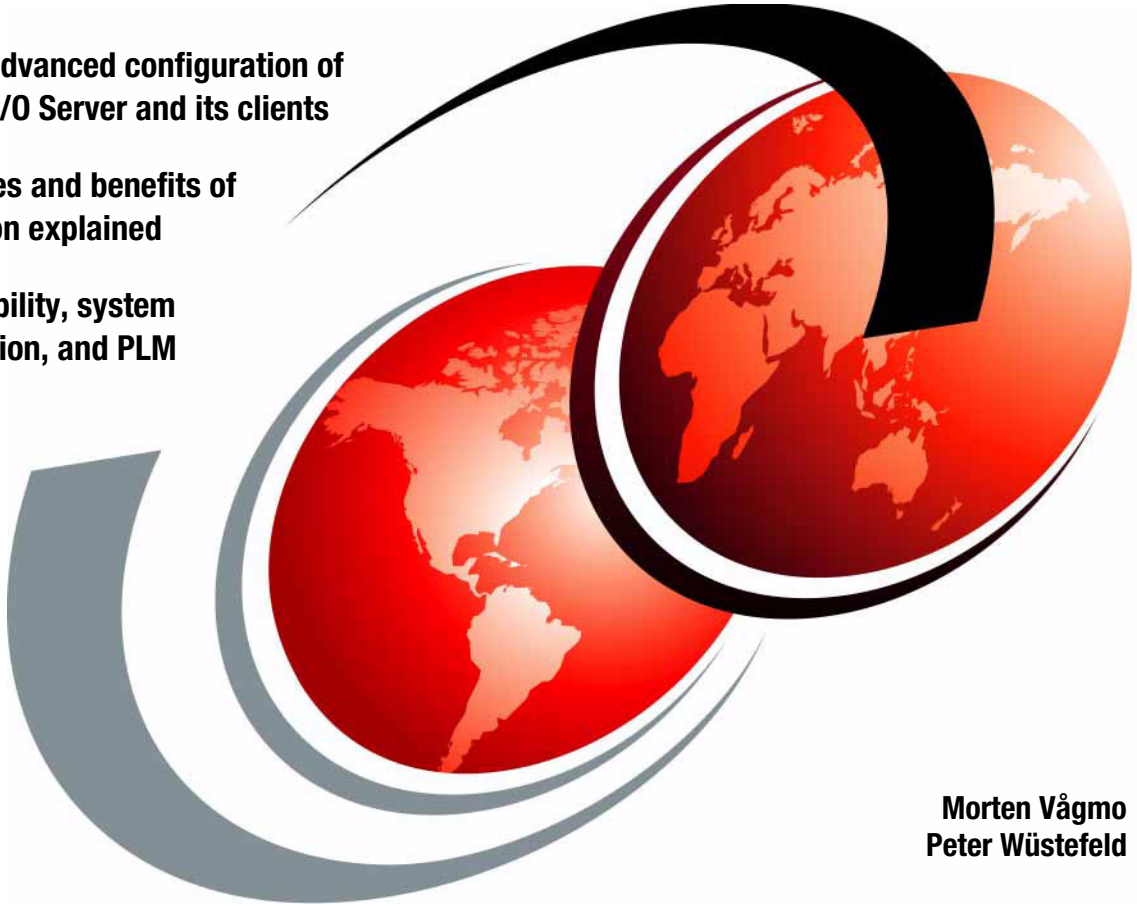
IBM

Advanced POWER Virtualization on IBM System p5:

Introduction and Configuration

- Basic and advanced configuration of the Virtual I/O Server and its clients
- New features and benefits of virtualization explained
- High availability, system administration, and PLM



Morten Vågmo
Peter Wüstefeld

Redbooks

ibm.com/redbooks

International Technical Support Organization

Advanced POWER Virtualization on IBM System p5: Introduction and Configuration

February 2007

SG24-7940-02

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

Third Edition (February 2007)

This edition applies to IBM AIX 5L Version 5.3, HMC Version 5 Release 2.1, Virtual I/O Server Version 1.3 running on IBM System p5 and IBM eServer p5 systems.

Contents

Figures

Tables

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

AIX 5L™
AIX®
AS/400®
Domino®
DB2®
DS4000™
DS6000™
DS8000™
Enterprise Storage Server®
Everyplace®
General Parallel File System™
Geographically Dispersed Parallel Sysplex™
GDPS®
GPFS™
HiperSockets™
HACMP™
i5/OS®
IBM®
iSeries™
LoadLeveler®
Lotus®
Micro-Partitioning™
MQSeries®
OpenPower™
Parallel Sysplex®
Passport Advantage®
PowerPC®
POWER™
POWER Hypervisor™
POWER4™
POWER5™
POWER5+™
PR/SM™
pSeries®
PTX®
Redbooks™
Redbooks (logo) ™
System i™
System p™
System p5™
System z™
System z9™
System Storage™
Tivoli®
TotalStorage®
TXSeries®
Virtualization Engine™
WebSphere®
z/OS®
z/VM®
zSeries®
z9™

The following terms are trademarks of other companies:

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

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

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

This IBM® Redbook provides an introduction to Advanced POWER™ Virtualization on IBM System p5™ servers.

The Advanced POWER Virtualization feature is a combination of hardware and software that supports and manages the virtual I/O environment on POWER5™ and POWER5+™ systems. The main technologies are:

- Virtual Ethernet
- Shared Ethernet Adapter
- Virtual SCSI Server
- Micro-Partitioning™ technology
- Partition Load Manager

The primary benefit of Advanced POWER Virtualization is to increase overall utilization of system resources by allowing only the required amount of processor and I/O resource needed by each partition to be used.

This redbook is also designed to be used as a reference tool for system administrators who manage servers. It provides detailed instructions for:

- Configuring and creating partitions using the HMC
- Installing and configuring the Virtual I/O Server
- Creating virtual resources for partitions
- Installing partitions with virtual resources

While the discussion in this IBM Redbook is focused on IBM System p5 hardware and the AIX® 5L™ operating system (and it also applies to IBM eServer™ p5 systems), the basic concepts can be extended to the i5/OS® and Linux® operating systems as well as the IBM System i™ and platform and OpenPower™ editions.

A basic understanding of logical partitioning is required.

This publication is an update of the IBM Redbook *Advanced POWER Virtualization on IBM System p5*, SG24-7940 that was published December 2005. It contains the enhancements in Virtual I/O Server Version 1.3, as well as expanding on key topics.

The team that wrote this redbook

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

Morten Vågmo is a Consulting IT Specialist in IBM Norway with 18 years of AIX and AIX 5L experience. He is working in technical pre-sale support and is now focusing on Advanced POWER Virtualization implementations. Morten holds a degree in Marine Engineering from the Technical University of Norway.

Peter Wuestefeld M.A. is a Pre-Sales Systems Engineer with IBM Premier Business Partner SVA GmbH in Germany. With twelve years of experience with AIX, he specializes in a wide field of AIX topics. He is also the Product Manager for Linux systems in his company, having twelve years of Linux experience. Peter holds a masters degree in Prehistoric Archaeology from the Eberhard-Karls- University of Tuebingen, Germany.

Authors of the Second Edition (December 2005) of this redbook were:

Annika Blank, **Paul Kiefer**, **Carlos Sallave Jr.**, **Gerardo Valencia**, **Jez Wain**, **Armin M. Warda**

Authors of the First Edition (October 2004) of this redbook were:

Bill Adra, Annika Blank, Mariusz Gieparda, Joachim Haust, Oliver Stadler, Doug Szerdi

The project that produced this document was managed by:

Scott Vetter
IBM U.S.

Thanks to the following people for their contributions to this project:

Bob Kovacs, Jim Pafumi, Jim Partridge, Jorge R Nogueras, Ray Anderson, Vasu Vallabhaneni, Vani Ramagiri
IBM U.S.

Nigel Griffiths, Paul Beedham
IBM UK

Volker Haug
IBM Germany

Guido Somers
IBM Belgium

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners or clients.

Your efforts will help increase product acceptance and client 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 Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@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

Summary of changes

This section describes the technical changes made in this edition of the this publication. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7940-02
for Advanced POWER Virtualization on IBM System p5
as created or updated on July 18, 2007.

January 2007, Third Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

The following is a list of the major enhancements with this edition:

- The Summary of changes section was added after the Preface.
- Chapter 1, “Introduction” on page 1
  - 1.6, “The value of the Advanced POWER Virtualization” on page 16 is a condensed version of the old Chapter 2. Therefore, the following Chapter numbers are a value of 1 less than the last edition.
- Chapter 2, “Virtualization technologies on System p servers” on page 19
  - New features in Version 1.3 of the Virtual I/O Server.
  - Additional information for Virtual SCSI optical devices.
- Chapter 3, “Setting up the Virtual I/O Server: the basics” on page 117
  - General refresh of installation and client dialogs to reflect window changes.
- Chapter 4, “Setting up virtual I/O: advanced” on page 181
  - Added 4.5, “Scenario 4: Network Interface Backup in the client” on page 234.
- Chapter 5, “System management” on page 257
  - Additional information about Hot Plugging adapters with respect to Fibre Channel.

- Additional information about the **topas** command.
- New monitoring commands included in 5.5.6, “New monitoring commands on the Virtual I/O Server” on page 343.
- A new section on security: 5.7, “Security considerations for Virtual I/O Servers” on page 358.

Changed information

The following is a list of the minor changes throughout this edition:

- General changes to better use the IBM System p5 brand.
- Removal of Chapter 2, now summarized in Chapter 1.
- Removal of Appendix A and Appendix B, two topics that are better covered in other publications.
- Updated screen captures and command output to reflect newer software levels.

1

Introduction

The first edition of the *Advanced POWER Virtualization on IBM System p5*, SG24-7940 was published over two years ago. Since that time, IBM and its partners have gained considerable experience with the technology and have received feedback from clients on how they are using virtualization in the field. This third edition takes the lessons learned over the past year to build on the foundation work of the first version of the redbook. The result is the IBM Redbook in your hands, a comprehensive re-write of the initial work, and an enhancement of the second edition.

This publication targets virtualization novices as well as those that have already experimented with the technology. It is written in a hands-on style to encourage you to *get started*. This latest edition enriches the basic virtualization scenarios of the first and extends them to include some of the more advanced configurations, in particular for improving availability, using the Partition Load Manager, performing administration tasks, and monitoring system resources.

The remainder of this chapter provides a short overview of the key virtualization technologies. Chapter 3, “Setting up the Virtual I/O Server: the basics” on page 117 covers the foundation technologies in detail as well as the new Virtual I/O Server Version 1.3 features. An understanding of this chapter is required for the remainder of the book. Chapter 4, “Setting up virtual I/O: advanced” on page 181 covers initial partition configurations while Chapter 4, “Setting up virtual I/O: advanced” on page 181 looks at more complex setups such as SAN-attached storage and multiple Virtual I/O Servers. Chapter 5, “System

management” on page 257 discusses system administration and the new monitoring tools and the last chapter. Chapter 6, “Partition Load Manager” on page 371 shows how to automate the management of resources using PLM. So read on...

1.1 Virtualization on IBM System p5

The IBM System p5 Virtualization system technologies available on the POWER5 processor-based System p5 servers are described in this section.

1.1.1 POWER Hypervisor

The POWER Hypervisor™ is the foundation for virtualization on a System p5 server. It enables the hardware to be divided into multiple partitions, and ensures strong isolation between them.

