to clients and application providers who want the reliability and flexibility of Power Systems in their Linux environment yet use applications that have not yet been ported to the platform. IBM PowerVM Lx86 allows most Linux x86 binaries to run unchanged on POWER systems using translation software that translates x86 instructions dynamically to POWER instructions and transforms x86 Linux system calls into calls to the POWER Linux kernel. It helps the consolidation effort by allowing Linux x86 applications to be consolidated onto Linux for Power partitions along side AIX partitions.

Translation technology has been used successfully in the computer industry over the past several years by several computer vendors. The technology creates an environment on which the applications that are translated run on the new target platform, in this case on Power Systems servers. Mapping and caching techniques are used by the translator to enhance application performance within the environment. Translation has proven effective and has now been extended to running x86 Linux applications on Linux for Power systems and on logical partitions (LPARs).

By adding x86 applications to Power Systems Linux solutions, clients and independent software vendors (ISVs) have the opportunity to expand application use in consolidation scenarios. In these scenarios, clients can get more use of their hardware investments and improve total cost of ownership by reducing time-consuming tasks that are involved in managing multiple servers. At the same time, they can enhance service quality and mission-critical application availability.

## 1.1 Executive summary

Since its appearance in the IT market, Linux has become widely adopted by corporations of all sizes, and thousands of applications have been ported to this operating system. The x86-based platforms have been the preferential choice for having these Linux applications in production environments, mainly because of its low acquisition costs. Clients want to use Linux to simplify infrastructure and reduce costs. ISVs want to use Linux to simplify development and to support the environment and by doing so, also to reduce costs.

IBM embraced Linux as a strategic operating system for its server line several years ago. With the announcement of the POWER5™ based product line, several reliability, availability, and serviceability (RAS) and virtualization (fractional CPU, virtual disks, virtual LAN, and so forth) features which were before only available to IBM System z® (mainframes) clients, became also available to UNIX and Linux clients. The combination of these features allow clients to consolidate on a single server tens (or even hundreds) of different applications that are running on top of Linux or AIX with the confidence that they will have a very robust and optimized environment responding dynamically to these applications, both in terms of workload requirement and priority from a business perspective.

The combination of an open source operating system such as Linux and IBM Power Systems mainframe inspired technology, gives clients a very good alternative designed to reduce their Total Cost of Ownership (TCO) costs by reducing training costs, data center foot print, as well as energy and heat dissipation requirements. It can also help in reducing software licensing costs, because there are thousands of open source applications available, or by reducing the licensing costs from ISVs applications whose licenses are based on the number of processor cores, because the POWER6® technology has an enormous processing capacity per core as several public available benchmarks show.

Up to now more than 3000 applications have been ported to Linux for Power, but still there are thousands only ported to x86 based platforms. With the IBM PowerVM Lx86 environment, a client can take the original installation media of a Linux on x86 application and install it *as is* within a Linux for Power partition running on IBM Power Systems servers. There are many workloads that will run well within this environment. There are a few workloads that are not recommended to be run in this environment

From a client perspective, this environment allows a very transparent and easy way to start taking the benefits of such an advanced infrastructure platform. From an ISV perspective this environment provides an excellent opportunity for a jump-start onto a new marketplace, postponing the decision of the code porting from Linux on x86 to Linux for Power to a more appropriate moment in time if he judges necessary. It also allows the ISV the opportunity for keeping his development and support costs on a lower level, since there is only a single source (x86-based) code.

## 1.2 PowerVM Lx86 benefits

PowerVM Lx86 provides an environment on which the benefits of running on Power Systems can be available to clients who need to run applications currently available only on x86 platforms. The Power Systems benefits that are inherent in the system architecture are available transparently to the x86 application. The benefits of IBM System BladeCenter JS*xx* blades and Power Systems servers can be made available to clients who require an x86 application in their operational suite. No application changes are required for clients to get the following benefits:

- Outstanding Power Systems RAS
- PowerVM Edition features that can offer dramatic savings in hardware and systems management for multiple servers

Applications have enhanced Power Systems options for growth scenarios such as:

- *Scale up*: Increasing performance by adding more processors and memory
- *Scale out*: Increasing performance by adding additional complete systems
- *Scale within*: Consolidating multiple workloads onto a single server utilizing virtualization

Use of PowerVM Lx86 to run x86 applications on Power Systems servers is transparent to the applications. The objective for x86 applications is that *they just run*. Most 32-bit x86 binaries can be installed in the environment in the same way they would be installed on an x86 system. Then they could be executed via a command that starts an application in the environment. The x86 application binary is ready to run without the technical effort or the delay of a native port. We discuss details of applications that are good candidates for this environment in 4.1, “Applications that are good candidates for PowerVM Lx86” on page 32.

ISVs with Linux solutions on x86 can take advantage of this new technology to deliver their applications on Power Systems more easily. This can mean offering their clients the benefits of Power Systems above while delivering the new support rapidly. ISVs can realize increased opportunity at dramatically reduced cost and complexity by maintaining a single source base. They just need to check their licenses to ensure that they are applicable to clients running x86 binaries on Power Systems as well. From this perspective, PowerVM Lx86 helps the ISVs by instantly increasing the hardware portfolio for a given application and postponing the decision of native porting to a later point in time.

## 1.3 PowerVM Lx86 components

PowerVM Lx86 is made up of two basic components and a structured environment:

- **The translator**: The main component of the PowerVM Lx86 environment is the translator itself. The translator is currently available for download and will be part of future Power Systems deliveries.
- **The libraries**: The libraries are needed to make up the x86 environment for the application and need to be installed. The x86 Red Hat Enterprise Linux 4 update 4, SUSE Linux Enterprise Server 9 SP3, and SUSE Linux Enterprise Server 10 libraries are currently being used. As new releases come from these Linux distributors, IBM intends to continue to work with them for ready availability of their x86 libraries for PowerVM Lx86.

**Important:** The same release levels of libraries should be used for PowerVM Lx86 as are being used on the POWER system to ensure compatibility between the environment and the POWER kernel that delivers all of the system services to it.

- **The environment structures**: A set of system directories and system files are also required to complete the environment sufficiently for it to function on its own with POWER kernel services underneath it. The directory structure that is used by PowerVM Lx86 revolves around an x86 directory that contains all x86 applications running under the x86 libraries that the application needs. The translator enforces the file system view to start at this x86 directory, acting much like a **`chroot`** function, but specifically for the translator.

  When moving an application over, data files can be copied from the x86 system to the POWER system or can be NFS mounted for access if they are on the same network.

## 1.4 PowerVM Lx86 operation

After PowerVM Lx86 is installed on a POWER system, it is ready to be invoked for an x86 application. Each application execution results in a separate invocation of PowerVM Lx86 in user space. Using POWER systems management tools, all of the separate processes can be displayed and tracked for management purposes like other processes running on the system. Likewise, a view of the file system from POWER shell command will show all of the file systems for Linux for Power and PowerVM Lx86 files.

If using process monitoring or file system views from within the PowerVM Lx86 environment, only those entities that are specific to the PowerVM Lx86 environment will be visible. The POWER application processes and the file system that has not been identified explicitly as accessible to the PowerVM Lx86 environment, are not visible to the requester.

When moving an application over, data files can be copied from the x86 system to the POWER system or can be NFS mounted for access if they are on the same network.

PowerVM Lx86 attempts to unify the definitions of users, groups, and passwords by collecting information from both the native POWER system and the x86World and producing a merged view of the /etc/passwd, /etc/group, and /etc/shadow files.

More operational details and options are located in the *PowerVM Lx86 Administrator's Guide*.

## 1.5 PowerVM Lx86 support

Support for PowerVM Lx86 is provided by IBM. If an application provider or user is unable to reproduce a problem on an x86 system, there is a possibility that it might be an issue with PowerVM Lx86. IBM will work with the issue reported to capture relevant information so that the item can be diagnosed and tracked to resolution. For information about how to contact IBM, see “Help from IBM” on page 47.

2

# PowerVM Lx86 architecture

This chapter provides technical information about how PowerVM Lx86 runs x86 Linux applications within the Linux for Power environment. These applications can be run on the same IBM Power Systems server with other Linux for Power applications or with AIX applications. This freedom of choice enables you to consolidate and optimize your IT workload on fewer servers when you are running virtualized environments.

## 2.1 PowerVM Lx86 virtual x86 environment

PowerVM Lx86 enables Linux for Power servers to run Linux on x86 applications alongside native Linux for Power applications. The Linux on x86 applications run on SUSE Linux Enterprise Server 9 SP3, SUSE Linux Enterprise Server 10, or Red Hat Enterprise Linux 4 U4 or later distributions within a Virtual x86 Environment (VxE). PowerVM Lx86 creates a VxE for each x86 application as it is run. No modifications or recompilations of the Linux/x86 applications are needed.

An Linux on x86 application running within a VxE appears to be just another user-space POWER process. PowerVM Lx86 does not run the x86 kernel on the server. The operating environment for the Linux on x86 applications running within the VxE appears to be Linux on x86. PowerVM Lx86 translates and maps all requests made from the within the VxE to the underlying Linux for Power operating system and POWER processor.

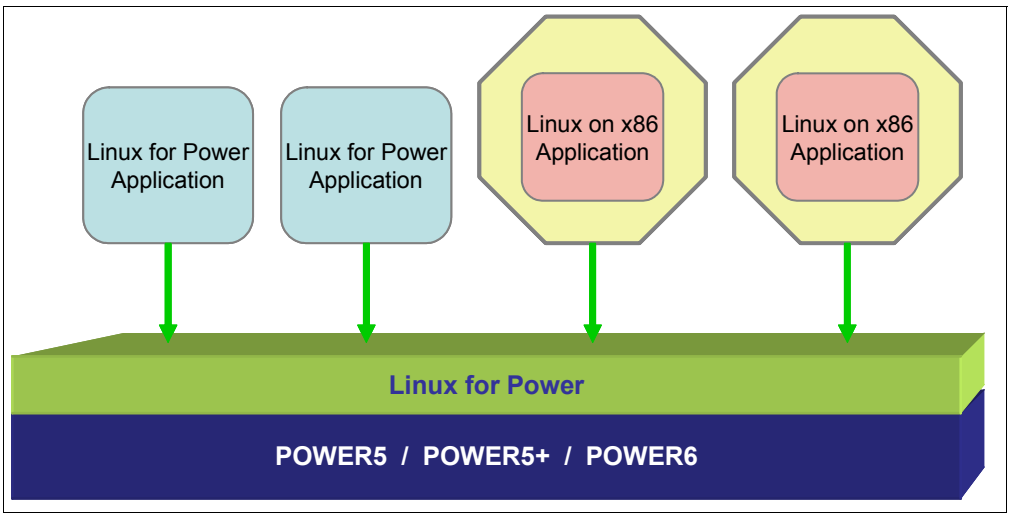



*Figure 2-1 Virtual environment for x86*

PowerVM Lx86 installs in two parts on the Linux for Power server:

- The first part is a single rpm package that includes PowerVM Lx86 translator, the translation daemon, and the tools to run VxE.
- The second part is the x86World that contains the x86 libraries, commands, and utilities that are installed from a SUSE Linux Enterprise Server 9 SP3, SUSE Linux Enterprise Server 10, or Red Hat Enterprise Linux 4 U4 or later x86 distribution.

The choice of the Linux on x86 distribution must either match the version of Linux running on the POWER server or be one update or service pack earlier than the POWER Linux version.

Figure 2-2 shows the PowerVM Lx86 package information and its contents. The PowerVM Lx86 daemon, which needs to be running all the time, is started when the server restarts by the rc.d scripts. The p-ave startup scripts in rc.d directories are linked to the /etc/init.d/p-ave script.