Always active on POWER5-based servers, the POWER Hypervisor is responsible for dispatching the logical partition workload across the physical processors. The POWER Hypervisor also enforces partition security, and can provide inter-partition communication that enables the Virtual I/O Server's virtual SCSI and virtual Ethernet function.

1.1.2 Simultaneous multithreading (SMT)

Enhancements in POWER5 processor design allow for improved overall hardware resource utilization. SMT technology allows two separate instruction streams (threads) to run concurrently on the same physical processor, improving overall throughput.

1.1.3 LPAR and shared-processor partitions

A Logical Partition (LPAR) is not constrained to physical processor boundaries, and may be allocated processor resources from a shared processor pool. An LPAR that utilizes processor resources from the shared processor pool is known as a Micro-Partition LPAR.

The percentage of a physical processor that is allocated is known as processor entitlement. Processor entitlement may range from ten percent of a physical processor up to the maximum installed processor capacity of the IBM System p5. Additional processor entitlement may be allocated in increments of one percent of a physical processor.

1.1.4 Dynamic reconfiguration

It is possible to dynamically move system resources, physical processors, virtual processors, memory, and I/O slots, between partitions without rebooting. This is known as dynamic reconfiguration or dynamic LPAR.

1.1.5 Virtual LAN

A function of the POWER Hypervisor, Virtual LAN allows secure communication between logical partitions without the need for a physical I/O adapter. The ability to securely share Ethernet bandwidth across multiple partitions increases hardware utilization.

1.1.6 Virtual I/O

Virtual I/O provides the capability for a single physical I/O adapter and disk to be used by multiple logical partitions of the same server, allowing consolidation of I/O resources and minimizing the number of I/O adapters required.

1.1.7 Capacity Upgrade on Demand

There are multiple Capacity Upgrade on Demand (CUoD) possibilities offered, including:

- Permanent Capacity Upgrade on Demand

  Enables permanent system upgrades by activating processors or memory.

- On/Off Capacity Upgrade on Demand

  Usage based billing that allows for activation and deactivation of both processors and memory as required.

- Reserve Capacity Upgrade on Demand

  Prepaid agreement that adds reserve processor capacity to the shared processor pool, which is used if the base shared pool capacity is exceeded.

- Trial Capacity Upgrade on Demand

  Partial or total activation of installed processors or memory for a fixed period of time.

1.1.8 Multiple operating system support

The POWER5 processor based System p5 products support IBM AIX 5L Version 5.2 ML2, IBM AIX 5L Version 5.3, i5/OS, and Linux distributions from SUSE and Red Hat.

1.1.9 Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) is a hardware management solution that inherits the most basic of Hardware Management Console (HMC) features and removes the requirement of an external HMC. It is limited to managing a single System p5 server. IVM runs on the Virtual I/O Server Version 1.3.

1.2 RAS of virtualized systems

As individual System p5 systems become capable of hosting more system images, the importance of isolating and handling outages that might occur becomes greater. Hardware and operating system functions have been integrated into the system design to monitor system operation, predict where outages may occur, isolate outage conditions that do occur, then handle the outage condition, and when possible, continue operation. IBM reliability, availability, and serviceability (RAS) engineers are constantly improving server design to help ensure that System p5 servers support high levels of concurrent error detection, fault isolation, recovery, and availability.

1.2.1 Reliability, availability, and serviceability

The goal of the RAS is to minimize outages. This section highlights specific RAS capabilities introduced or enhanced in the System p5 products.

Reducing and avoiding outages

The base reliability of a computing system is, at its most fundamental level, dependent upon the intrinsic failure rates of the components that comprise it. Highly reliable servers are built with highly reliable components. The System p5 servers allow for redundancies of several system components and mechanisms that diagnose and handle special situations, errors, or failures at the component level.

System surveillance

Fault detection and isolation are key elements of high availability and serviceable server design. In the System p5 systems, three elements, the POWER Hypervisor, service processor, and Hardware Management Console (HMC), cooperate in detecting, reporting, and managing hardware faults.

Service processor

The service processor enables POWER Hypervisor and Hardware Management Console surveillance, selected remote power control, environmental monitoring, reset and boot features, and remote maintenance and diagnostic activities, including console mirroring. On systems without an HMC, the service processor can place calls to report surveillance failures with the POWER Hypervisor, critical environmental faults, and critical processing faults.

The service processor provides the following services:

- Environmental monitoring
- Mutual surveillance with the POWER Hypervisor
- Self protection for the system to restart it from unrecoverable firmware error, firmware hang, hardware failure, or environmentally induced failure.
- Fault monitoring and operating system notification at system boot

POWER Hypervisor RAS

Elements of the POWER Hypervisor are used to manage the detection and recovery of certain errors. The POWER Hypervisor communicates with both the service processor and the Hardware Management Console.

Predicting failures

IBM has implemented a server design called First Failure Data Capture (FFDC) that builds-in thousands of hardware error check stations that capture and help to identify error conditions within the server and enable System p5 systems to *self-diagnose* and *self-heal*. Each of these checkers is viewed as a *diagnostic probe* into the server, and when coupled with extensive diagnostic firmware routines, allows assessment of hardware error conditions at runtime.

Because of the first failure data capture technology employed in the System p5 systems, the need to recreate diagnostics for CEC failures has been greatly reduced. The service processor working in conjunction with the FFDC technology provides the automatic detection and isolation of errors without having to recreate the failure. This means that errors will be correctly detected and isolated at the time of the failure occurrence.

Improving service

The HMC software includes a wealth of improvements for service and support, including automated install and upgrade, and concurrent maintenance and upgrade for hardware and firmware. The HMC also provides a Service Focal Point for service receiving, logging, tracking system errors and, if enabled, forwarding problem reports to IBM Service and Support organizations.

For the System p5 systems, components such as power supplies, fans, disks, HMCs, PCI adapters, and devices can be repaired while powered on. The HMC supports many new concurrent maintenance functions in System p5 systems.

Firmware on the System p5 systems is planned to be released in a cumulative sequential fix format for concurrent application and activation. The objective is that the majority of firmware updates will be able to be installed and activated without having to power cycle or reboot the system.

For more information about System p5 systems RAS, refer to the whitepaper *IBM System p5: a Highly Available Design for Business-Critical Applications*, available at:

http://www.ibm.com/servers/eserver/pseries/library/wp_lit.html

For information about service and productivity tools for Linux on POWER, refer to:

http://techsupport.services.ibm.com/server/lopdiags

1.2.2 Availability and serviceability in virtualized environments

In partitioned environments where more business-critical applications are consolidated on different operating systems with the same hardware, additional availability and serviceability is needed to ensure a smooth recovery of single failures and allow most of the applications to still be operative when one of the operating systems is out of service. Furthermore, high availability functions at the operating system and application levels are required to allow for quick recovery of service for the end users.

In the System p5 systems, there are several mechanisms that increase overall system and application availability by combining hardware, system design, and clustering.

Dynamic processor deallocation and sparing

The POWER Hypervisor will deallocate failing processors and replace them with processors from unused CUoD capacity, if available.

Memory sparing

If, during power-on, the service processor identifies faulty memory in a server that includes CUoD memory, the POWER Hypervisor will replace the memory with unused memory in the CUoD pool. If there is no spare memory, the POWER Hypervisor will reduce the capacity of one or more partitions. On the IBM System p5 Models 590 and 595, memory sparing will substitute unused CUoD memory for all general memory card failures. On the IBM System p5 Model 570, memory sparing will substitute CUoD memory for up to one memory card failure.

Check with your IBM representative for the availability of this function on IBM System p5 servers.

Adapter sparing

Adapter sparing can be achieved by maintaining a set of recovery PCI adapters as global spares for use in DR operations in the event of adapter failure. Include in the configuration the recovery PCI adapters in different partitions of the system, including Virtual I/O Servers so that they are configured and ready for use.

Redundant Virtual I/O Servers

Since an AIX 5L or Linux partition can be a client of one or more Virtual I/O Servers at a time, a good strategy to improve availability for sets of AIX 5L or Linux client partitions is to connect them to two Virtual I/O Servers. One key reason for redundancy is the ability to upgrade to the latest Virtual I/O Server technologies without affecting production workloads. This technique provides a redundant configuration for each of the connections of the client partitions to the external Ethernet network or storage resources. This IBM Redbook discusses these configurations.

Figure 1-1 shows some of the mechanisms used to improve the availability and serviceability of a partitioned System p5 server.

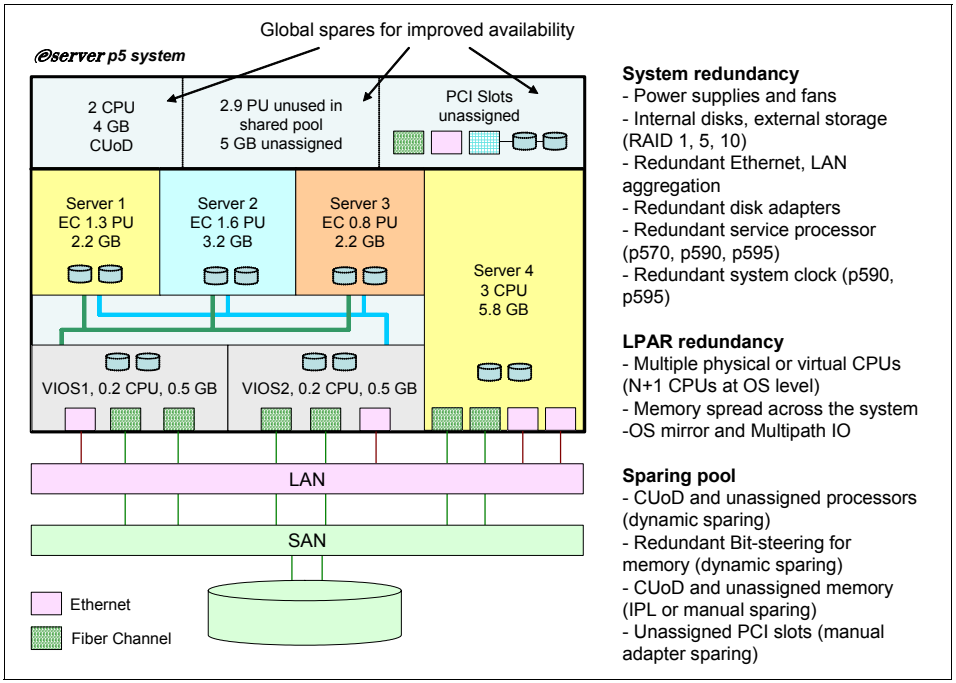


Figure 1-1 Redundant components in a virtualized system

1.3 Security in a virtualized environment

Architectural enhancements made in the POWER5 system platform design make cross-partition sharing and communication possible. The new virtualization functions maintain system security requirements. The cross-partition features do not introduce any security exposure beyond what is implied by the function. For example, a virtual LAN connection would have the same security considerations as a physical network connection.

1.4 Operating system support

The System p5 systems are designed to support a wide variety of operating system types and versions to allow clients to consolidate heterogeneous environments. Table 1-1 lists the supported operating systems with Advanced POWER Virtualization (APV).

Table 1-1 Operating systems supported in a virtualized System p5 system

Operating system	Client of APV functions	Server for APV functions
IBM AIX 5L V5.2 ML2 or later		
IBM AIX 5L V5.3	X	
VIOS V1.1, V1.2, and V1.3		X
IBM i5/OS	X	
RHEL AS V3 update 3 or later	X	X[a,b]
RHEL AS V4	X	Xa,b
Novell SLES V9 SP1	X	X[b]

a. Not for virtual SCSI
b. Not for clients running AIX 5L

Important: The Virtual I/O Server does not run end-user applications.

AIX 5L Version 5.3 partitions can use physical and virtual resources at the same time.

1.4.1 IBM AIX 5L for System p5 systems

AIX 5L is supported on the System p5 servers in partitions with dedicated processors similar to IBM eServer pSeries® systems with POWER4™ processors, but System p5 servers do not support affinity partitioning. For System p5 systems configured with the Advanced POWER Virtualization feature (APV), AIX 5L Version 5.3 or later is required for shared-processor partitions, virtual I/O, and virtual Ethernet.

On System p5, mixed environments of AIX 5L V5.2 ML2 and AIX 5L V5.3 partitions with dedicated processors and adapters, and of AIX 5L V5.3 partitions using Micro-Partitioning and virtual devices, is supported. AIX 5L V5.3 partitions can use physical and virtual resources at the same time.

1.4.2 Linux for System p5 systems

Linux is an open source operating system that runs on numerous platforms from embedded systems to mainframe computers. It provides a UNIX®-like implementation across many computer architectures.

This section discusses two versions of Linux to be run in partitions; it does not discuss the Linux-based Virtual I/O Server. The supported versions of Linux on System p5 servers are:

- Novell SUSE Linux Enterprise Server V9 and V10
- Red Hat Enterprise Linux Advanced Server V3 and V4

The APV virtualization features, except for virtual SCSI server, are supported in Version 2.6.9 of the Linux kernel. The commercially available latest distributions from Red Hat, Inc. (RHEL AS 4) and Novell SUSE LINUX (SLES 9 and SLES 10) support the IBM POWER4, POWER5, and PowerPC® 970 64-bit architectures and are based on this 2.6 kernel series. Also supported is the modified 2.4 kernel version of Red Hat Enterprise Server AS 3 (RHEL AS 3) with update 3. The Linux kernel first shipped with SLES9 SP1 also supports virtual SCSI server.

Clients wishing to configure Linux partitions in virtualized System p5 systems should consider the following:

- Not all devices and features supported by the AIX 5L operating system are supported in logical partitions running the Linux operating system.
- Linux operating system licenses are ordered separately from the hardware. Clients can acquire Linux operating system licenses from IBM, to be included with their System p5 systems, or from other Linux distributors.
- Clients or authorized business partners are responsible for the installation and support of the Linux operating system on the System p5 systems.

- Regardless of how a Linux distribution is ordered, the distributors offer maintenance and support. IBM also has support offerings from IBM Global Services for these distributions.
- While Linux can be run successfully on partitions with more than eight processors (the Linux kernel 2.6 may scale up to 16 or even 24 processors for certain workloads) typical Linux workloads will only effectively utilize up to 4 or 8 processors.

Supported virtualization features

SLES9 and RHEL AS4 support the following virtualization features:

- Virtual SCSI, including for the boot device
- Shared-processor partitions and virtual processors, capped and uncapped
- Dedicated-processor partitions
- Dynamic reconfiguration of processors
- Virtual Ethernet, including connections through the Shared Ethernet Adapter in the Virtual I/O Server to a physical Ethernet connection
- Simultaneous multithreading (SMT)

Neither SLES9 or RHAS4 support the following:

- Dynamic reconfiguration of memory
- Dynamic reconfiguration of I/O slots
- Partition Load Manager (PLM)

1.4.3 IBM i5/OS for System p5 systems

The IBM i5 operating system (i5/OS) provides flexible workload management options for the rapid deployment of high-performance enterprise applications. With its base set of functions and no-charge options, i5/OS provides ease of implementation, management, and operation by integrating the base software that most businesses need.

Operating system positioning for System p5 systems

i5/OS running on a System p5 system is intended for clients who have a limited amount of i5/OS workload, limited growth anticipated for this workload, and wish to consolidate onto a single server where the majority of the workload will be either AIX 5L or Linux. Clients with new medium or small i5/OS workloads or with older models of AS/400® or iSeries™ can upgrade existing System p5 systems to include i5/OS partitions with dedicated processors or shared-processor partitions.

Note: i5/OS partitions are only supported on p5-570, p5-590, and p5-595 servers and require the I/O sub-system for i5/OS.

1.4.4 Summary

Table 1-2 shows a summary of the relationship between APV functions and operating systems supported by the systems.

Table 1-2 Supported operating systems for the APV features

System / APV feature	VIOS V1.1, V1.2, V1.3	VIOS Linux	i5/OS	AIX 5L V5.2 ML2	AIX 5L V5.3	RHEL AS V3	RHEL AS V4	SLES V9
Partitions with dedicated processors	X	X	X	X	X	X	X	X
Micro-partitions	X	X	X		X	X	X	X
Virtual TTY	X	X		X	X	X	X	X
Virtual console client/server		X	X			X	X	X
Virtual Ethernet	X	X	X		X	X	X	X
Boot from virtual Ethernet					X	X	X	X
Shared Ethernet Adapter	X							
Ethernet bridge with STP support		X				X	X	X
Virtual SCSI server	X	X						
Virtual SCSI client					X	X	X	X
Boot from Virtual SCSI client disk					X	X	X	X
Virtual Tape			X					
Virtual CD	X	X	X		X	X	X	X
Boot from Virtual CD					X	X	X	X
Partition Load Manager				X	X			
Integrated Virtualization Manager	X				X	X	X	X
Dynamic LPAR CPU	X	X	X	X	X		X	X
Dynamic LPAR RAM	X		X	X	X			
Dynamic LPAR physical I/O	X		X	X	X			
Dynamic LPAR virtual adapters	X		X		X			
On-line scan of Virtual SCSI devices	X		X		X		X	X

1.5 Comparison of two IBM virtualization technologies

The IBM System z9™ has a long heritage in partitioning and virtualization. Some of the new virtualization features introduced in the IBM System p5 systems have been adopted from IBM System z9 and its predecessors. If you are familiar with the System z9 system's partitioning and virtualization concepts, you should be aware that some of them are similar but not identical on the IBM System p5.

There are two virtualization options on z9: PR/SM™ and z/VM®. PR/SM provides logical partitioning and basic virtualization, while z/VM offers advanced virtualization technology. z/VM can be deployed in a System z9 LPAR and provides Virtual Machines (VM) with virtual resources. The capabilities of System p5 virtualization lie somewhere between those of the two System z9 virtualization options.

Table 1-3 provides a brief overview of the virtualization capabilities available on IBM System z9 and System p5. Some differences are summarized in the following discussion.

Table 1-3 Virtualization capabilities of IBM System z9 and System p5 compared

Function / capability	**IBM System z9 virtualization technology**		**IBM System p5 virtualization technology**
Enabling software	Processor Resource / Systems Manager (PR/SM).	z/VM.	POWER Hypervisor.
Maximum number of virtualized servers	Maximum of 60 Logical Partitions (LPARs), depending on model.	Arbitrary number of Virtual Machines (VMs), also called Guests, limited only by resources.	Maximum of 254 Logical Partitions (LPARs), depending on model, maximum of 10 per processor.
Sharing processor resources	LPARs are assigned logical processors and weighted shares of central processors or dedicated processors. LPARs with shared processors can be uncapped or capped.	VMs are assigned shared or dedicated virtual processors and absolute or relative weighted shares of virtual processors. VMs with shared processors can be uncapped, soft-, or hard-capped.	LPARs are assigned either dedicated physical CPUs, entitlements of physical CPUs, and a number of virtual processors. LPARs with shared CPUs can be either capped or uncapped.
Interpartition Load Management	Intelligent Resource Director (IRD) with z/OS® partitions.	Virtual Machine Resource Manager (VMRM).	Partition Load Manager (PLM) with AIX 5L partitions.

Function / capability	IBM System z9 virtualization technology		IBM System p5 virtualization technology
Sharing memory resources	LPAR memory is fixed and private; for LPARs running z/OS, the memory size can be dynamically changed with some considerations.	Portions of VM memory can be shared read-only or read-write with other VMs; changing the memory size of a VM requires IPLing this VM.	LPAR memory is fixed and private; memory size can be dynamically changed for LPARs running AIX 5L.
Virtual interpartition communication	TCP/IP with HiperSockets™.	TCP/IP with virtual HiperSockets, TCP/IP, and other protocols through virtual Ethernet with IEEE 802.1Q VLAN support.	TCP/IP and other protocols through virtual Ethernet with IEEE 802.1Q VLAN support.
Sharing connections to external networks	Enhanced Multiple Image Facility (EMIF) multiplexes Open Systems Adapter (OSA) to multiple LPARs.	Virtual Ethernet switch bridges virtual Ethernet to an external Ethernet though an OSA, z/VM also takes advantage of multiplexed access to OSA through EMIF.	Shared Ethernet Adapter (SEA), hosted by the Virtual I/O Server (VIOS), acts as layer-2-bridge between physical and virtual Ethernet adapters.
Sharing I/O resources.	EMIF multiplexes channels and devices to multiple LPARs.	z/VM provides virtual devices, such as minidisks, which are partitions of physical disks, and it provides shared access to physical devices.	VIOS provides virtualized disks, which can be partitions of physical disks and are accessed through virtual SCSI adapters.
Supported OS.	z/OS, Linux, z/VM, and others zSeries® operating systems.		AIX 5L, Linux, and i5/OS on some models.

The mechanisms for sharing of processor resources on IBM System z9 and System p5 are similar. The number of virtual processors in a System p5 LPAR or in a z/VM virtual machine can exceed the number of installed physical processors. There are some differences with capping that are beyond the scope of this discussion.

On a System z9 server, integrated inter- and intra-partition workload management can be performed with the Intelligent Resource Director (IRD) for z/OS LPARs, while on a System p5, inter-partition management is done with the Partition Load Manager (PLM), and intra-partition management with the AIX 5L Workload Manager (WLM). Since PLM and WLM are not integrated, PLM on System p5, like VMRM on System z9, monitors WLM's application priorities and

goals, while IRD can adjust resource allocation if some application's performance goals are missed.

Using LPARs on System z9 and System p5 memory is partitioned. Thus, the sum of memory assignments to LPARs cannot exceed the physically installed memory. z/VM can share memory by implementing paging of VM. Thus, the sum of assigned memory can exceed physically installed memory and generally does. Paging automatically and dynamically adjusts physical memory assignments for VMs.

System p5 and z/VM provide virtual Ethernet and implement virtual Ethernet switches and bridges, which operate on layer-2 and could be used with any layer-3 protocol. PR/SM's HiperSockets operate on layer-3 and provide IP-based communication only. Shared OSA supports layer-2 and layer-3 access to external networks.

A substantial difference between System z9 PR/SM and System p5 exists with respect to sharing of I/O resources and access to external networks:

- On System z9 with PR/SM, access to disks, tapes, network adapters, and other I/O resources is multiplexed by EMIF. Thus, shared physical resources can be owned by multiple LPARs on a System z9.
- z/VM, in addition to the features of PR/SM, allows virtual devices to be shared between VMs. Bridged access from virtual Ethernet networks to external networks is available (similar to System p5).
- On System p5, virtual disks are created that are backed by physical disks, and virtual Ethernet networks can be bridged to physical Ethernet adapters. These physical resources are owned by the Virtual I/O Server on a System p5.