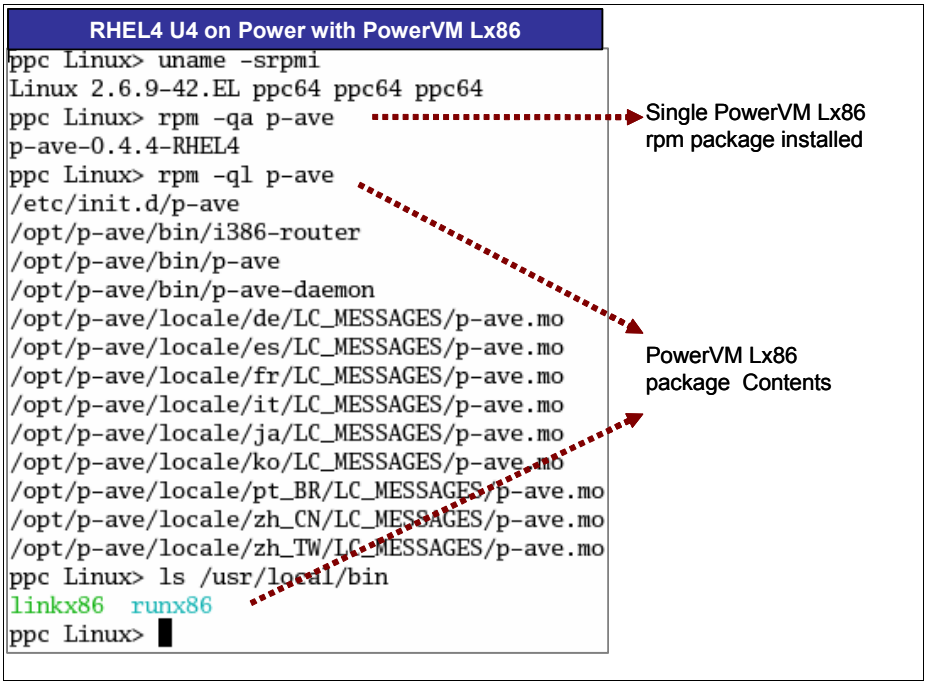



*Figure 2-2 PowerVM Lx86 package contents*

Figure 2-3 shows the x86World, which is required by the VxE to work. The x86World contains x86 libraries, commands, and utilities and is installed from a Red Hat Enterprise Linux 4 Update 4, SUSE Linux Enterprise Server 9 SP3, or SUSE Linux Enterprise Server 10 or later x86 distribution.

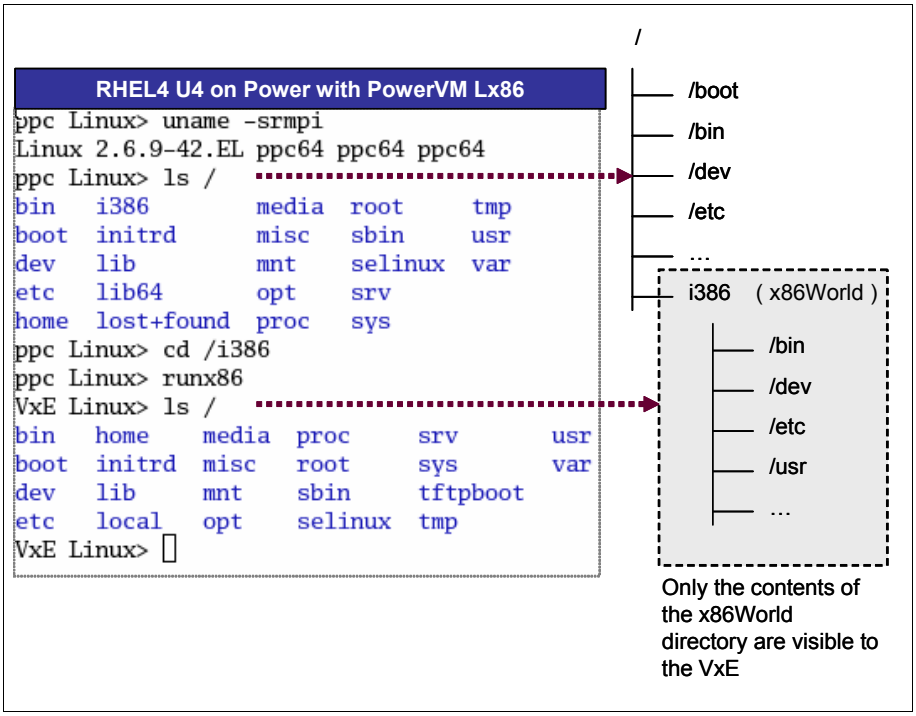



*Figure 2-3 The x86World directory*

## 2.2 PowerVM Lx86 translation process

PowerVM Lx86 works by translating x86 instructions dynamically to POWER instructions and by transforming x86 Linux kernel calls to POWER Linux kernel calls. It operates much like the Just-in-Time (JIT) compiler in a Java™ system.

The PowerVM Lx86 translation is multi-stage and iterative. After an x86 application is loaded into memory, it undergoes a continuous process of translation and optimization as shown in Figure 2-4 on page 9.

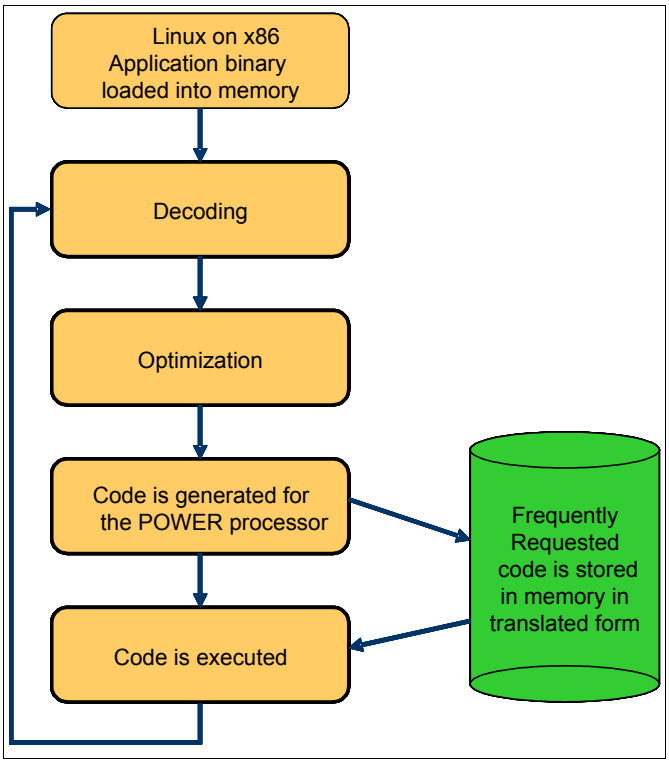



*Figure 2-4 PowerVM Lx86 translation process*

The translation is a three-stage process:

- **Decoding**: x86 binary instructions from the x86 binary are decoded as the application requests them.
- **Optimization**: The optimization is iterative, so more optimization is done on frequently-used code.
- **Generation of POWER code**: Frequently-used code is stored in memory, so it does not need to be translated again the next time it runs.

When an application is started using the PowerVM Lx86 environment, the environment is activated for the application to run in.Instructions are issued from the application to the translator. The instructions are translated to POWER instructions, as shown in Figure 2-5.

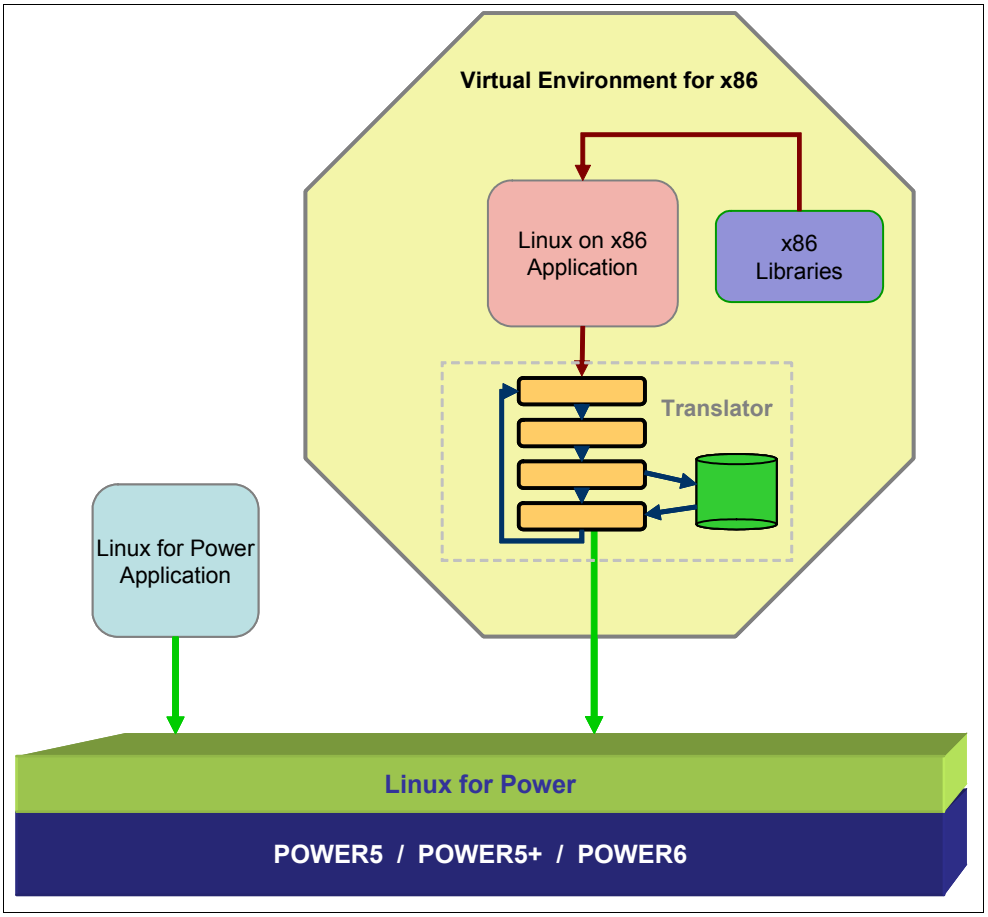



*Figure 2-5 Virtual Environment for x86 translator*

As the instructions are translated, they are cached. Instruction sequences that are executed more times are optimized in the background. The result of these actions is that:

1. The instructions will not need to be translated again.
2. They will run faster the more they are run.

Instructions that are not issued are not translated. Most often, transactional types of applications, Web servers and user written applications, and client (user facing) applications that lend themselves to this optimization, and we see a good performance for them running in translated mode.

Because translated applications can run side by side with native applications, they can access the same data and open sockets between the two processes to communicate. System calls are mapped each time they occur. Applications that make a large proportion of systems calls in their execution will not have the optimization benefits that the instruction translation has from a performance standpoint.

x86 applications can be installed using their normal installation scripts from within the PowerVM Lx86 environment. To start the same shell already in use in the Lx86 environment, use the start environment command without arguments as follows:

```
runx86
```

You terminate the shell and environment with an **exit** command.

After installation, x86 applications are run using the same command with the application information as arguments:

```
runx86 ./myx86binary
```

Execution of the command starts the environment, and then the x86 binary runs.

**Note:** Any Linux on x86 application that is coded little-endian specific should work as it is under PowerVM Lx86. PowerVM Lx86 translates little-endian x86 applications to run on the underlying Linux for Power system, even though the Power system is big-endian.

If an x86 application needs to access files outside of the x86World, there is a mechanism known as *escape* to achieve this. An escape links a path inside x86World with a path outside x86World. You can create an escape with the **linkx86** command:

```
linkx86 /var/myapp
```

This command creates a link from /var/myapps within x86World to the /var/myapps within the host Linux for Power.

### 2.2.1 Linux on x86 system calls

The Linux on x86 applications use system call instructions to request services from the Linux on x86 kernel. PowerVM Lx86 does not run the Linux on x86 kernel, so when the Linux on x86 application is executing within the VxE the translator maps the Linux on x86 system call instructions to their Linux for Power equivalent. The mapped system call instructions are served by the Linux for Power kernel that is running on the host system. This is one more reason why you need to have the same Linux on x86 distribution version as the host Linux for Power version when installing PowerVM Lx86 on the server to avoid incompatibilities between Linux versions.

### 2.2.2 Linux on x86 floating point instructions

x86 floating point instructions are translated to POWER floating point instructions as long as the precision is sufficient to fit in 64 bits. If full precision of the internal x86 80-bit registers is required by an application, it is simulated correctly, but requires a software library to do it, thus slowing performance for these instances. In general, it is recommended that high performance computing (HPC) applications be recompiled on POWER and ported natively.