Virtual I/O and virtual Ethernet are based on system software and Hypervisor firmware on a System p5 and with z/VM, while most of EMIF is implemented in the System z9 hardware and firmware. Thus, the processor load of I/O processing with EMIF on System z9 is lower than that of virtual I/O on System p5 or z/VM. It comes close to dedicated physical I/O adapters on System p5.

z/VM runs in a LPAR of System z™; thus, PR/SM and z/VM are actually nested. System z9 implements two levels of interpretive execution to allow hardware execution of virtual machines on z/VM in a logical partition. z/VM can be self-hosting, which means that you can run z/VM in a z/VM guest, thus running z/VM on z/VM in an LPAR of a System z9. z/VM can host z/VM to any practical level. You cannot nest LPARs of System z9 or System p5.

1.6 The value of the Advanced POWER Virtualization

With POWER5 processor-based systems, introduced in 2004, customers have realized the immense value of the Advanced POWER Virtualization. There are quite a few fields where this value is recognized as a huge advantage in planning an architecture of POWER5 systems as well as in day to day operations on those systems. This technology now is widely perceived by customers as extremely stable, mature, and production ready.

There have been a number of case studies on how to set up and use those systems, as well as literature (APV Deployment Examples) showing how to implement POWER5 systems in smaller and larger environments.

Those techniques are now well known and implementations of easier and more complex installations either with the Integrated Virtualization Manager or under control of the Hardware Management Console are now common.

Possibly the more challenging tasks with POWER5 systems are moving from planning and implementation to the careful design of systems management. The background for this is that the more these systems and their virtualization features are used and the higher the average number of logical partitions per system grows, the more the challenge lies in coordinating overall tasks that impact the parts of the system, such as system code upgrades. Coordination of systems now requires a more holistic datacenter view, rather than a single server view.

To say that the most challenging tasks are no longer associated with the actual management and operation of machines rather than with organizational planning and coordination means that Advanced Power Virtualization fulfills its task well. It enables administrators and companies to build their own flexible and reliable infrastructure. It allows for the rapid deployment of new systems without the normally lengthy and time-consuming process of validation, acquisition, and installation of new physical systems. It is designed to simplify the movement of operating system instances from one physical system to another given that the infrastructure is carefully planned and certain requirements are met. It allows for fast setup of test and development machines in the same physical environment and even on the same machine as production machines, taking away the bias if different types of physical machines were used. And it adds trust and confidence in systems operations since the sheer computing power and the stability of POWER5 processor based systems are proven and recorded by many installations. Overall, this leads to a substantial improvement in TCO and simplified IT operations.

The real value of the Advanced Power Virtualization as perceived by many customers is smooth systems operations and fast responses to client demands as a result of all the factors mentioned above.

2

Virtualization technologies on System p servers

In this chapter, the various technologies that are part of IBM System p™ systems are discussed. Specifically, the following topics are covered:

- New features in Version 1.3 of the Virtual I/O Server
- Features in Version 1.2 of the Virtual I/O Server
- The Advanced POWER Virtualization feature
- Micro-Partitioning introduction
- Introduction to simultaneous multithreading
- Introduction to the POWER Hypervisor
- Software licensing in a virtualized environment
- Virtual and Shared Ethernet introduction
- Virtual SCSI introduction
- Partition Load Manager introduction
- Integrated Virtualization Manager
- Dynamic LPAR operations
- Linux virtual I/O concepts

2.1 New features in Version 1.3 of the Virtual I/O Server

Version 1.3 of the Virtual I/O Server, available since 08/2006, includes various new functions and enhancements. All of these are also contained in Fix Pack 8.0 whose installation effectively updates an installed Virtual I/O Server to Version 1.3 (ioslevel 1.3.0.0).

2.1.1 New and enhanced features in Virtual I/O Server Version 1.3

The following provides a list of the new and enhanced features in Version 1.3:

- Virtual I/O Server

  The Virtual I/O Server received the following major enhancements:

  - Newly added software: SSH now comes preinstalled with the Virtual I/O Server

    For instructions on how to configure automated non-prompted logins through SSH, refer to 6.2.2, “Install and configure SSL and SSH” on page 382.

    In the default installation, SSH is meant for remote login to the Virtual I/O Server. Therefore, the padmin user cannot use the **ssh** command itself. Neither is it possible to use the **scp** command from a remote server to copy files from or to the Virtual I/O Server.

  - Improved monitoring with additional **topas** and **viostat** command performance metrics

    The **topas** command received a new -cecdisp flag that shows the overall performance of the physical machine as well as the load on active LPARs.

    The **viostat** command received a new -extdisk flag that shows detailed disk statistics. For the **viostat** command to work with disk history, the system has to be changed to keep the measurements using the following command:

    ```
    $ chdev -dev sys0 -attr iostat=true
    ```

  - Enabled Performance (PTX®) agent (PTX is a separately purchased LPP.)

  - Increased performance for virtual SCSI and virtual Ethernet

    Due to improvements in the code, the performance of both is increased.

  - Enhanced command-line interface

    Several new commands have been added:

    - The **wklmgr** and **wkldagent** commands for handling Workload Manager. If enabled, they can be used to record performance data that then can be analyzed by **wkldout** and viewed.

- The **chtcpip** command for online management of TCP/IP parameters.
- The **crontab** command for handling cron jobs.
- The **viosecure** command for handling the security settings. This is in fact a central point to control security settings of the Virtual I/O Server, also handling the local firewall settings to secure the Virtual I/O Server based either on standard values or customized on a per-port basis.

Other commands that received enhancements are **startnetsvc**, **stopnetsvc**, **lstcpip**, **lsdev**, and **mkbdsp**.

A complete list of commands and their parameters can be found in the Virtual I/O Server documentation at:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/home.html

Look in the Virtual I/O Server Commands Reference, located under “Printable PDFs”.

– Enabled additional storage solutions

This refers to the enablement of N series Storage Subsystems. The complete and always updated list of supported storage subsystems, operating systems, and adapters can be found at:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html

► Virtual SCSI and shared Fibre Channel adapters

Disk-based I/O was enhanced in the following ways:

– Support for iSCSI TOE adapter and virtual devices backed by System Storage™ N series iSCSI disks

– Virtual SCSI (VSCSI) functional enhancements

- Support for SCSI reserve/release for limited configurations. Enables virtual SCSI for certain clustered/distributed configurations.
- Support on the AIX 5L client for a changeable queue depth attribute for virtual disks. Allows the system administrator to tune the size of the I/O queue for each virtual disk.

  With this enhancement, it is now possible to tune virtual disks using the command:

  ```
  $ chdev -dev hdiskN -attr queue_depth=X
  ```

  where N is the number of the hdisk and X denotes the desired depth of the queue. Therefore it is now possible to have values larger than three, which was the previous non-changeable default. The new value should be chosen carefully as this can have impacts on system

throughput and performance. For additional information, see *IBM System p Advanced POWER Virtualization Best Practices*, REDP-4194.

- Improvements for altering virtual device capacity without disruption.

  When expanding the LUN size on an FC attached storage subsystem as well as extending a Logical Volume within a volume group, the Virtual I/O Server will detect this and the virtual SCSI server adapter will present this change to the virtual SCSI client adapter. Upon recognition of the change, the volume group in the LPAR containing the hdisk represented by the LUN can take adavantage of the bigger disk with the command:

  ```
  # chvg -g volume_group
  ```

  and will immediately be able to use the additional storage capacity. This is valid for non-rootvg disks only. The **chvg** command is not supported with rootvg, as stated in the manual page.

- A configurable attribute that, when enabled, allows the virtual client adapter driver to determine the health of the VIOS to improve and expedite path failover processing.

  The vscsi_path_to attribute of the virtual SCSI client adapter was added. A value of 0 (default) disables it, while any other value defines the number of seconds the VSCSI client adapter will wait for commands issued to the VSCSI server adapter and not serviced. If that time is exceeded, the VSCSI client adapter attempts the commands again and waits up to 60 seconds until it fails the outstanding commands, writes an error to the error log and, if MPIO is used, tries another path to the disk. Therefore, this parameter should only be set for MPIO installations with dual Virtual I/O Servers.

- VSCSI device error log enhancements.

► Virtual Ethernet and shared Ethernet adapter

Ethernet support was enhanced in the following ways:

– TCP Segmentation Offload

TCP segmentation offload (large send) enables TCP large send capability (also known as segmentation offload) from logical partitions to the physical adapter. The physical adapter in the SEA must be enabled for TCP largesend for the segmentation offload from the LPAR to the SEA to work, and the SEA itself must be enabled for large send. Also, the operating system must be capable of performing a large send operation. AIX 5L virtual Ethernet adapters are, by default, capable of large sends.

You can enable or disable TCP large send on the SEA using the CLI **chdev** command. To enable it, use the -attr largesend=1 option. To disable it, use

the -attr largesend=0 option. For example, the following command enables large send for Shared Ethernet Adapter ent1:

```
# chdev -dev ent1 -attr largesend=1.
```

By default, the setting is disabled (largesend=0).

LPARs can now offload TCP segmentation to VIOS Ethernet adapter, benefitting CPU intensive applications running on the LPAR.

If a physical Ethernet adapter is used that supports large_send (packets of 64 KB are directly handed to the physical adapter layer where they are processed and split up in the packet size the physical network supports), this parameter can be enabled to boost performance. All writes to the physical network will benefit from this; the internal virtual network will use performance optimized network settings.

To enable large send on a physical adapter on AIX 5L, use the following command:

```
# chdev -l entX -a large_send=yes
```

To enable large send on a physical adapter on the on the Virtual I/O Server, use the following command:

```
# chdev -dev entX -attr large_send=yes
```

As of Virtual I/O Server Version 1.3, the default value for large send on physical Ethernet adapters is yes. The default for the Shared Ethernet Adapter is 0 (meaning no).

This will benefit only outbound traffic that is data to be send. Inbound traffic will be received in packets as they arrive from the physical network.

Note: The following is a common source of errors:

To modify large send on a physical Ethernet adapter, use the command option:

large_send=yes or large_send=no

To modify large send on a Shared Ethernet Adapter (SEA), use the command option:

largesend=1 or largesend=0

- Monitoring

  Agents for monitoring using Topas (included with AIX 5L V5.3) and PTX. PTX is a separately purchased LPP.

- IVM enhancements

  Industry leading functions in this release include:

  - Support for dynamic logical partitioning (dynamic LPAR) for memory and processors in managed partitions. This can be done on the command-line as well as in the IVM Web interface.
  - IP configuration support for IVM

    After initial configuration of TCP/IP, the Web interface can be contacted; from that time on, new interfaces can be created and configured via the Web interface.
  - Task Manager for long-running tasks

    The last 40 tasks can graphically be monitored and shown with the button “Monitor Tasks” in the Web interface. Every task can be examined to see the properties of that task.

2.1.2 Additional information

There is a number of resources listed in “Related publications” on page 443 that should be considered for further reading and information.

There exists an encouraging collaboration as well as exchange of knowledge that will help with many questions ranging from performance monitoring and tools, virtualization, and AIX 5L to Linux on POWER.

Follow this link to start an overview of the forums available in the AIX 5L and Linux on POWER community:

http://www-03.ibm.com/systems/p/community/

We want to encourage you to participate in the community. It already contains much information and continues to grow. AIX developers are also contributing their knowledge and help as well as many users of the technology. The more participants work interactively in the forums and wikis, the more useful it will be for everyone concerned with POWER processor based machines.