## 2.3 Little endian and big endian

**Note:** PowerVM Lx86 translates x86 applications (which are themselves little-endian) into the underlying Linux for Power system, even though the POWER system is big-endian.

In big endian the high-order byte of the number is stored in memory at the lowest address, and the low-order byte at the highest address as shown in Example 2-1.

*Example 2-1 Big endian*

```
A 4-byte Integer with hex value 0x1A2B3C4D will be arranged in memory as follows:
    Base Address+0  0x1A
    Base Address+1  0x2B
    Base Address+2  0x3C
    Base Address+3  0x4D
```

In little endian, the number is stored in memory at the lowest address and the high-order byte at the highest address, as shown in Example 2-2. x86 processors use little endian byte order.

*Example 2-2 Little endian*

```
4 byte Integer with hex value 0x1A2B3C4D will be arranged in memory as follows:
    Base Address+0  0x4D
    Base Address+1  0x3C
    Base Address+2  0x2B
    Base Address+3  0x1A
```

POWER processors can switch big endian and little endian. The Linux for Power and AIX operating systems boot big endian.

Taking a Linux on x86 application along with its data and running under PowerVM Lx86 will not have any endian issues. PowerVM Lx86 translates x86 applications and runs them in little-endian mode just like a native Linux x86 server.

The Linux on x86 applications can also share data files with Power applications that have a defined format, which takes into account the endian format of the data. For example, x86 OpenOffice can share data files with the Power version of OpenOffice. Because there is a defined file format, both versions of the applications will save the data in the agreed endian (if necessary), independent of the endian of the application or the CPU on which it is running. It is the same for network traffic. Little- and big-endian systems can communicate over the network, even though the network data is in fact big endian.

3

# Installing and troubleshooting

This chapter explains the basics of the installation process, some things to try after IBM PowerVM Lx86 is running to get you familiar with the environment, and then hints of what to do if you have any issues.

## 3.1 The installation files included in the package

When you receive the PowerVM Lx86 software package, the first thing you notice is that the package is not very large. The beta release we used is less than 4 MB in size.

On Linux unpack the package as follows:

- Uncompress the file with: **`gunzip filename.tar.gz`**
- Extract the files with: **`tar xvf filename.tar`**

After you extract the files, you find the list of files. The files include two documents that inform you of the file content:

- *Release Notes*
  - Currently about 14 pages, it documents the details of this particular release.
  - The *Release Notes* include details of changes from previous releases, known issues with this release, and how to report problems so that they can be fixed.
- *Administration Guide*
  - Currently about 50 pages and likely to grow over time, this document details how to install and use PowerVM Lx86.
  - The *Administration Guide* includes the following sections:
    - An Overview and Concepts
    - Full details on the installation process
    - Systems Administration reference

**Tip:** To save time, you should read these documents completely before installing and using PowerVM Lx86. These documents are well written by the developers and include valuable and helpful information.

It is not our intention to reproduce these two documents here in this paper. In addition, the installation tool details might well change in future releases. In this paper, we cover the overall installation process so that you can know what to expect and have a clear idea of what you are doing at each stage. Then, we will show some examples of using the PowerVM Lx86 that we used to investigate this new environment.

## 3.2 Before beginning the installation

You need Linux running on your Power Systems server or Power Systems LPAR to support PowerVM Lx86. This installation is a straightforward installation of the Linux version that you prefer. There is nothing special or unusual for this installation nor is the installation method or media (CD, DVD, or network) important. You do need to install one of the supported versions. At the time of writing this is:

- Novell SUSE Linux Enterprise Server 10
- Novell SUSE Linux Enterprise Server 10 Service Pack 1
- Novell SUSE Linux Enterprise Server 9 Service Pack 3
- Red Hat Enterprise Linux 4 AS Update 4
- Red Hat Enterprise Linux 4 AS Update 5

If you have a previous release, update, or service pack, you need to upgrade your Linux for Power system before using PowerVM Lx86. *This is not optional*.

To run x86 Linux binaries on a Power Systems server, you need something more than just the PowerVM Lx86 software. Any running application needs many other files such as the libraries that are loaded when it starts, the system files that it expects in the correct format, as well as other operating system commands (their binary files) that it might call and rely on. These need to be installed along with the PowerVM Lx86 package itself.

Fortunately, these are available and in a suitable format. For example, if for Linux for Power you are running SUSE Linux Enterprise Server 9 Service Pack 3, then you probably installed from the CD-ROMs for the POWER platform as supplied by Novell for the POWER servers. There is also the same operating system version available for the x86 platform and the same release from Novell.

The same goes for Red Hat Enterprise Linux versions. These x86 versions are available from Red Hat and Novell and must match the Linux for Power version that you are currently running on your Power Systems or LPAR. Specifically, you need all of the RPM packages from the installation media. To make it simpler, we recommend you obtain a copy of the CD-ROMs or DVD-ROMs. See Table 3-1.

*Table 3-1 Linux distributions*

| Linux on x86 version you want to install | Required base Linux for Power Version |
|---|---|
| SUSE Linux Enterprise Server 9 SP3 | SUSE Linux Enterprise Server 9 SP3 |
| SUSE Linux Enterprise Server 10 | SUSE Linux Enterprise Server 10<br>SUSE Linux Enterprise Server 10.1 |
| SUSE Linux Enterprise Server 10.1 | SUSE Linux Enterprise Server 10.1 |
| Red Hat Enterprise Linux AS 4 U3 | Red Hat Enterprise Linux AS 4 U4 |
| Red Hat Enterprise Linux AS 4 U4 | Red Hat Enterprise Linux AS 4 U4<br>Red Hat Enterprise Linux AS 4 U5 |
| Red Hat Enterprise Linux AS 4 U5 | Red Hat Enterprise Linux AS 4 U5 |

Note the following requirements:

- If you have Red Hat Enterprise Linux 4 update 4 for POWER on the Power Systems or LPAR, you need the Red Hat Enterprise Linux 4 update 4 version for x86 media. However, you can use the AS versions of Red Hat Enterprise Linux.
- If you have SUSE Linux Enterprise Server 9 Service Pack 3 for POWER on the Power Systems or LPAR, you need the SUSE Linux Enterprise Server 9 Service Pack 3 version for x86 media.
- If you have SUSE Linux Enterprise Server 10 base release for POWER on the Power Systems or LPAR, you need the SUSE Linux Enterprise Server 10 base release version for x86 media.

**Important:** Make sure the x86 distribution is the same as your Linux for Power distribution with the caveats above. Do not try to use any other version because other versions will not be supported by the PowerVM Lx86 installer.

**Tip:** Because obtaining the installation media for x86 Linux can take some time in either downloading or ordering the media, we recommend that you do obtain this media well in advance of beginning the PowerVM Lx86 installation.

The overall tasks for the PowerVM Lx86 installation involve:

1. Checking the hardware and software prerequisites. Check the *Release Notes and Administration Guide* for specific details, but briefly these prerequisites are:
   - Only POWER5, POWER5+™ or POWER6 or JS21, JS12, or JS22
   - 1 GB of memory
   - Disk space for the RPMs and x86 environment (allow 4+ GB)
2. Unpacking the PowerVM Lx86 package.
3. Making the Linux of x86 RPMs available on the system.
4. Creating the x86 environment.
5. Installing the x86 libraries and binaries from the x86 media (that is RPMs).
6. Making some settings in the Linux for Power environment.

**Note:** Tasks 3 through 6 are perform by the PowerVM Lx86 Installer.

Figure 3-1 shows an overview of these tasks.

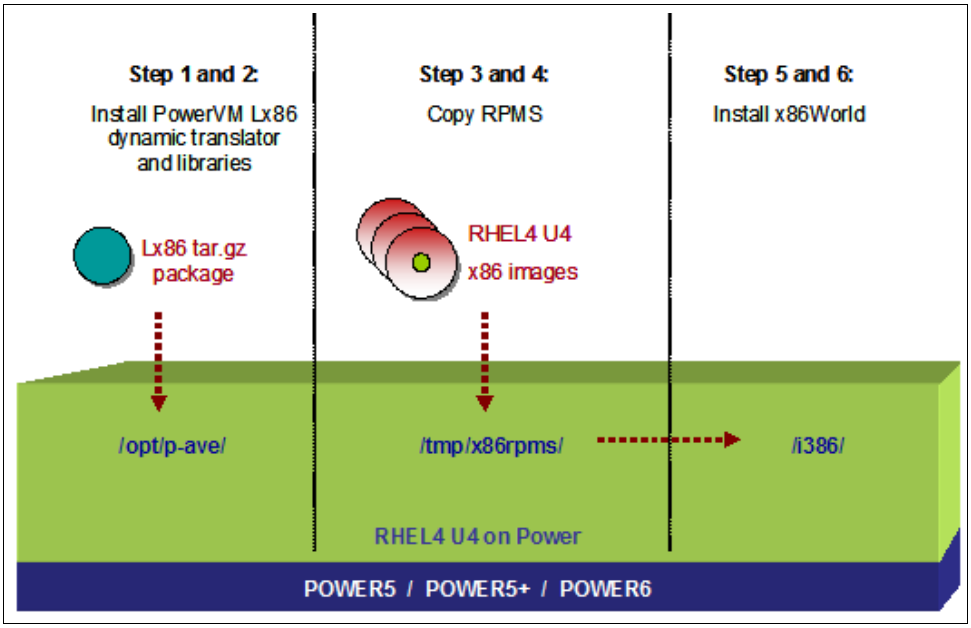



*Figure 3-1 PowerVM Lx86 installation environment*

In Figure 3-1:

- On the Power Systems server or LPAR, the regular Linux for Power file systems are at the top of the diagram.
- The PowerVM Lx86 installer installs the x86 RPM files on to the system to build the x86World. By default the x86World is installed in directory /i386.

- We need to collect the RPM files from the Linux on x86 media or directly from .iso images and place them in a directory so that they can be used to create the x86World environment. The default directory is /tmp/x86rpms.
- The Installer takes the RPM files and builds the x86World by default in directory /i386. This directory contains all the commands, tools, files, and libraries that the running x86 binaries will need.
- By default, the /home directory can be accessed from the Linux for Power side (as normal) and also from the PowerVM Lx86, which makes it easy to move files between the environment and to access shared files jointly.

### 3.2.1 Checking the hardware and software prerequisites

The *Administration Guide* discusses all prerequisites in detail. As a example, here is what we used for our trial of PowerVM Lx86:

- Our system was a System p model p510 with 2 CPU at 1.65 GHz, 2 GB memory running IVM with two internal 73 GB SCSI disks.
- We created a Shared Processor Logical Partition (LPAR) for PowerVM Lx86 testing.
- The LPAR CPU settings were 0.2 CPU Entitlement, uncapped and virtual processors 2.
- LPAR memory was 768 MB. Technically, this value is below the minimum, but we had no choice because of the system. We did not have issues but note that we only had a handful of users. If you want support for more users, then you will need 1 GB of LPAR memory or more.
- LPAR disk was 8 GB of disk space as a single large root file system and 256 GB of paging space.
- We did a fresh installation over the network of Red Hat Enterprise Linux AS 4 Update 4 for Linux for Power and the exact same release of Linux for x86 on CD-ROMs for PowerVM Lx86.
- We used all the default locations for PowerVM Lx86 with the exception that the RPM files were placed in the /x86rpms directory, which makes the installation much simpler to perform and understand.
- The /home file system was visible from Linux for Power and the PowerVM Lx86.
- We performed all the tests as the root user. We later tried running as a regular Linux for Power user with no issues after we created the user.

**Note:** We tried SUSE Linux Enterprise Server 10 in a different LPAR, and it behaved exactly the same as Red Hat Enterprise Linux.

## 3.3 Installing PowerVM Lx86 and x86World

This section describes the procedure to install PowerVM Lx86 and x86World.

The procedure assumes that you are installing on to a system for the first time, or that any previous installations of PowerVM Lx86 have been uninstalled. If the default locations are accepted, then the PowerVM Lx86 translation software installs in the /opt/p-ave directory. You can install it in an alternative location if necessary.

**Note:** The default location for the x86World libraries and binaries is the /i386 directory, which is frequently referred to as X86WORLD_ROOT in this document. Also note that this procedure uses RHEL as an example, but the SLES installation is very similar.