2.2 Features in Version 1.2 of the Virtual I/O Server

This section contains a short overview of the features that are included in Version 1.2 of the Virtual I/O Server, such as:

- Virtual optical device support
- Shared Ethernet Adapter failover

- Integrated Virtualization Manager
- Storage pool commands
- HMC enhancements for easier configuration and maintenance

Many of the improvements in the Virtual I/O Server Version 1.2 are intended to simplify the configuration and management of the virtualized I/O environment.

Note: To use all the new features listed in this section, an update of the system's microcode, HMC code, and the Virtual I/O Server may be needed.

2.2.1 Virtual DVD-RAM, DVD-ROM, and CD-ROM

Virtual SCSI (VSCSI) enables the sharing of physical storage devices (SCSI and Fibre Channel) between client partitions. Version 1.2 of the Virtual I/O Server adds support for optical devices, such as DVD-RAM and DVD-ROM. CD-ROM was supported in previous versions.

Writing to a shared optical device is currently limited to DVD-RAM. DVD+RW and DVD-RW devices are not supported.

The physical storage device must be owned by the Virtual I/O Server and it is mapped in a similar way as a virtual disk to a virtual SCSI server adapter using the **mkvdev** command.

The virtual optical device can be assigned to only one client partition at a time. In order to use the device on a different client partition, it must first be removed from the partition currently owning the shared device and reassigned to the partition that will use the device. This is an advantage over dynamic LPAR because you do not have to manually move the device's adapter.

For more information about sharing optical devices, refer to 2.9, "Virtual SCSI introduction" on page 89.

2.2.2 Shared Ethernet Adapter failover

Version 1.2 of the Virtual I/O Server introduces a new way to configure backup Virtual I/O Servers to provide higher availability, for external network access, over Shared Ethernet Adapters (SEA).

Shared Ethernet Adapter failover provides redundancy by configuring a backup Shared Ethernet Adapter on a different Virtual I/O Server partition that can be used if the primary Shared Ethernet Adapter fails. The network connectivity of the client logical partitions to the external network continues without disruption.

Shared Ethernet Adapter failover can be configured by first creating a virtual adapter with the access to external network flag (trunk flag) set. This virtual adapter must have the same PVID or VLAN ID as the corresponding virtual adapter on a backup Virtual I/O Server. It uses a priority value given to the virtual Ethernet adapters during their creation to determine which Shared Ethernet Adapter will serve as the primary and which will serve as the backup. The Shared Ethernet Adapter that has the virtual Ethernet configured with the numerically lower priority value will be used preferentially as the primary adapter.

For the purpose of communicating between themselves to determine when a failover should take place, Shared Ethernet Adapters in failover mode use a VLAN dedicated for such traffic, named the control channel. A virtual Ethernet (created with a PVID that is unique on the system) must be created on each Virtual I/O Server that provides a SEA for the failover purpose. This virtual Ethernet is then specified as the control channel virtual Ethernet when each Shared Ethernet Adapter is created in failover mode. Using the control channel, the backup Shared Ethernet Adapter discovers when the primary adapter fails, and network traffic from the client logical partitions is sent over the backup adapter. When the primary Shared Ethernet Adapter recovers from its failure, it again begins actively bridging all network traffic.

For more information about the SEA Failover function, refer to 4.1.3, “High availability for communication with external networks” on page 189.

2.2.3 Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) is a basic hardware management solution, included in the VIO software Version 1.2, that inherits key Hardware Management Console (HMC) features.

The IVM is used to manage partitioned System p5 systems with a Web-based graphical interface without requiring an HMC. This reduces the hardware needed for adoption of virtualization technology, particularly for the low-end systems. This solution fits in small and functionally simple environments where only few servers are deployed or not all HMC functions are required.

Note: The IVM feature is not available, at the time of writing, for Virtual I/O Server partitions on the IBM System p5 Models 570, 575, 590, and 595.

For more information about the concepts, installation, and configuration of IVM, refer to 2.11, “Integrated Virtualization Manager” on page 103. For further information, refer to *Integrated Virtualization Manager on IBM System p5*, REDP-4061.

2.2.4 New storage pool commands

Similar to volume groups, storage pools are collections of one or more physical volumes that abstract the organization of the underlying disks. The physical volumes that comprise a storage pool can be of varying sizes and types.

Using storage pools, you are no longer required to have extensive knowledge on how to manage volume groups and logical volumes to create and assign logical storage to a client partition. Devices created using the storage pool are not limited to the size of the individual physical volumes. Storage pools are created and managed using the following commands:

mksp	Creates a storage pool, using the physical volumes represented by the physical volume parameter.
chsp	Adds or removes physical volumes from a storage pool, or sets the default storage pool.
lssp	Lists information about storage pools.
mkbdsp	Attaches storage from a storage pool to a virtual SCSI adapter.
rmbdsp	Removes storage from a virtual SCSI adapter and returns it to the storage pool.

The default storage pool is rootvg. We recommend creating another storage pool where you define your backing devices that are used as virtual disks on your client partition.

The backing devices are created using the **mkbdsp** command. In only one step, you can create a backing device with a specified size and map it to the virtual SCSI server adapter that is assigned to the appropriate client partition.

Note: When assigning whole physical volumes as backing devices, they cannot be part of a storage pool. In this case, you have to directly map the physical disk.

2.2.5 HMC enhancements

The HMC software provides an enhanced graphical interface to ease the configuration and maintenance of the virtual I/O adapters for managed servers. This new feature provides enhancements to simplify the Virtual I/O environment starting with HMC Version 5.1.

To ease the configuration of virtual SCSI adapters, an HMC offers you the option to add a virtual SCSI server adapter dynamically to the Virtual I/O Server when creating the client partition or when adding another virtual SCSI client adapter to

the partition. This allows you to create client and server SCSI adapters in one step. Then you have to add the virtual SCSI server adapter to the appropriate Virtual I/O Server partition profile to make the change permanent. Another way of doing that would be to save a new partition profile with a different name. That profile will automatically contain any dynamically added features, thereby simplifying the task of adding the adapter configuration to a profile and excluding errors that may arise through manual addition of dynamically defined adapters to the static profile. To use the new profile as the default profile it has to be assigned by right-clicking the LPAR and setting this to be the default

To ease maintenance and configuration changes, the HMC provides an overview of the virtual Ethernet and virtual SCSI topologies configured on the Virtual I/O Server.

IBM supports up to ten Virtual I/O Servers within a single CEC managed by an HMC. Though architecturally up to 254 LPARS are supported, more than ten Virtual I/O Server LPARs within a single CEC has not been tested and therefore is not recommended.

For more information and configuration details, refer to 2.12, “Dynamic LPAR operations” on page 108.

2.3 The Advanced POWER Virtualization feature

This section provides information about the packaging and ordering information for the Advanced POWER Virtualization feature available on the IBM System p platform.

The Advanced POWER Virtualization feature is a combination of hardware enablement and software that includes the following components that are available together as a single priced feature:

- Firmware enablement for Micro-Partitioning
- Installation image for the Virtual I/O Server software, which supports:
  - Shared Ethernet Adapter
  - Virtual SCSI server
  - Integrated Virtualization Manager (IVM) for supported systems
- Partition Load Manager (only supported for HMC managed systems and not part of POWER Hypervisor and Virtual I/O Server FC 1965 on IBM OpenPower systems).

Virtual Ethernet is available without this feature for servers attached to an HMC or managed using the IVM.

When the hardware feature is specified with the initial system order, the firmware is shipped activated to support Micro-Partitioning and the Virtual I/O Server. For upgrade orders, IBM will ship a key to enable the firmware (similar to the CUoD key).

Clients can visit the following Web site:

http://www-912.ibm.com/pod/pod

to look at the current activation codes for a specific server by entering the machine type and serial number. The activation code for Advanced POWER Virtualization feature has a type definition of *VET* in the window results.

For systems attached to an HMC, Figure 2-1 shows the HMC window where you enable the Virtualization Engine™ Technologies.

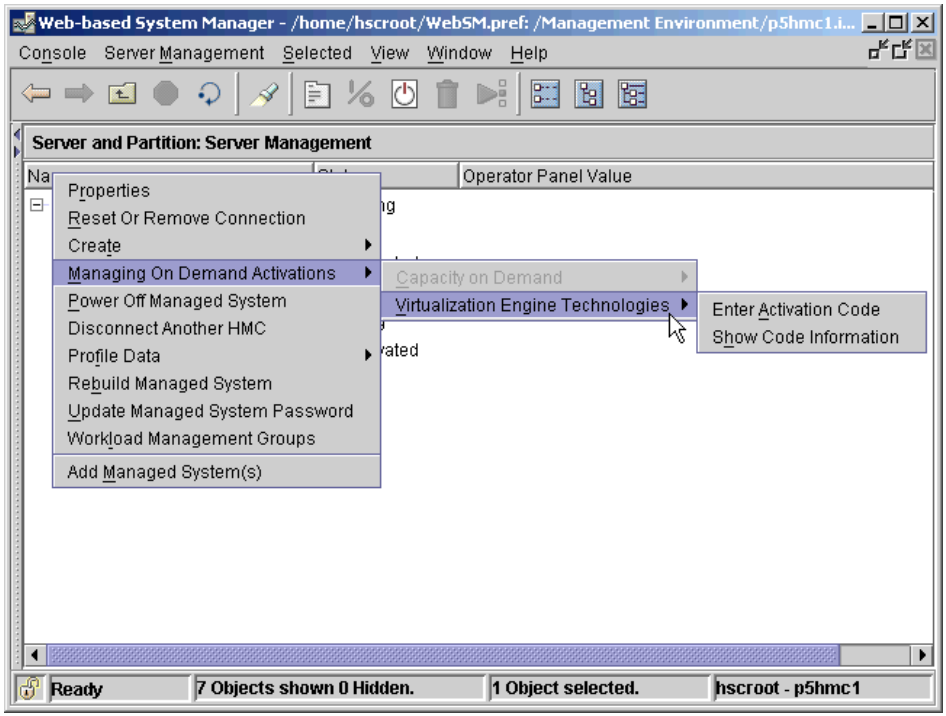


Figure 2-1 HMC window to enable the Virtualization Engine Technologies

When using the IVM within the Virtual I/O Server to manage a single system, Figure 2-2 on page 30 shows the Advanced System Management Interface (ASMI) menu to enable the Virtualization Engine Technologies. For more information about this procedure, refer to *Integrated Virtualization Manager on IBM System p5*, REDP-4061.