Follow these steps to install PowerVM Lx86 and x86World

1. Become root on the Power Systems server.
2. Download and untar the tar file.
3. Run the PowerVM Lx86 installation script:

   ```
   ./installer.pl
   ```

4. You are asked to read through and agree to the license agreement before continuing with the installation. The following prompt displays:

   ```
   Host OS detected as Red Hat Enterprise Linux release 4, Update 4
   Welcome to the PowerVM Lx86 (System p AVE) Installer
   ```

   To run Linux/x86 applications the installation require s Linux/x86 application binaries, libraries, and infrastructure files, as well as the PowerVM Lx86 (System p AVE) software.

   Refer to the *PowerVM Lx86 (System p AVE) Administration Guide and Release Notes* for further details about the installation requirements.

   ```
   You must agree to the following license to install PowerVM Lx86. If you do not
   accept the license terms, the installation will abort.
   Page through the license with SPACE. Press enter to view the license.
   ```

5. Press Enter to view the license, and press Enter again to scroll to the next page. The following prompt displays:

   ```
   Do you accept the license terms? [Y/N]
   ```

6. Enter **y** to agree to the terms and to continue the installation. The license text is saved to disk. You can print this file for your records. You will be asked for information that will be sent by e-mail to activate the product. The following prompt displays:

   ```
   Registration with IBM You must provide some information to IBM before starting
   the installation process. This Activation information will be sent to IBM as an
   email. Note: You must provide your company name for the installation to
   proceed.

   All other details are optional. The Activation email will also include some
   details about this system. Once you have entered this information it will be
   presented to you to review. You will have the option to send the information to
   IBM via email or abort the installation. Please enter the name of your company:
   ```

7. You must enter the name of your company. You are also prompted for your name, address, phone number, and e-mail address. These fields are optional.

8. Enter the requested information, or to skip any of the questions, press Enter at the prompt. The system model number, serial number, and current time and date information are collected automatically from your system. All the information collected displays and is similar to the information shown in Example 3-1:

*Example 3-1 System information collected during installation*

```
The following is about to be sent to IBM:
Date: Wed Mar 28 15:03:33 GMT 2007
System: IBM,nnnn-nnn
Serial: IBM,nnnnnnnnn
Company: Your Company
Name: Your name
Address: Your address
Phone Number: 555-5555
Email: yourname@example.com
Is the above information correct? [Y/N]
```

If this information is correct, enter y. The following prompt displays:

```
Send the above information to IBM (pave@us.ibm.com)?

(If you answer 'no' here, the installation will be aborted.) [Y/N]
```

9. If you agree with sending the information to IBM, press Enter, and you then see the following output:

```
 Email sent successfully.
```

10. For RHEL 4 only, the following prompt is displayed Example 3-2:

*Example 3-2 Registering with Red Hat Inc.*

```
Registration with Red Hat Inc. Would you like to Register? [Y/N]

If you would like to register with Red Hat to upgrade your system to support
both Linux on x86 and Linux on POWER, then enter y and go to the next step. If
you do not wish to register now, you can register after installation.
```

11. For RHEL 4 only, the following prompt is displayed Example 3-3:

*Example 3-3 Entering your company name with the Red Hat Network*

```
The Linux/x86 binaries and libraries in the x86 World can be kept up to date
using the Red Hat Network. To receive this support, you must upgrade your
current Red Hat subscription to support both Linux/x86 and Linux/POWER. This
upgrade is at no additional charge.

To upgrade the subscription on this system, you need to provide some
information that will be emailed to Red Hat Inc. Once this has been sent, you
will receive an acknowledgement email from Red Hat within approximately the
next 24 hours (1 business day). You will then receive another email confirming
that the subscription has been upgraded within approximately the next 48 hours
(2 business days).

Please enter the name of your company [Your Company]:

To receive an upgraded Red Hat Network Subscription, you must provide your
name, your company name, and an e-mail address, and, optionally, your Red Hat
login name and your Red Hat account number. Your name, your company name, and
```

```
an e-mail address are provided as default from the IBM Registration process you
entered above.
```

12. For RHEL 4 only, if you are happy with the defaults, press Enter for each prompt, and then, optionally, answer the login and account number prompts. To skip these last two questions, press Enter at the prompt. The system model number, serial number, and current time and date information are now collected automatically from your system. All the information collected is displayed on the screen. The following prompt displays:

```
Date: Wed Feb 28 15:03:33 GMT 2007
Company: Your Company
RHN Login:
Account No:
Name: You name
Email: yourname@example.com
Is the above information correct? [Y/N]
```

13. For RHEL 4 only, if the information is correct, enter y. The following prompt displays:

```
Send the above information to Red Hat Inc. (pave@redhat.com)? (If you answer
'no' here, the installation will be aborted.) [Y/N]
```

    If you agree with sending the information to Red Hat to upgrade your subscription, press Enter, and you see this output:

```
Email sent successfully.
```

    The following menu displays:

```
1. Install Software.
2. Upgrade Software.
3. Uninstall Software.
4. Show Installed Products.
5. Configure Software.
6. Quit.
Please select the type of operation you would like to perform:
```

14. Enter 1 to begin the installation. The following menu displays:

```
1. Install p-ave + x86 World.
2. Install p-ave.
3. Install x86 World.
4. Go back to the Main Menu.
5. Quit.
Please select the type of operation you would like to perform:
```

15. Enter **1** to begin the installation of the PowerVM Lx86 package and x86 World. The following prompt displays:

```
Install p-ave to: [/opt/p-ave]
```

16. Press Enter to accept the default installation location, or enter an alternative location. The following prompt displays:

```
Log files created by p-ave will be stored in [/var/opt/p-ave/log]:
```

17. Press Enter to accept the default log file location, or enter an alternative location. The following prompt displays:

```
Install x86 World to: [/i386]
```

18. Press Enter to install x86World in the recommended default location, or enter an alternative location. The following prompt displays:

```
1. p-ave-<version>-<OS>
2. Go back to the Main Menu. Please select the version of p-ave you would like
to install:
```

19. Select the version of PowerVM Lx86 to install by entering 1. The following prompt displays:

```
Which distribution will you be installing for your x86 World?
1. Red Hat Enterprise Linux AS release 4, Update 4 (full)
2. Red Hat Enterprise Linux AS release 4, Update 4 (minimal)
3. Red Hat... Please select a distribution from the list above:
```

20. Enter the number of the appropriate distribution. The following prompt displays:

```
Where are the home directories located on this system (type 'skip' if you don't
want the x86 World to share your home directories)? [/home]
```

21. This step escapes the x86World home directory. Press Enter to accept the default home directory location, or enter an alternative location. If you do not want to escape the home directories, enter skip. If your environment uses multiple locations for home directories, such as the /home1 directory and the /home2 directory, then you need to use the linkx86 script to configure additional escapes after the installation script has finished. The prompt shown in Example 3-4 displays:

*Example 3-4 Configuring additional escapes*

```
To create the x86 World the p-ave installer will need the appropriate set of
x86 RPMs. This set of RPMs is generated from the Red Hat Enterprise Linux AS 4
Update 4 FULL (<install type>) x86 install media (ISO or CD/DVD-ROM).

Please enter the directory or files containing Red Hat Enterprise Linux AS
release 4, Update 5 (<install type="">).

- ISO9660 images are accepted (if they end with .iso)
- RPM files are accepted (if they end with .rpm)
- If you have a CD/DVD-ROM, please mount it and enter the mount path

You can enter each file or mount path individually or provide a pattern to
select multiple files at once, e.g. /files/rhel/RHEL4-U5-i386-AS-disc*.iso.

To view a list of missing RPM files, type 'list'
To return to the main menu, type 'skip'

Enter directory or file:
```

22. To view the list of outstanding RPMs, enter list. You can source the RPMs from CD, DVD, ISO images, or a directory of RPMs. You can add the sources individually, and the required RPMs will be copied from the media. For CD and DVD installation, you can mount the disks individually.

23. If mounting CDs or DVDs, mount the first disk, and then enter the mount location. After the RPMs are copied from that disk, unmount the disk and mount the next disk. Type in the new mount location, and continue until all of the RPMs are copied across.

    If you are installing from ISO images, you can use a pattern match if the ISOs have similar names. For example, you can add RHEL4-U4-i386-AS-disc1.iso and RHEL4-U4-i386-AS-disc2.iso by typing in the pattern RHEL4-U4-i386-AS-disc*.iso.

24. (If mounting CDs or DVDs) Enter the location of the media. For each medium, the installer prints the following message:

```
nnn required RPMs found. Copying...
```

After all of the required RPMs are copied onto the local system, the RPMs are installed into x86World, and the following output displays:

```
All RPMs found successfully.

About to proceed with the following:

- Install p-ave to /opt/p-ave
- Log files will be stored in /var/opt/p-ave/log
- Install x86 World to /i386
- Selected distribution is Red Hat Enterprise Linux AS release 4, Update 5
(minimal)
- Home directories are in /home
1. Continue.
2. Change settings.
3. Go back to the Main Menu.

Please select the type of operation you would like to perform:
```

25. If the settings are correct, Enter **1** to continue.

```
p-ave installed successfully. Installing x86 World...
Step 1/2: Installing base RPMs...
100 [====================>]
[ OK ]
Step 2/2: Installing x86 world...
100% [====================>]
```

26. Enter 6 to quit the installation. The installation exits, with the following summary of the installation process and details of where the installation log and activation e-mails are stored, as shown in Example 3-5:

*Example 3-5 Summary of the installation process*

```
The Activation email data is stored on the system in
/etc/opt/p-ave/activation-detail.

If this system is not set up to to send email, please email this data file to
IBM at <pave@us.ibm.com>.

The Red Hat Subscription email data is stored on the system in
/etc/opt/p-ave/redhat-activation-detail.

If this system is not set up to to send email or you do not get an
acknowledgement email from Red Hat within the next 24 hours, please email this
data file to Red Hat at <pave@redhat.com>. This will ensure that your Red Hat
subscription is upgraded.

License text saved in
/etc/opt/p-ave/redhat-activation-detail. You may print this file for your
records.
```

```
Thank you for using the System p AVE Installer

Today you performed the following transactions:
Accepted license agreement
Registered with IBM
Registered with Red Hat
Installed System p AVE
Installed installer into /opt/p-ave/installer
Installed x86 World (<OS><install type>)

End of Transaction log.

The full log of your session is in /tmp/p-ave_install_xxxxxx.log
```

## 3.4 A first look around PowerVM Lx86

The first questions to address is how best to gain access to the computer where you installed Power Systems or LPAR given that most Linux for Power do not have graphics adapters, keyboard, nor a mouse? Two options are immediately obvious:

- First, you can telnet, or ssh to the server, logon and then start PowerVM Lx86. This gives you just one session or window to the computer and to understand PowerVM Lx86. It was found that a number of sessions or windows was best because you can then have one session or window in the Linux for Power environment and one session or window in the PowerVM Lx86. You can then investigate the differences by opening a number of sessions.
- A second alternative is to start a VNC server on Linux for Power and then a VNC viewer on your workstation or Windows® computer. Then, you can have multiple xterm's open at one time and run X Windows graphics applications too.

**Tip:** We recommend you setup and use VNC to access your Linux for Power environment if you need a graphical desktop.

Some x86 applications often have prerequisites of other RPMs that are supplied on the Linux media. Fortunately, you have these in the RPM repository that you used to install PowerVM Lx86 and its x86World. It is worth keeping this repository to make installation of other RPM files simple. The alternative is using a network installation process and accessing the RPM files that way.

**Tip:** Have all the RPM files available to make the installation of applications simple.

Before we take a look at running the PowerVM Lx86, take a look at what was installed. As the root user, log in to Linux for Power. Then, run:

```
ls /x86rpms
```

You should see a list of RPM files as shown in Example 3-6.