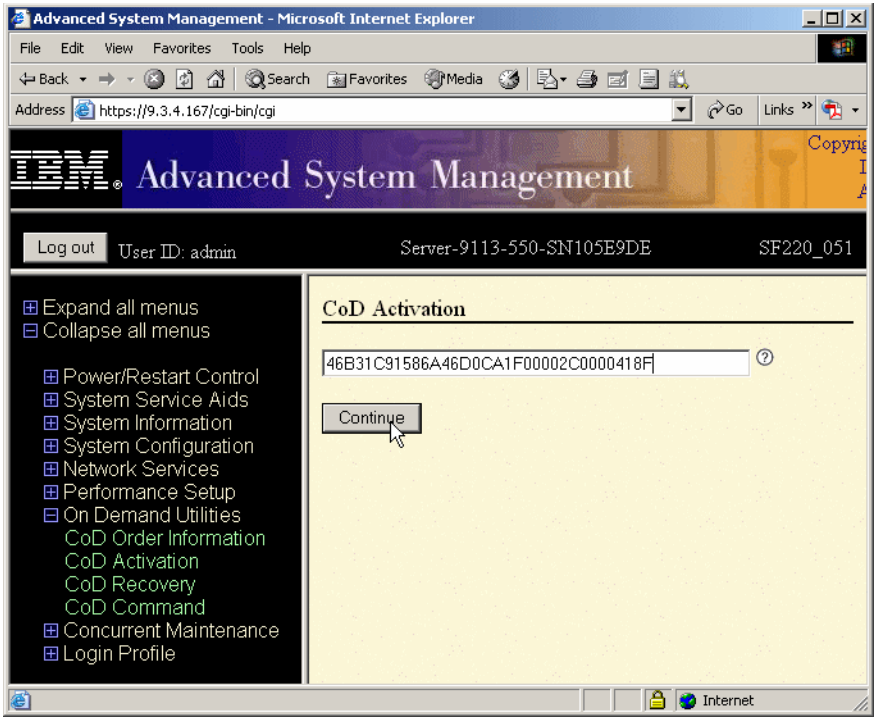


Figure 2-2 ASMI menu to enable the Virtualization Engine Technologies

The Virtual I/O Server and Partition Load Manager (PLM) are licensed software components of the Advanced POWER Virtualization feature. They contain one charge unit for each installed processor, including software maintenance. The initial software license charge for the Virtual I/O Server and PLM is included in the price of the Advanced POWER Virtualization feature.

Table 2-1 provides an overview of the Advanced POWER Virtualization features on the IBM System p systems.

Table 2-1 APV feature code overview

Servers	Feature code	Included in base configuration?	IVM supported?
9115-505	7432	No	Yes
9110-510	7432	No	Yes
9123-710[a]	1965	No	Yes
9111-520	7940	No	Yes
9131-52A	7940	No	Yes
9124-720[a]	1965	No	Yes
9113-550	7941	No	Yes
9133-55A	7941	No	Yes
9117-570	7942	No	No
9118-575	7944	No	No
9119-590	7992	Yes	No
9119-595	7992	Yes	No

[a] PLM is not shipped with FC 1965

The Advanced POWER Virtualization feature is configured optionally and charged for all mentioned systems above except for the 9119-590 and 595 systems, which include the Advanced POWER Virtualization feature as a part of the base system configuration. The software maintenance is charged additionally for all mentioned systems.

For each Virtual I/O Server license ordered, an order for either the one-year (5771-VIO) or three-year (5773-VIO) Software Maintenance (SWMA) is also submitted. You must purchase a license for each active processor on the server. For an HMC attached system, the processor-based license enables you to install multiple Virtual I/O Server partitions on a single server to provide redundancy and to spread the I/O workload across multiple Virtual I/O Server partitions.

The supported Virtual I/O clients are:

- AIX 5L Version 5.3
- SUSE LINUX Enterprise Server 9 for POWER
- SUSE LINUX Enterprise Server 10 for POWER
- Red Hat Enterprise Linux AS 3 for POWER (update 2 or later)
- Red Hat Enterprise Linux AS 4 or POWER or later

The Virtual I/O Server provides the virtual SCSI server and Shared Ethernet Adapter virtual I/O function to client partitions (Linux or AIX 5L), and the IVM

management interface for systems without an HMC. This POWER5 partition is not intended to run end-user applications or for user login.

For each Partition Load Manager V1.1 (5765-G31) license ordered, an order for either the one-year (5771-PLM) or three-year (5773-PLM) Software Maintenance (SWMA) must also be submitted. The software maintenance for the Partition Load Manager is priced on a per processor basis, by processor group.

Partition Load Manager for AIX 5L helps clients to maximize the utilization of processor and memory resources on the IBM System p platform that support dynamic logical partitioning. Within the constraints of a user-defined policy, resources are automatically moved to partitions with a high demand, from partitions with a lower demand. Resources that would otherwise go unused can now be more fully utilized.

2.4 Micro-Partitioning introduction

Micro-Partitioning is the ability to divide a physical processor's computing power into fractions of a processing unit and share them among multiple logical partitions. Obtaining and entering an activation code for most IBM System p models is optional, except for the p5-590 and p5-595 models, where it is included automatically in the configuration.

The benefit of Micro-Partitioning is that it allows increased overall utilization of CPU resources within the managed system. Better granularity of CPU allocation in a logical partition means efficient use of processing power.

This section discusses the following topics about Micro-Partitioning:

- Shared processor partitions
- Shared processor pool overview
- Capacity Upgrade on Demand
- Dynamic processor de-allocation and processor sparing
- Dynamic partitioning
- Shared processor considerations

2.4.1 Shared processor partitions

The virtualization of physical processors in POWER5 systems introduces an abstraction layer that is implemented within the hardware microcode. From an operating system perspective, a virtual processor is the same as a physical processor.

The key benefit of implementing partitioning in the hardware allows any operating system to run on POWER5 technology with little or no changes. Optionally, for optimal performance, the operating system can be enhanced to exploit shared processor pools more in-depth, for example, by voluntarily relinquishing CPU cycles to the hardware when they are not needed. AIX 5L Version 5.3 is the first version of AIX 5L that includes such enhancements.

Micro-Partitioning allows multiple partitions to share one physical processor. Logical partitions using Micro-Partitioning technology are referred to as shared processor partitions.

A partition may be defined with a processor capacity as small as .10 processor units. This represents 1/10 of a physical processor. Each processor can be shared by up to 10 shared processor partitions. The shared processor partitions are dispatched and time-sliced on the physical processors under control of the POWER Hypervisor.

Micro-Partitioning is supported across the entire POWER5 product line from entry level to the high-end systems. Table 2-2 shows the maximum number of logical partitions and shared processor partitions supported on the different models.

Table 2-2 Micro-Partitioning overview

Server/Model	505/510/ 520/52A	550	55A	570	575	590	595
Processors	2	4	8	16	16	32	64
Dedicated processor partitions	2	4	8	16	16	32	64
Shared processor partitions	20	40	80	160	160	254	254

It is important to point out that the maximums stated are supported by the hardware, but the practical limits based on production workload demands may be significantly lower.

Shared processor partitions still need dedicated memory, but the partitions I/O requirements can be supported through virtual Ethernet and virtual SCSI. Utilizing all virtualization features, up to 254 shared processor partitions is currently supported.

The shared processor partitions are created and managed by the HMC. When you start creating a partition, you have to choose between a shared processor partition and a dedicated processor partition.

When setting up a partition, you have to define the resources that belong to the partition, such as memory and I/O resources. For processor shared partitions, you have to configure these additional options:

- Minimum, desired, and maximum processing units of capacity
- The processing sharing mode, either capped or uncapped
- Minimum, desired, and maximum virtual processors

These settings are the topic of the following sections.

Processing units of capacity

Processing capacity can be configured in fractions of 1/100 of a processor. The minimum amount of processing capacity that has to be assigned to a partition is 1/10 of a processor.

On the HMC, processing capacity is specified in terms of *processing units*. The minimum capacity of 1/10 of a processor is specified as 0.1 processing units. To assign a processing capacity representing 75% of a processor, 0.75 processing units are specified on the HMC.

On a system with two processors, a maximum of 2.0 processing units can be assigned to a partition. Processing units specified on the HMC are used to quantify the minimum, desired, and maximum amount of processing capacity for a partition.

Once a partition is activated, processing capacity is usually referred to as capacity entitlement or entitled capacity.

Capped and uncapped mode

Micro-partitions have a specific processing mode that determines the maximum processing capacity given to them from the shared processor pool. The processing modes are:

Capped mode The processing units given to the partition at a time never exceed the guaranteed processing capacity (the entitlement capacity is guaranteed by the system and it is not exceeded when resources are available in the shared processing pool).

Uncapped mode The processing capacity given to the partition at a time may exceed the guaranteed processing capacity when resources are available in the shared processing pool. You must specify the uncapped weight of that partition.

If multiple uncapped logical partitions require idle processing units, the managed system distributes idle processing units to the logical partitions in proportion to each logical partition's uncapped weight. The higher the uncapped weight of a logical partition, the more processing units the logical partition gets.

The uncapped weight must be a whole number from 0 to 255. The default uncapped weight for uncapped logical partitions is 128. A partition's share is computed by dividing its variable capacity weight by the sum of the variable capacity weights for all uncapped partitions. If you set the uncapped weight at 0, the managed system treats the logical partition as a capped logical partition. A logical partition with an uncapped weight of 0 cannot use more processing units than those that are committed to the logical partition.

A weight of 0 allows automated software to provide the equivalent function as a dynamic LPAR operation to change uncapped to capped.

Virtual processors

A virtual processor is a depiction or a representation of a physical processor to the operating system of a partition that makes use of a shared processor pool. The processing power allocated to a partition, be it a whole or a fraction of a processing unit, will be distributed by the server firmware evenly across virtual processors to support the workload. For example, if a logical partition has 1.60 processing units and two virtual processors, each virtual processor will have 0.80 processing units.

Selecting the optimal number of virtual processors depends on the workload in the partition. Some partitions benefit from greater concurrence, while other partitions require greater power.

By default, the number of processing units that you specify is rounded up to the minimum number of virtual processors needed to satisfy the assigned number of processing units. The default settings maintain a balance of virtual processors to processor units. For example:

- If you specify 0.50 processing units, one virtual processor will be assigned.
- If you specify 2.25 processing units, three virtual processors will be assigned.

You also can use the Advanced tab in your partition profile to change the default configuration and to assign more virtual processors.

A partition in the shared processing pool will have at least as many virtual processors as its assigned processing capacity. By making the number of virtual processors too small, you limit the processing capacity of an uncapped partition. If you have a partition with 0.50 processing units and one virtual processor, the partition cannot exceed 1.00 processing units, because it can only run one job at a time, which cannot exceed 1.00 processing units. However, if the same

partition with 0.50 processing units was assigned two virtual processors and processing resources were available, the partition could use an additional 1.50 processing units.

The minimum number of processing units you can have for each virtual processor depends on the server model. The maximum number of processing units that you can have for each virtual processor is always 1.00. This means that a logical partition cannot use more processing units than the number of virtual processors that it is assigned, even if the logical partition is uncapped. Additionally, the number of processing units cannot exceed the Total Managed system processing units.

Virtual processor folding

Starting with maintenance level 3, AIX 5L V5.3 provides an improved management of virtual processors. This feature enhances the utilization of a shared processor pool by minimizing the use of virtual processors that are idle most of the time. The important benefit of this feature is improved processor affinity, when there is a large number of largely idle shared processor partitions, resulting in effective use of processor cycles. It increases the average virtual processor dispatch cycle, resulting in better cache utilization and reduced Hypervisor workload.

The following are the functions of the virtual processor folding feature:

- Idle virtual processors are not dynamically removed from the partition. They are put to sleep or disabled, and only awoken when more work arrives.
- There is no benefit from this feature when partitions are busy.
- If the feature is turned off, all virtual processors defined for the partition are dispatched to physical processors.
- Virtual processors having attachments, such as **`bindprocessor`** or **`rset`** command attachments, are not excluded from being disabled.
- The feature can be turned off or on. The default is on.

When a virtual processor is disabled, threads are not scheduled to run on it unless a thread is bound to that CPU.

Note: In a shared partition, there is only one affinity node, hence only one node global run queue.

The tunable parameter for this feature is vpm_xvcpus and the default value is 0, which signifies the function is on. Use the **`schedo`** command to change the tunable parameter.

Dedicated processors

Dedicated processors are whole processors that are assigned to a single partition. If you choose to assign dedicated processors to a logical partition, you must assign at least one processor to that partition. You cannot mix shared processors and dedicated processors in the same partition.

By default, a powered-off logical partition using dedicated processors will have its processors available to the shared processing pool. When the processors are in the shared processing pool, an uncapped partition that needs more processing power can use the idle processing resources. However, when you power on the dedicated partition while the uncapped partition is using the processors, the activated partition will regain all of its processing resources. If you want to prevent dedicated processors from being used in the shared processing pool, you can disable this function on the HMC by deselecting the **Allow idle processor to be shared** check box in the partition's properties.

Note: The option "Allow idle processor to be shared" is activated by default. It is not part of profile properties and it cannot be changed dynamically.

2.4.2 Shared processor pool overview

A shared processor pool is a group of physical processors that are not dedicated to any logical partition. Micro-Partitioning technology coupled with the POWER Hypervisor facilitates the sharing of processing units between logical partitions in a shared processing pool.

In a shared logical partition, there is no fixed relationship between virtual processors and physical processors. The POWER Hypervisor can use any physical processor in the shared processor pool when it schedules the virtual processor. By default, it attempts to use the same physical processor, but this cannot always be guaranteed. The POWER Hypervisor uses the concept of a home node for virtual processors, enabling it to select the best available physical processor from a memory affinity perspective for the virtual processor that is to be scheduled.

Affinity scheduling is designed to preserve the content of memory caches, so that the working data set of a job can be read or written in the shortest time period possible. Affinity is actively managed by the POWER Hypervisor since each partition has a completely different context. Currently, there is one shared processor pool, so all virtual processors are implicitly associated with the same pool.

Figure 2-3 on page 38 shows the relationship between two partitions using a shared processor pool of a single physical CPU. One partition has two virtual

processors and the other a single one. The figure also shows how the capacity entitlement is evenly divided over the number of virtual processors.

When you set up a partition profile, you set up the desired, minimum, and maximum values you want for the profile. When a partition is started, the system chooses the partition's entitled processor capacity from this specified capacity range. The value that is chosen represents a commitment of capacity that is reserved for the partition. This capacity cannot be used to start another shared partition; otherwise, capacity could be overcommitted.

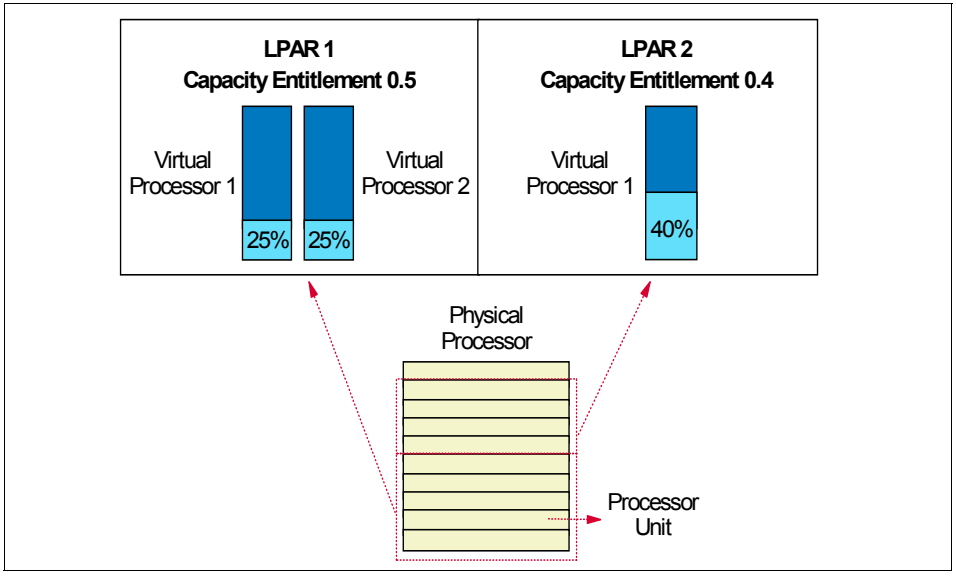


Figure 2-3 Distribution of capacity entitlement on virtual processors

When starting a partition, preference is given to the desired value, but this value cannot always be used because there may not be enough unassigned capacity in the system. In that case, a different value is chosen, which must be greater than or equal to the minimum capacity attribute. Otherwise, the partition will not start.

The entitled processor capacity is distributed to the partitions in the sequence the partitions are started. For example, consider a shared pool that has 2.0 processing units available.

Partitions 1, 2, and 3 are activated in sequence:

- Partition 1 activated
  Min. = 1.0, max = 2.0, desired = 1.5
  Allocated capacity entitlement: 1.5

- Partition 2 activated
  Min. = 1.0, max = 2.0, desired = 1.0
  Partition 2 does not start because the minimum capacity is not met.
- Partition 3 activated
  Min. = 0.1, max = 1.0, desired = 0.8
  Allocated capacity entitlement: 0.5

The maximum value is only used as an upper limit for dynamic operations.

Figure 2-4 shows the usage of a capped partition of the shared processor pool. Partitions using the capped mode are not able to assign more processing capacity from the shared processor pool than the capacity entitlement will allow.

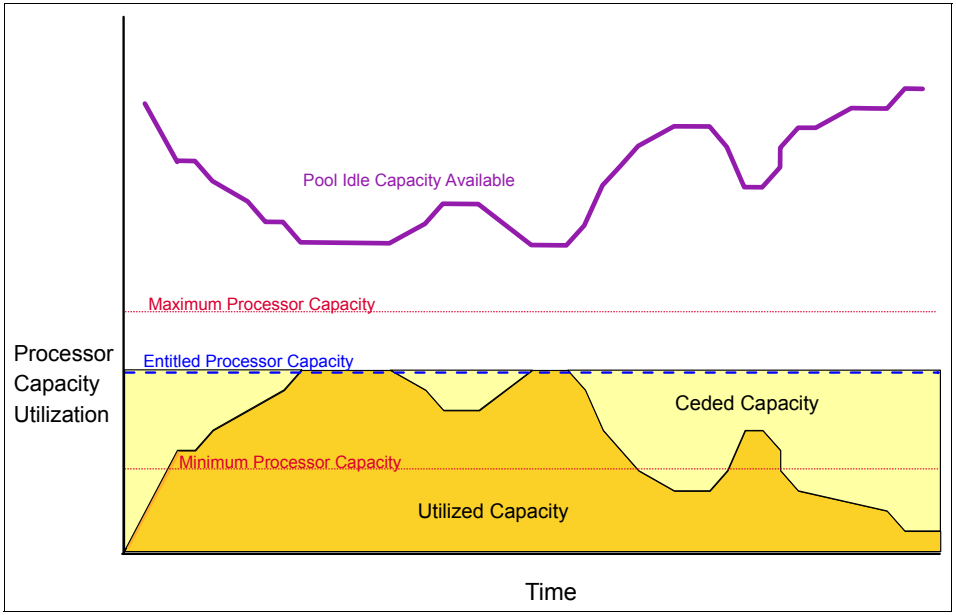


Figure 2-4 Capped shared processor partitions

Figure 2-5 shows the usage of the shared processor pool by an uncapped partition. The uncapped partition is able to assign idle processing capacity if it needs more than the entitled capacity.

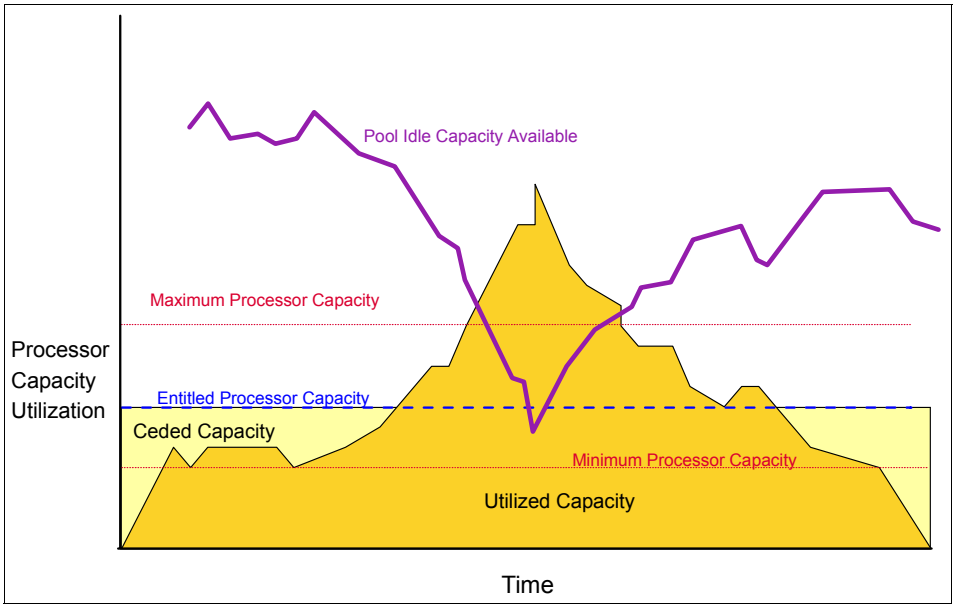


Figure 2-5 Uncapped shared processor partition

2.4.3 Capacity Upgrade on Demand

Capacity Upgrade on Demand (CUoD) adds operational and configuration flexibility for the IBM System p platform. Available as a set of fee-based offerings, CUoD allows additional resources to be added as they are needed. Processors and memory can be brought online to meet increasing workload demands. If the system is configured for dynamic LPAR, this can be accomplished without impacting operations.

When activating a processor featured for Capacity Upgrade on Demand on a system with defined shared processor partitions, the activated processor is automatically assigned to the shared processor pool. You can then decide to add the processor dynamically to a dedicated processor partition or to dynamically add capacity entitlement to the shared processor partitions.

To remove a Capacity Upgrade on Demand processor (for example, when using On/Off Capacity Upgrade on Demand, which enables users to temporarily activate processors), you have to make sure that there are enough processing units left to deactivate the processor. You can dynamically remove the needed capacity entitlement from the partitions.

A type of Capacity Upgrade on Demand is named Reserve CUoD. It represents an *autonomic* way to activate temporary capacity. Reserve CUoD enables you to place a quantity of inactive processors into the server's Shared Processor Pool which then become available to the pool's resource manager. When the server recognizes that the base (purchased/active) processors assigned across uncapped partitions have been 100% utilized, and at least 10% of an additional processor is needed, then a *Processor Day* (good for a 24 hour period) is charged against the Reserve CUoD account balance. Another Processor Day will be charged for each additional processor put into use based on the 10% utilization rule. After a 24-hour period elapses, and there is no longer a need for the additional performance, so no Processor Days will be charged until the next performance spike.

2.4.4 Dynamic processor de-allocation and processor sparing

If a physical processor in the shared processor pool reaches a failure threshold and needs to be taken offline (guarded out), the POWER Hypervisor will analyze the system environment to determine what action will be taken to replace the processor resource. The options for handling this condition are the following:

- If there is a CUoD processor available, the POWER Hypervisor will transparently switch the processor to the shared pool, and no partition loss of capacity would result.
- If there is at least 1.0 unallocated processor capacity available, it can be used to replace the capacity lost due to the failing processor.