**Note:** We installed the RPM files in this directory rather than the default /tmp/x86rpms.

*Example 3-6 RPMs copied for x86World installation*

```
[root@ivmb4]# ls -al /x86rpms| more
total 355124
drwxr-xr-x   2 root root    20480 Mar 12 19:58 .
drwxrwxrwt  17 root root     4096 Mar 15 04:02 ..
-rw-r--r--   1 root root   309473 Mar 12 19:58 alsa-lib-1.0.6-5.RHEL4.i386.rpm
-rw-r--r--   1 root root   174373 Mar 12 19:58 atk-1.8.0-2.i386.rpm
-rw-r--r--   1 root root   101847 Mar 12 19:58 audiofile-0.2.6-1.el4.1.i386.rpm
-rw-r--r--   1 root root   197170 Mar 12 19:58 audit-1.0.14-1.EL4.i386.rpm
-rw-r--r--   1 root root    37360 Mar 12 19:58 audit-libs-1.0.14-1.EL4.i386.rpm
-rw-r--r--   1 root root   642863 Mar 12 19:58 autoconf-2.59-5.noarch.rpm
-rw-r--r--   1 root root   452212 Mar 12 19:58 automake-1.9.2-3.noarch.rpm
-rw-r--r--   1 root root     2777 Mar 12 19:58 basesystem-8.0-4.noarch.rpm
-rw-r--r--   1 root root  1824030 Mar 12 19:58 bash-3.0-19.3.i386.rpm
-rw-r--r--   1 root root   102595 Mar 12 19:58 bc-1.06-17.1.i386.rpm
-rw-r--r--   1 root root    69834 Mar 12 19:58 beecrypt-3.1.0-6.i386.rpm
-rw-r--r--   1 root root  2988135 Mar 12 19:58 binutils-2.15.92.0.2-21.i386.rpm
-rw-r--r--   1 root root    48570 Mar 12 19:58 bzip2-1.0.2-13.EL4.3.i386.rpm
-rw-r--r--   1 root root    35981 Mar 12 19:58 bzip2-devel-1.0.2-13.EL4.3.i386.rpm
-rw-r--r--   1 root root    34242 Mar 12 19:58 bzip2-libs-1.0.2-13.EL4.3.i386.rpm
-rw-r--r--   1 root root    85726 Mar 12 19:58 checkpolicy-1.17.5-1.i386.rpm
-rw-r--r--   1 root root   131894 Mar 12 19:58 chkconfig-1.3.13.4-1.i386.rpm
--More--
```

In this directory, there are 253 RPM files that come to approximately 347 MB of files. These were installing into the x86World that is in /i386. To see the x86World, you can list the files in the directory where you installed the x86World. As the root user, run:

```
ls /i386
```

You should see the list of directories shown in Example 3-7.

*Example 3-7 /i386 directories*

```
[root@ivmb4]# ls
bin   dev  home    lib    mnt  proc  sbin     srv  tmp  var
boot  etc  initrd  media  opt  root  selinux  sys  usr
```

You can see this looks like a normal Linux root directory and these directories are what you will see when running in PowerVM Lx86 (also called *x86 mode*). Just to be sure, list the *real* root directory of your Linux for Power server. Run:

```
ls / .
```

You should find something similar to the list of directories shown in Example 3-8.

*Example 3-8 / directories*

```
[root@ivmb4]# ls /
bin   dev  home  initrd  lib64       media  mnt  proc  sbin     srv  tmp  var
boot  etc  i386  lib     lost+found  misc   opt  root  selinux  sys  usr
```

Now, you can start PowerVM Lx86. Run this command:

```
runx86
```

Alternatively, if /usr/local/bin is not in your PATH use the full path name:

```
/usr/local/bin/runx86
```

The prompt changes to something similar to `bash-3.00 #`.

This prompt is the default prompt for the root user. If you are running PowerVM Lx86 as a regular user, then the prompt is dependant on the normal shell settings. If you now list the root file system (/), you find it is the Linux for Power /i386 directory as shown in Example 3-9.

*Example 3-9 The root directories within the runx86 environment*

```
bash-3.00# ls /
bin dev home lib mnt proc sbin srv tmp var
boot etc initrd media opt root selinux sys usr
bash-3.00#
```

Next, it is worth checking that this is actually in x86 mode, using the following command as shown in Example 3-10:

```
arch
```

*Example 3-10 arch example*

```
Script started on Tue Apr 22 15:10:08 2008
172_29_139_79:/tmp/pubserver # arch
i686
172_29_139_79:/tmp/pubserver # exit
exit
```

The i686 references tell you that the command thinks it is running on a x86 processor.

**Tip:** If you are unsure whether you are at a Linux for Power prompt or PowerVM Lx86 prompt, you can use the command **`uname -a`** to determine which prompt you are using.

If you try this command at a Linux for Power prompt, you see output as shown in Example 3-11.

*Example 3-11 Linux for Power arch example*

```
Script started on Tue Apr 22 15:10:42 2008
172_29_139_79:/tmp # arch
ppc64
172_29_139_79:/tmp # exit
exit
```

In Example 3-11, the references to ppc64 show that this it is a POWER processor and that you are in the regular Linux for Power environment. For more information about the processor on Linux, you can take a look at the /proc/cpuinfo file. For PowerVM Lx86, we see information similar to Example 3-12.

*Example 3-12 x86 /proc/cpuinfo information*

```
bash-3.00# cat /proc/cpuinfo
processor       : 0
vendor_id       : p-ave
cpu family      : 6
model           : 8
model name      : p-ave
stepping        : 6
cpu MHz         : 801.511
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 mtrr pge mca cmov pat pse36
mmx
bogomips        : 1585.15

processor       : 1
vendor_id       : p-ave
cpu family      : 6
model           : 8
model name      : p-ave
stepping        : 6
cpu MHz         : 801.511
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 mtrr pge mca cmov pat pse36
mmx
bogomips        : 1585.15
```

**Note:** The MHz rating and bogomips rating. are the same as the underlying Power Systems server.

On the regular Linux for Power environment, we get the following much simpler output as shown in Example 3-13.

*Example 3-13 Linux for Power /proc/cpuinfo*

```
[root@l4_rhel4_2 i386]# cat /proc/cpuinfo
processor       : 0
cpu             : POWER5 (gr)
clock           : 1504.352000MHz
revision        : 2.2

processor       : 1
cpu             : POWER5 (gr)
clock           : 1504.352000MHz
revision        : 2.2

timebase        : 188044000
machine         : CHRP IBM,9124-720
```

Now, take a look at the processes that you can see running. Within PowerVM Lx86, Example 3-14 shows the output from the **ps -ef** command.

*Example 3-14 The output from ps -ef example in x86World*

```
bash-3.00# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root      3650  3645  0 03:09 pts/0    00:00:00 ps -ef
root      6396     1  0 Mar12 hvc0     00:00:02 /usr/libexec/gconfd-2 4
root      5032  5031  0 Mar12 hvc0     00:00:01 /bin/bash
root      3645  3644  3 03:08 pts/0    00:00:00 /bin/bash
bash-3.00#
```

This example shows that within PowerVM Lx86, only the processes running in this environment are listed. You can think of these as the processes running inside the PowerVM Lx86 virtual machine. On the regular Linux for Power system, it looks quite different as shown in Example 3-15.

*Example 3-15 The output from ps -ef example in Linux for Power*

```
[root@l4_rhel4_2 i386]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Mar09 ?        00:00:00 init [3]
root         2     1  0 Mar09 ?        00:00:00 [migration/0]
root         3     1  0 Mar09 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 Mar09 ?        00:00:00 [migration/1]
root         5     1  0 Mar09 ?        00:00:00 [ksoftirqd/1]
root         6     1  0 Mar09 ?        00:00:00 [migration/2]
root         7     1  0 Mar09 ?        00:00:00 [ksoftirqd/2]
root         8     1  0 Mar09 ?        00:00:00 [migration/3]
root         9     1  0 Mar09 ?        00:00:00 [ksoftirqd/3]
root        10     1  0 Mar09 ?        00:00:00 [events/0]
root        11     1  0 Mar09 ?        00:00:00 [events/1]
root        12     1  0 Mar09 ?        00:00:00 [events/2]
root        13     1  0 Mar09 ?        00:00:00 [events/3]
root        14    10  0 Mar09 ?        00:00:00 [khelper]
root        15    10  0 Mar09 ?        00:00:00 [kblockd/0]
root        16    10  0 Mar09 ?        00:00:00 [kblockd/1]
```

```
root        17    10  0 Mar09 ?        00:00:00 [kblockd/2]
root        18    10  0 Mar09 ?        00:00:00 [kblockd/3]
root        19     1  0 Mar09 ?        00:00:00 [khubd]
root        48     1  0 Mar09 ?        00:00:00 [rtasd]
root        53    10  0 Mar09 ?        00:00:00 [aio/0]
root        51     1  0 Mar09 ?        00:00:01 [kswapd1]
root        52     1  0 Mar09 ?        00:00:01 [kswapd0]
root        54    10  0 Mar09 ?        00:00:00 [aio/1]
root        55    10  0 Mar09 ?        00:00:00 [aio/2]
root        56    10  0 Mar09 ?        00:00:00 [aio/3]
root       211    10  0 Mar09 ?        00:00:00 [khvcd]
root       214     1  0 Mar09 ?        00:00:00 [kseriod]
root       320     1  0 Mar09 ?        00:00:00 [scsi_eh_0]
root       328     1  0 Mar09 ?        00:00:00 [scsi_eh_1]
root       335     1  0 Mar09 ?        00:00:00 [scsi_eh_2]
root       354     1  0 Mar09 ?        00:00:02 [kjournald]
root      1209     1  0 Mar09 ?        00:00:00 udevd
root      1236    12  0 Mar09 ?        00:00:00 [kauditd]
root      1285    13  0 Mar09 ?        00:00:00 [kmirrord]
root      1307     1  0 Mar09 ?        00:00:00 [kjournald]
root      1308     1  0 Mar09 ?        00:00:04 [kjournald]
root      1309     1  0 Mar09 ?        00:00:00 [kjournald]
root      1310     1  0 Mar09 ?        00:00:24 [kjournald]
root      1311     1  0 Mar09 ?        00:00:01 [kjournald]
root      1727     1  0 Mar09 ?        00:00:03 /sbin/dhclient -1 -q -lf
/var/lib/dhcp/dhclient-eth0.leases -pf /var/run/dhclient-eth0.pid eth0
root      1772     1  0 Mar09 ?        00:00:00 syslogd -m 0
root      1776     1  0 Mar09 ?        00:00:00 klogd -x
rpc       1788     1  0 Mar09 ?        00:00:00 portmap
root      1808     1  0 Mar09 ?        00:00:00 rpc.statd
root      1841     1  0 Mar09 ?        00:00:00 rpc.idmapd
root      2010     1  0 Mar09 ?        00:02:18 /sbin/iprinit --daemon
root      2021     1  0 Mar09 ?        00:02:18 /sbin/iprupdate --daemon
root      2035     1  0 Mar09 ?        00:00:17 /sbin/iprdump
root      2120     1  0 Mar09 ?        00:00:00 /usr/sbin/sshd
root      2154     1  0 Mar09 ?        00:00:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
root      2174     1  0 Mar09 ?        00:00:00 sendmail: accepting connections
smmsp     2182     1  0 Mar09 ?        00:00:00 sendmail: Queue runner@01:00:00 for
/var/spool/clientmqueue
root      2212     1  0 Mar09 ?        00:00:00 gpm -m /dev/input/mice -t exps2
root      2366     1  0 Mar09 ?        00:00:00 crond
xfs       2396     1  0 Mar09 ?        00:00:00 xfs -droppriv -daemon
root      2415     1  0 Mar09 ?        00:00:00 /usr/sbin/atd
dbus      2454     1  0 Mar09 ?        00:00:00 dbus-daemon-1 --system
root      2467     1  0 Mar09 ?        00:00:00 cups-config-daemon
root      2478     1  0 Mar09 ?        00:00:00 hald
root      2487     1  0 Mar09 ?        00:00:00 login -- root
root     19399     1  0 Mar11 ?        00:00:00 cupsd
root      1921  2487  0 Mar12 hvc0     00:00:00 -bash
root      1969     1  0 Mar12 ?        00:00:24 [rpciod]
root      1970     1  0 Mar12 ?        00:00:00 [lockd]
root      3096    13  0 Mar12 ?        00:00:01 [pdflush]
root      3098    11  0 Mar12 ?        00:00:07 [pdflush]
daemon    4346     1  0 Mar12 ?        00:00:00 /opt/p-ave/bin/powerx86daemon
```

```
root      4796     1  0 Mar12 hvc0     00:19:22 Xvnc :1 -desktop
l4_sles9_2.itsc.austin.ibm.com:1 (root) -httpd /usr/share/vnc/classes -auth /root/.Xauthority
-geometry 1024x768 -depth 16 -rfbwait 30000
root      4800     1  0 Mar12 hvc0     00:00:29 vncconfig -iconic
root      4801     1  0 Mar12 hvc0     00:03:24 xterm -geometry 80x24+10+10 -ls -title
l4_sles9_2.itsc.austin.ibm.com:1 (root) Desktop
root      4802     1  0 Mar12 hvc0     00:00:00 /bin/sh /usr/bin/startkde
root      4828  4801  0 Mar12 pts/1    00:00:00 -bash
root      4876     1  0 Mar12 ?        00:00:00 kdeinit: Running...
root      4879     1  0 Mar12 ?        00:00:00 kdeinit: dcopserver --nosid
root      4881  4876  0 Mar12 ?        00:00:00 kdeinit: klauncher
root      4884     1  0 Mar12 ?        00:00:00 kdeinit: kded
root      4886     1  0 Mar12 ?        00:00:00 /usr/libexec/gam_server
root      4969  4876  0 Mar12 ?        00:00:01 /usr/bin/artsd -F 10 -S 4096 -s 60 -m
artsmessage -c drkonqi -l 3 -f
root      4972     1  0 Mar12 ?        00:00:01 kdeinit: knotify
root      4975  4802  0 Mar12 hvc0     00:00:00 kwrapper ksmserver
root      4977     1  0 Mar12 ?        00:00:00 kdeinit: ksmserver
root      4978  4876  0 Mar12 ?        00:00:01 kdeinit: kwin
root      4980     1  0 Mar12 ?        00:00:02 kdeinit: kdesktop
root      4986     1  0 Mar12 ?        00:00:02 kdeinit: kicker
root      4987  4876  0 Mar12 ?        00:00:00 kdeinit: kio_file file
/tmp/ksocket-root/klauncherSxGUjc.slave-socket /tmp/ksocket-root/kdesktop7xUijc.slave-socket
root      4995     1  0 Mar12 ?        00:00:00 kdeinit: khotkeys
root      4997  4876  0 Mar12 ?        00:00:00 pam-panel-icon
root      4998  4876  0 Mar12 ?        00:00:25 /usr/bin/python /usr/bin/rhn-applet-gui
root      5000  4997  0 Mar12 ?        00:00:00 /sbin/pam_timestamp_check -d root
root      5002     1  0 Mar12 ?        00:00:00 /usr/libexec/gconfd-2 14
root      5031  4828  0 Mar12 hvc0     00:00:00 /bin/bash /usr/local/bin/runx86
root      5032  5031  0 Mar12 hvc0     00:00:01 /opt/p-ave/bin/p-ave -z /etc/opt/p-ave/config
/i386/i386/bin/bash
root      6396     1  0 Mar12 hvc0     00:00:02 /opt/p-ave/bin/p-ave -z /etc/opt/p-ave/config
-f3ff -argv0 /usr/libexec/gconfd-2 /i386/i386/usr/libexec/gconfd-2 4
root      6630  4828  0 Mar12 pts/1    00:14:27 top
root      7607  4980  0 Mar13 ?        00:00:00 /usr/bin/kdesktop_lock
root      7608  7607  0 Mar13 ?        00:00:01 kblankscrn.kss -window-id 29360139
root      3274  2120  0 02:32 ?        00:00:00 sshd: root@pts/0
root      3276  3274  0 02:32 pts/0    00:00:00 -bash
root      3737  3276  0 03:24 pts/0    00:00:00 ps -ef
```

Ignoring the regular Linux daemons and processes, Example 3-15 on page 27 shows

- VNC (process 4796), which is running an xterm (process 4801) and in turn running a bash shell (process 4828)
- The shell has started the runx86 process through a further bash shell (process 5031)

If you have started the **runx86** command many times to run different applications or different users, then you will have many more such processes running. The process 6396 is running a daemon that PowerVM Lx86 uses. See the *Administration Guide* for more details. You also will see other processes that are running in PowerVM Lx86.

Within PowerVM Lx86, let us check the other Linux commands. You can run x86 binaries with the **which** and **file** commands as shown in Example 3-16. So, within the PowerVM Lx86, you use the x86 version of the vi editor.

*Example 3-16 x86World which command*

```
bash-3.00# which vi
/bin/vi
bash-3.00# file /bin/vi
/bin/vi: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux
2.2.5, dynamically linked (uses shared libs), stripped
bash-3.00#
```

## 3.5 What to do if you have issues

The *Release Notes* that come as part of the PowerVM Lx86 package includes details of what to do if you have a failure. There is an e-mail address to report the issue. Please also provide the log files that are created and a clear description of the issue that you are having.

4

# Consolidating Linux workloads with IBM PowerVM Lx86

This chapter goes into more details about the benefits, from the perspective of the client and the ISV, of using IBM PowerVM Lx86. Because the client has already realized the benefits coming from server consolidation initiatives in reducing their Total Cost of Ownership (TCO) in data center infrastructure, we present an example of a frequently found application environment running in separate x86 servers and how these servers can be consolidated on a single IBM Power Systems. In this example, we show the use of IBM PowerVM Lx86, if applicable, for each of the application migration scenarios.

# 4.1 Applications that are good candidates for PowerVM Lx86

From a workload behavior perspective, applications expected to use IBM PowerVM Lx86 effectively fall into the following categories:

- Transactional: Either enterprise resource planning (ERP), supply chain management (SCM), customer relationship management (CRM), application servers, or other user written applications
- I/O-heavy applications: ERP, SCM, CRM, database, security, systems management, or other user written applications
- User applications from Linux clients or other user interface applications

Alternatively, application performance is not at its best when running applications on top of IBM PowerVM Lx86 if the applications fall into one of these categories:

- High Performance Computing (HPC) or other floating point-intensive computation. This can include some graphics and user interface applications as well as traditional HPC applications.
- Applications that use more than 400 threads (at the time that this book was written).
- Applications that are heavily dependent on shell scripts that affect the user.

**Note:** If your application falls into one of the two last categories listed here, you should consider the native porting for Linux for Power.

# 4.2 When IBM PowerVM Lx86 is helpful

This section discusses situations in which IBM PowerVM Lx86 is a good alternative.

## 4.2.1 Fast startup

If you are the developer of an application that is not ported to native Linux for Power yet and if you want to have that application running in this advanced environment as soon as possible, consider IBM PowerVM Lx86 because it can help in the following aspects:

- No time spent in changing source code and compiling issues
- No need to walk through the code looking for little endian and big endian issues such as bitwise operations

**Note:** For more detailed information about little endian and big endian issues, see 2.3, “Little endian and big endian” on page 12.

## 4.2.2 An existing application with no source code available

If you simply do not have the original Linux on x86 source code for your application, then IBM PowerVM Lx86 is a good alternative for have it running on top of Linux for Power.

### 4.2.3 Maintaining a single source code

Maintaining a single source code is a very cost effective way of keeping your development and maintenance costs on a reasonable level. With IBM PowerVM Lx86, you can maintain a single source code because there is no change of the original source code.

In fact you can go further in this field. You can compile programs on PowerVM Lx86 to the x86 platform using the GNU C compiler and then run them on native Linux on x86. This was achieved in our test environment with several C programs that ran perfectly within PowerVM Lx86 and also when transferred to the x86 system. Thus, you can use PowerVM Lx86 as a development environment, which might save you from purchasing and setting up a x86 server just to compile a few programs.

Just as an example of how feasible this backward porting capability is, Example 4-1 shows the compiling of the GNU C compiler source code inside IBM PowerVM Lx86. We used this binary (the GNU C compiler itself) successfully for compiling codes on a native x86 server.

*Example 4-1 Successful compiling of GNU gcc on PowerVM Lx86*

```
     fi; \
    fi; \
  done; \
fi
make[3]: Leaving directory `/tmp/gcc-4.1.0/i686-pc-linux-gnu/libiberty'
make[2]: Leaving directory `/tmp/gcc-4.1.0/i686-pc-linux-gnu/libiberty'
make[1]: Leaving directory `/tmp/gcc-4.1.0'
real    100m40.521s
user    79m58.406s
sys     8m18.321s
x86 linux >
```

## 4.3 Server consolidation and IBM PowerVM Lx86

IBM PowerVM Lx86 could be very useful in server consolidation efforts as it increases the portfolio of applications that are capable of being consolidated on a single server and, by doing so, exploiting the benefits of the advanced virtualization capabilities and reliability, availability, and serviceability (RAS) features of IBM Power Systems architecture.

From the perspective of the underlying Linux for Power LPAR, PowerVM Lx86 is *just another application running* and all the features that Linux for Power supports, such as fractional CPU, virtual LAN, virtual SCSI, and the dynamic movements of these resources between LPARs, are available.

Figure 4-1 on page 34 shows RAS aspects for the following environments:

- Linux on Power Systems (Linux for Power)
- Linux on x86 systems (Linux on x86)
- Linux on x86 virtual environment for Power Systems (Linux on PowerVM Lx86)

| Reliability, Availability and Serviceability features | Linux on System p | Linux on x86 system | x86 Linux on PowerVM Lx86 | Comments |
|---|---|---|---|---|
| Automatic First-Failure Data Capture and diagnostic fault isolation capabilities | Yes | No | Yes | Used by Error Log Analysis Tool |
| Self-healing internal POWER5 processor array redundancy | Yes | No | Yes | ECC, bit steering, memory scrubbing, etc |
| Industry-first PCI bus parity error recovery | Limited | No | Limited | EEH detection: partition down vs system |
| Scrubbing and redundant bit-steering for self-healing in main storage | Yes | Limited | Yes | Linux on x86 not as robust |
| ECC and Chipkill correction in main storage | Yes | Yes | Yes | |
| Fault tolerance with N+1 redundancy, dual line cords, and concurrent maintenance for power/cooling | Yes | Yes | Yes | |
| Predictive failure analysis on processors, caches, memory, I/O and DASD | Yes | Limited | Yes | Intel does not have predictive analysis of I/O |
| Processor run-time and boot-time de-allocation based on run-time errors (Dynamic Processor De-allocation and Persistent Processor De-allocation) | Yes | No | Yes | FFDC advantage |
| Fault avoidance through highly reliable component selection, component minimization and error mitigation technology internal to chips | Yes | No | Yes | |
| Concurrent run-time diagnostics based on First-Failure Data Capture for power, cooling, and I/O | Limited | No | Limited | Planned for Linux |
| Service Processor is a separate, independent processor that provides hardware initialization during system IPL, operation monitoring of environmental and error events | Yes | Limited | Yes | Linux on Intel not as robust |

*Figure 4-1 RAS features comparison*

As an example of server consolidation using IBM PowerVM Lx86, consider that you want to consolidate on a single server the several workloads currently running on separate different x86 and POWER processor-based servers, as illustrated in Figure 4-2 on page 35. On three separate x86 servers running on Linux:

- Apache on SUSE Linux Enterprise Server 10
- IBM DB2® Database on Red Hat Enterprise Linux 4 AS Update 4
- IBM Tivoli Storage Manager on SUSE Linux Enterprise Server 9 SP3

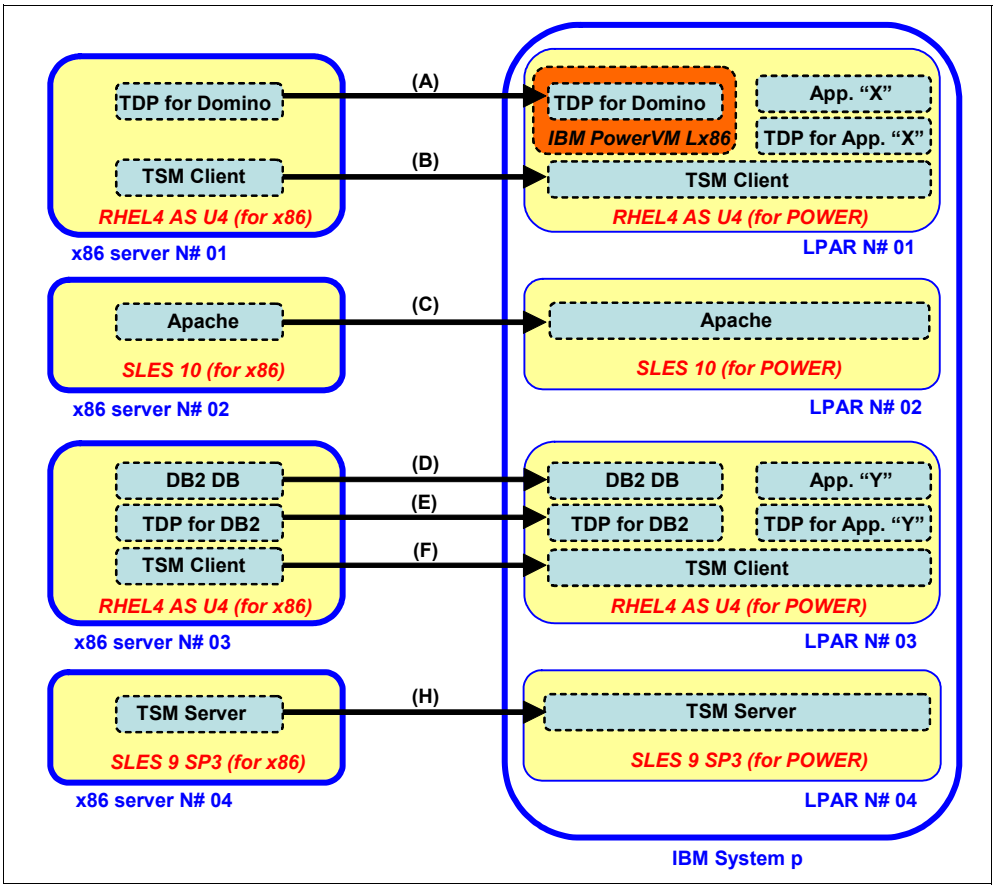



*Figure 4-2 x86 Linux applications server consolidation*

As shown in Figure 4-2, four logical partitions were created on the target server to consolidate the four stand-alone systems. This provides for each application a confined environment with its own operating system image. By doing so, any specific operating system tuning, software maintenance, and even shutdown and reboots on a LPAR can be done without interfering on the other LPARs. For each application migration shown on Figure 4-2, there is a corresponding letter (A, B, C, and so forth) that will be used to point out some specific considerations.

The first comment regarding this example environment is that DB2 is a backup client of a Tivoli Storage Manager Server, and this requires an extra application running on each client partition called Tivoli Storage Manager Client. To deal with the specific requirements and backup tools of each client application an extra Tivoli utility should be installed on each client partition called Tivoli Data Protection for *<the specific backup client application>*.

Let us discuss some specific aspects of each application migration. For the purpose of this example the system is called the *target system*. The steps of the migration are as follows:

1. Migration of x86 Server N# 01:
   - **Migration (A)**: The same applies for Tivoli Data Protection for Domino®.
   - **Migration (B)**: For the Tivoli Storage Manager Client, although we have two options of installation (native Linux on Power Systems servers or PowerVM Lx86), you should choose the native Linux for Power for performance reasons and because it opens the opportunity of having the Tivoli Storage Manager client being used with other native LoP applications on this LPAR, such as Application *X* in Figure 4-2.

2. Migration of x86 Server N# 02:
   - **Migration (C)**: For Apache, again you have two options, but the decision of native installation on Linux for Power comes mainly for performance reasons. In case there are some Apache modules not ported yet to Linux for Power, you still have a way of consolidating this application on the target system by using IBM PowerVM Lx86.
3. Migration of x86 Server N# 03:
   - **Migration (D)**: Again there are two options here, but as there are versions of DB2 for Linux for Power we should consider this one for performance reasons. For large terabyte database migrations, the time factor is crucial sometimes. So, in these situations the use of IBM PowerVM Lx86 is a good alternative as there are no migration activities to be done from the application side. You just have to install the database applications (binaries) and then connect the original database file system (let us say *ext3* format) on the target system. There is no little endian and big endian issue to be addressed.
   - **Migration (E)**: Just like the application (DB2), the Tivoli Data Protection for Domino should be installed on native Linux for Power.
   - **Migration (F)**: The same applies to Tivoli Storage Manager Client. As on Migration C, by installing Tivoli Storage Manager Client on Linux for Power allows further client applications to take the benefits of the Tivoli Storage Manager backup infrastructure.
4. Migration of x86 Server N# 04:
   - **Migration (G)**: The native Linux for Power environment for performance reasons is obviously the better choice. As it will be running on a LPAR with dynamic LPAR capabilities, when there is no backup going on, you can move the amount of resources originally allocated for Tivoli Storage Manager server to another resource constrained LPAR, for example. Another interesting option is that you can use the Virtual Ethernet capability of the POWER Hypervisor™ for transferring large amounts of data from the client LPARs. With such kind of virtual device you can achieve higher transfer rates when compared to native Gigabit Ethernet adapters, although you have to be aware of the additional cpu resource consumption required to achieve such higher throughputs. For more details about the virtual environment capabilities of the Power Systems product line please see *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration*, SG24-7940.

**Note:** The time reducing facility that is introduced by using IBM PowerVM Lx86 in large database migrations that come from Linux on x86 environments also applies to other database products such as Oracle and MySQL.

5

# PowerVM Lx86 example workloads

This chapter presents several practical examples of application usage for each recommended type of workload that are considered as good candidates for IBM PowerVM Lx86. In addition, although benchmarking these applications is not the purpose of this paper, some performance data was captured using the **nmon** tool to show the computational resources of the Power Systems hardware, according to the specific requirements of each kind of application workload.

## 5.1 Setting up the system environment in our lab

The server used for the examples in this paper was an IBM System p5® 550Q Express, configured with Power5+ 1.65GHz processors and 32 GB of memory. Figure 5-1 shows the physical aspects of this model.

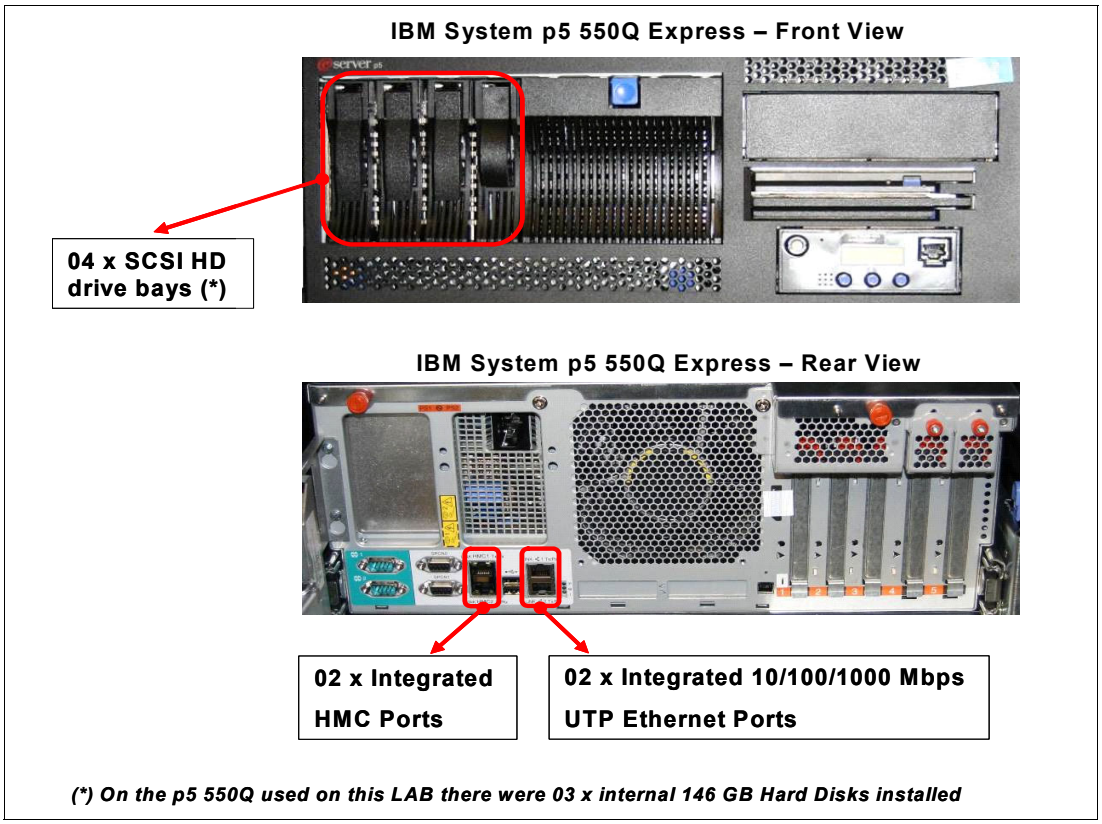



*Figure 5-1 The IBM System p5 550Q Express model*

It is not the purpose of this paper to go into further detail of how to set up LPARs on a Power Systems virtual environment. However, you can find information about this subject in *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration*, SG24-7940.

Assuming that you have all the required software images (VIOS and Linux), after you have the system turned on and all the wiring done (basically networking and storage connections), it would take no more than four hours to set up an environment equivalent to the one presented in Figure 5-2 on page 39, including Virtual I/O Server (VIOS) and client Linux partitions images installed. This environment was the environment that we used for the examples in this paper.

**Note:** After installed the Linux images on each logical partition, do not forget to download and install the *Service and productivity tools* for Linux for Power systems to allow dynamic (*hot*) LPAR operations, such as CPU addition, removal, and movement between LPARs resources and I/O addition and removal. these tools are available at:

`http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html?fetch=info/LinuxAlerts.html`

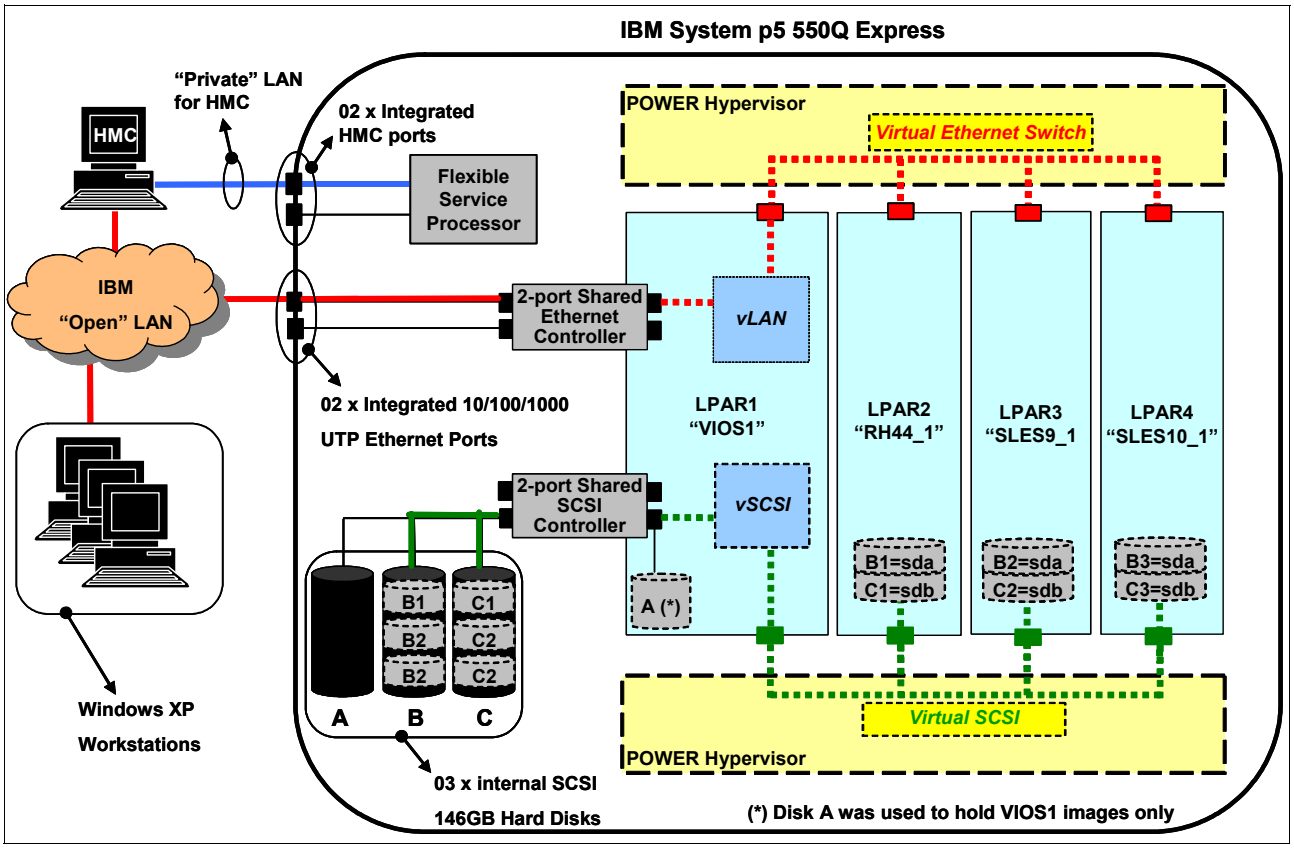



*Figure 5-2 IBM LPAR configuration setup*

The resource allocation configuration of each logical partition was done according to the information presented in Figure 5-3.

| LPAR Name | | VIOS_1 | RH44_1 | | SLES9_1 | | SLES10_1 | |
|---|---|---|---|---|---|---|---|---|
| CPU | Resource | Dedicated Procs | Proc. Units | Virtual Procs | Proc. Units | Virtual Procs | Proc. Units | Virtual Procs |
| | Minimum | 1 | 0.10 | 1 | 0.10 | 1 | 0.10 | 1 |
| | Desired | 1 | 0.80 | 2 | 0.80 | 2 | 0.80 | 2 |
| | Maximum | 1 | 8.00 | 8 | 8.00 | 8 | 8.00 | 8 |
| Memory (MB) | Minimum | 512 | 1024 | | 1024 | | 1024 | |
| | Desired | 512 | 4096 | | 4096 | | 4096 | |
| | Maximum | 512 | 32768 | | 32768 | | 32768 | |
| I/O | Boot device | 1 x Physical Disk (146GB) | 1 x Virtual Disk (sda=10GB) | | 1 x Virtual Disk (sda=10GB) | | 1 x Virtual Disk (sda=10GB) | |
| | x86 world | N/A | 1 x Virtual Disk (sdb=10GB) | | 1 x Virtual Disk (sdb=10GB) | | 1 x Virtual Disk (sdb=10GB) | |
| | LAN adapter | 1 x Physical GbEth. Adapter | 1 x Virtual GbEth. Adapter | | 1 x Virtual GbEth. Adapter | | 1 x Virtual GbEth. Adapter | |

*Figure 5-3 LPAR resource allocation*

**Note:** Remember, after you set up the environment, you need to install the IBM PowerVM Lx86 on top of the already installed and configured Linux for Power LPARs. We explain how to do so in 3.1, "The installation files included in the package" on page 14.

6

# ISV considerations

This chapter discusses topics and provides information that is important to independent software vendors (ISVs). When you change a paradigm of computing, you need to look at the supporting materials such as licensing and support.

## 6.1 User license agreements

ISVs might need to consider whether changes need to be made to user license agreements to support PowerVM Lx86. The user license agreement is specific to the x86 platform, then it needs to be expanded to include Power System servers, as shown in Figure 6-1.

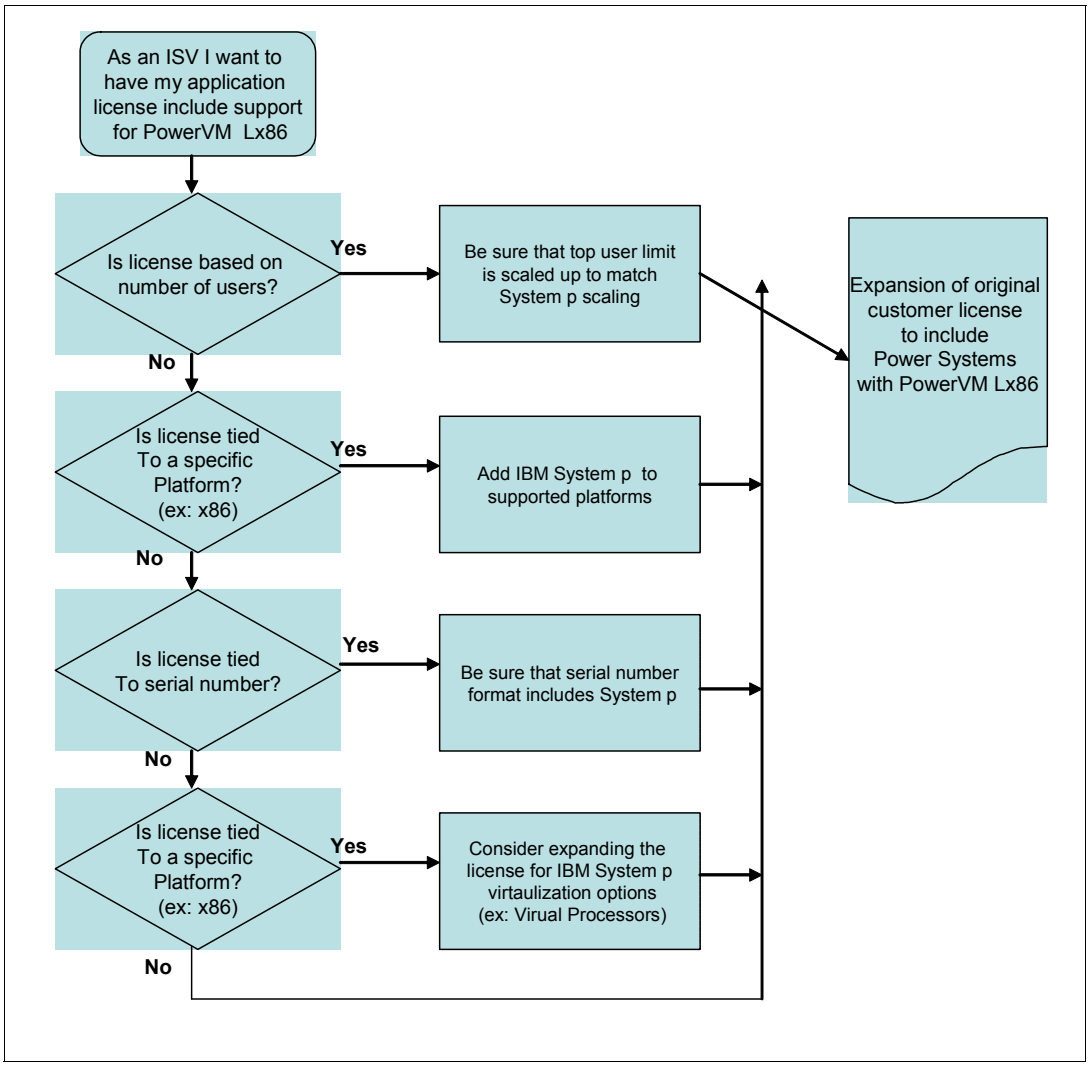



*Figure 6-1 ISV user license agreements*

If you have experience with a translator already and have reviewed the detailed information in Appendix A, “IBM PowerVM Lx86 minimum requirements” on page 45, you might already be confident that your solution will run, which is typically the case.

If there are aspects of your application that you want to test explicitly on the new platforms, there is an offering for ISVs in the Virtual Loaner Program (VLP) where you can access PowerVM Lx86 using a special promotional code. Log in to the VLP site at:

`http://www-304.ibm.com/jct09002c/partnerworld/wps/pub/systems/vlp`

Enter the promotional code 69726. You can also download the technology at:

`http://www.ibm.com/developerworks/linux/lx86/index.html?S_TACT=105AGX03&S_CMP=SPLT`

## 6.2 ISV support

IBM supports the PowerVM Lx86 product as part of the PowerVM offerings. You need to accept the license agreement when you use it or download it from the developerWorks® site link. PowerVM, a family of IBM virtualization technologies, capabilities, and offerings is designed to increase server utilization dramatically by providing innovative capabilities that enable one server to act like many, while giving users the ability to move more processing power to critical applications automatically when needed. Support for PowerVM Lx86 is acquired with PowerVM support.

In addition to regular IBM support, there is a forum that is available at:

http://www-128.ibm.com/developerworks/forums/dw_forum.jsp?forum=1058&cat=72

A

# IBM PowerVM Lx86 minimum requirements

This appendix lists the minimum requirements to install IBM PowerVM Lx86 at the time that we installed the code.

# Minimum requirements

These were the minimum requirements for having IBM PowerVM Lx86 installed and running with full support from IBM:

- Hardware requirements
  - IBM Power Systems hardware based on Power PC, POWER5/5+ and the recently announced POWER6 including POWER technology-based blades
- Operating System requirements for the host system (Linux for Power)
  - Red Hat Enterprise Linux 4 Update 4
  - Novell SuSE Linux Enterprise Server 9 Service Pack 3
  - Novell SuSE Linux Enterprise Server 10
- IBM PowerVM Lx86 requirements:
  - PowerVM Lx86 version 1
  - The Linux x86 (RPMS) distribution equivalent to the host system (Linux for Power)
  - x86 virtual environment compatibility

  PowerVM Lx86 runs most x86 Linux applications except those that has one or more of the characteristics:
  - Directly access hardware (for example, 3D graphics adapters)
  - Are hardware architecture specific
  - Provide unique kernel modules not available for Power Systems
  - Use instructions outside of the Intel® IA-32 instruction set architecture SSE2 as defined by the 1997 Intel Architecture Software Developer's Manual consisting of Basic Architecture (Order Number 243190), Instruction Set Reference Manual (Order Number 243191), and the System Programming Guide (Order Number 243192) all dated 1997
  - Are built with Red Hat or Novell Linux versions earlier than versions Red Hat Enterprise Linux 4 U4 or SUSE Linux Enterprise Server 9 SP3 and do not run correctly on Red Hat Enterprise Linux 4 U4, SUSE Linux Enterprise Server 9 SP3, or SUSE Linux Enterprise Server 10.
  - Require x86 real-mode.

**Note:** For more details about requirements, read Chapter 3 in *IBM PowerVM Lx86 Administration Guide*.

# Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this paper.

## Other publications

These publications, which are available when you install PowerVM Lx86, are also relevant as further information sources:

- *PowerVM Lx86 Administration Guide*
- *PowerVM Lx86 Release Notes*

## Online resources

- Linux for Power service and productivity tools

  http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html?fetch=info/LinuxAlerts.html
- Virtual Loaner Program from IBM

  http://www-304.ibm.com/jct09002c/partnerworld/wps/pub/systems/vlp
- PowerVM Lx86 developerWorks technology site

  http://www.ibm.com/developerworks/linux/lx86/index.html?S_TACT=105AGX03&S_CMP=SPLT
- IBM support Power Systems forum

  http://www-128.ibm.com/developerworks/forums/dw_forum.jsp?forum=1058&cat=72

## How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

**ibm.com**/redbooks

## Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

# Getting Started with PowerVM Lx86


**Runs x86 Linux binaries without porting**

**Allows you to consolidate your x86 and Power Systems servers**

**Provides greater access to software solutions from ISVs**

IBM PowerVM Lx86 brings new benefits to clients and application providers who want the reliability and flexibility of IBM Power Systems in their Linux environment yet use applications that have not yet been ported to the platform. IBM PowerVM Lx86 allows most Linux x86 binaries to run unchanged on IBM POWER processor-based systems using translation software that translates x86 instructions dynamically to POWER instructions and transforms x86 Linux system calls into calls to the POWER Linux kernel. It helps the consolidation effort by allowing Linux x86 applications to be consolidated onto Linux Power partitions along side AIX partitions.

This IBM Redpaper publication can help you understand how to optimize your IT environment by running x86 Linux applications within a Linux for Power environment. If you are running Linux for Power environments and x86 Linux environments, you can lower total cost of ownership by consolidating servers using PowerVM virtualization. This paper shows the planning and installation choices as well as hints and tips to bring the most value to the IBM PowerVM Lx86 solution.

This paper is intended for independent software vendors (ISVs) and their clients to assist them in understanding and installing IBM PowerVM Lx86. Managers can use this paper as they set server strategies. System administrators can also use this paper as they install IBM PowerVM Lx86 with applications.

REDP-4298-01

**INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

**BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**
**ibm.com**/redbooks