If not enough unallocated resource exists, the POWER Hypervisor will determine how much capacity each partition must lose to eliminate the 1.00 processor units from the shared pool. As soon as each partition varies off the processing capacity and virtual processors, the failing processor is taken offline by the service processor and Hypervisor.

The amount of capacity that is requested to each micro-partition is proportional to the total amount of entitled capacity in the partition. This is based on the amount of capacity that can be varied off, which is controlled by the minimum processing capacity of the partition defined in the attribute *min* in the partition profile.

2.4.5 Dynamic partitioning

Partitions with AIX 5L Version 5.2 are supported on servers with dedicated processors. They also support the dynamic movement of the following resources:

- One dedicated processor
- 256 MB memory region
- One I/O adapter slot

A partition with AIX 5L Version 5.3 consists of dedicated processors or shared processors with a specific capacity entitlement running in capped or uncapped mode, dedicated memory region, and virtual or physical I/O adapter slots. All these resources can be dynamically changed.

For dedicated processor partitions, it is only possible to dynamically add, move, or remove whole processors. When you dynamically remove a processor from a dedicated partition, it is then assigned to the shared processor pool.

For shared processor partitions, it is also possible to dynamically:

- Remove, move, or add entitled shared processor capacity.
- Change the weight of the uncapped attribute.
- Add and remove virtual processors.
- Change mode between capped and uncapped processing.

2.4.6 Shared processor considerations

The following considerations must be taken into account when implementing shared processor partitions:

- The minimum size for a shared processor partition is 0.1 processing units of a physical processor. So the number of shared processor partitions you can create for a system depends mostly on the number of processors in a system.
- The maximum number of partitions in a server is 254.
- The maximum number of virtual processors in a partition is 64.
- The minimum number of processing units you can have for each virtual processor depends on the server model. The maximum number of processing units that you can have for each virtual processor is always 1.00. This means that a logical partition cannot use more processing units than the number of virtual processors that it is assigned, even if the logical partition is uncapped.
- A mix of dedicated and shared processors within the same partition is not supported.

- If you dynamically remove a virtual processor, you cannot specify an identification for particular virtual processor to be removed. The operating system will choose the virtual processor to be removed.
- Shared processors may render AIX 5L affinity management useless. AIX 5L will continue to utilize affinity domain information as provided by firmware to build associations of virtual processors to memory, and it will continue to show preference to redispatching a thread to the virtual processor that it last ran on.
- An uncapped partition with a weight of 0 has the same performance impact as a partition that is capped. The HMC can dynamically change either the weight or a partition from capped to uncapped.

Dispatching of virtual processors

There is additional processing associated with the maintenance of online virtual processors, so you should carefully consider their capacity requirements before choosing values for these attributes.

Under AIX 5L V5.3 ML3, a new feature is introduced to help manage idle virtual processors.

Virtual processors have dispatch latency since they are scheduled. When a virtual processor is made runnable, it is placed on a run queue by the POWER Hypervisor, where it waits until it is dispatched. The time between these two events is referred to as dispatch latency.

The dispatch latency of a virtual processor depends on the partition entitlement and the number of virtual processors that are online in the partition. The capacity entitlement is equally divided among these online virtual processors, so the number of online virtual processors impacts the length of each virtual processor's dispatch. The smaller the dispatch cycle, the greater the dispatch latency.

At the time of writing, the worst case virtual processor dispatch latency is 18 milliseconds, since the minimum dispatch cycle that is supported at the virtual processor level is one millisecond. This latency is based on the minimum partition entitlement of 1/10 of a physical processor and the 10 millisecond rotation period of the Hypervisor's dispatch wheel. It can be easily visualized by imagining that a virtual processor is scheduled in the first and last portions of two 10 millisecond intervals. In general, if these latencies are too great, then clients may increase entitlement, minimize the number of online virtual processors without reducing entitlement, or use dedicated processor partitions.

Number of virtual processors

In general, the value of the minimum, desired, and maximum virtual processor attributes should parallel those of the minimum, desired, and maximum capacity attributes in some fashion. A special allowance should be made for uncapped partitions, since they are allowed to consume more than their entitlement.

If the partition is uncapped, then the administrator may want to define the desired and maximum virtual processor attributes greater than the corresponding entitlement attributes. The exact value is installation-specific, but 50 to 100 percent more is a reasonable number.

Table 2-3 shows several reasonable settings of number of virtual processor, processing units, and the capped and uncapped mode.

Table 2-3 Reasonable settings for shared processor partitions

Min VPs[a]	Desired VPs	Max VPs	Min PU[b]	Desired PU	Max. PU	Capped
1	2	4	0.1	2.0	4.0	Y
1	3 or 4	6 or 8	0.1	2.0	8.0	N
2	2	6	2.0	2.0	6.0	Y
2	3 or 4	8 or 10	2.0	2.0	10.0	N

a - Virtual processors
b - Processing units

Virtual and physical processor relationship

In a shared partition, there is not a fixed relationship between the virtual processor and the physical processor. The POWER Hypervisor will try to use a physical processor with the same memory affinity as the virtual processor, but it is not guaranteed. Virtual processors have the concept of a home physical processor. If the Hypervisor cannot find a physical processor with the same memory affinity, then it gradually broadens its search to include processors with weaker memory affinity, until it finds one that it can use. As a consequence, memory affinity is expected to be weaker in shared processor partitions.

Workload variability is also expected to be increased in shared partitions because there are latencies associated with the scheduling of virtual processors and interrupts. SMT may also increase variability, since it adds another level of resource sharing that could lead to a situation where one thread interferes with the forward progress of its sibling.

Therefore, if an application is cache-sensitive or cannot tolerate variability, then it should be deployed in a dedicated partition with SMT disabled. In dedicated partitions, the entire processor is assigned to a partition. Processors are not shared with other partitions, and they are not scheduled by the POWER Hypervisor. Dedicated partitions must be explicitly created by the system administrator using the Hardware Management Console.

Processor and memory affinity data is only provided in dedicated partitions. In a shared processor partition, all processors are considered to have the same affinity. Affinity information is provided through RSET APIs.

2.5 Introduction to simultaneous multithreading

Conventional processors run instructions from a single instruction stream, and despite micro architectural advances, execution unit utilization remains low in today's microprocessors. It is not unusual to see average execution unit utilization rates of approximately 25 percent in many environments.

SMT as implemented by the POWER5 processor fetches instructions from more than one thread. What differentiates this implementation is its ability to schedule instructions for execution from all threads concurrently. With SMT, the system dynamically adjusts to the environment, allowing instructions to run from each thread if possible, and allowing instructions from one thread to utilize all the execution units if the other thread encounters a long latency event.

2.5.1 POWER5 processor SMT

In SMT mode, the POWER5 processor uses two separate program counters, one for each threads. Instruction fetches alternate between the two threads. The two threads share the instruction cache.

Not all applications benefit from SMT. For this reason, the POWER5 processor supports *single-threaded* (ST) execution mode. In this mode, the POWER5 processor gives all the physical processor resources to the active thread, the POWER5 processor uses only one program counter and fetches instructions for that thread every cycle.

It is possible to switch between ST and SMT modes dynamically (without rebooting) in AIX 5L V5.3. Linux partitions require a restart to change SMT mode.

The benefit of SMT is greatest where there are numerous concurrently executing threads, as is typical in commercial environments, for example, for a Web server or database server. Data-intensive, memory bound, or single-threaded high performance computing workloads will generally perform better with ST.

2.5.2 SMT and AIX 5L

The AIX 5L operating system scheduler dispatches execution threads to logical processors. Dedicated and virtual processor have one or two logical processors depending whether SMT is enabled or not. With SMT enabled, both logical processors are always in the same partition. Figure 2-6 shows the relationship between physical, dedicated, and logical processors for dedicated and shared-processor partitions. It is possible to have some shared-processor partitions with SMT enabled and others with SMT disabled at the same time.

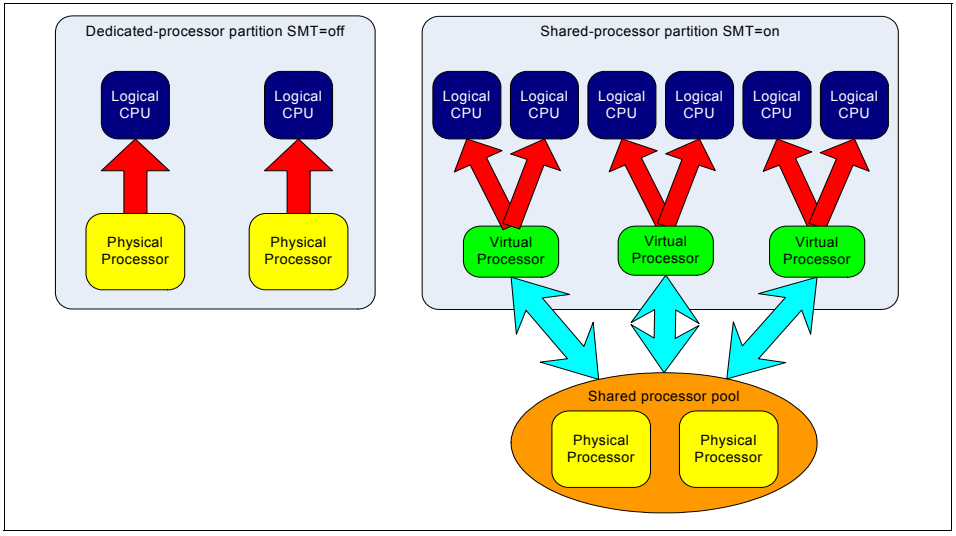


Figure 2-6 Physical, virtual, and logical processors

SMT control in AIX 5L

SMT enablement is controlled by the AIX 5L **`smtctl`** command or with the system management interface tool (SMIT). SMT can be enabled or disabled dynamically on a logical partition or for the next operating system reboot.

Setting SMT mode using the command line

The **smtctl** command must be run by users with root authority.

The two flags associated with **smtctl** are -m and -w; they are defined as follows:

-m off Will set SMT mode to disabled.

-m on Will set SMT mode to enabled.

-w boot Makes the SMT mode change effective on the next and subsequent reboots.

-w now Makes the mode change effective immediately, but will not persist across reboot.

The **smtctl** command does not rebuild the boot image. If you want to change the default SMT mode of AIX 5L, the **bosboot** command must be used to rebuild the boot image. The boot image in AIX 5L Version 5.3 has been extended to include an indicator that controls the default SMT mode.

Note: If neither the -w boot nor the -w now flags are entered, the mode change is made immediately and will persist across reboots. The boot image must be remade with the **bosboot** command in order for a mode change to persist across subsequent boots, regardless of -w flag usage.

The **smtctl** command entered without a flag will show the current state of SMT in the partition. An example of the **smtctl** command follows:

```
# smtctl

This system is SMT capable.

SMT is currently enabled.

SMT boot mode is set to enabled.
Processor 0 has 2 SMT threads
SMT thread 0 is bound with processor 0
SMT thread 1 is bound with processor 0
```

Setting SMT mode using SMIT

Use the **smitty smt** fast path to access the SMIT SMT control panel. From the main SMIT panel, the selection sequence is **Performance & Resource Scheduling** → **Simultaneous Multi-Threading Mode** → **Change SMT Mode**. Figure 2-7 shows the SMIT SMT panel.

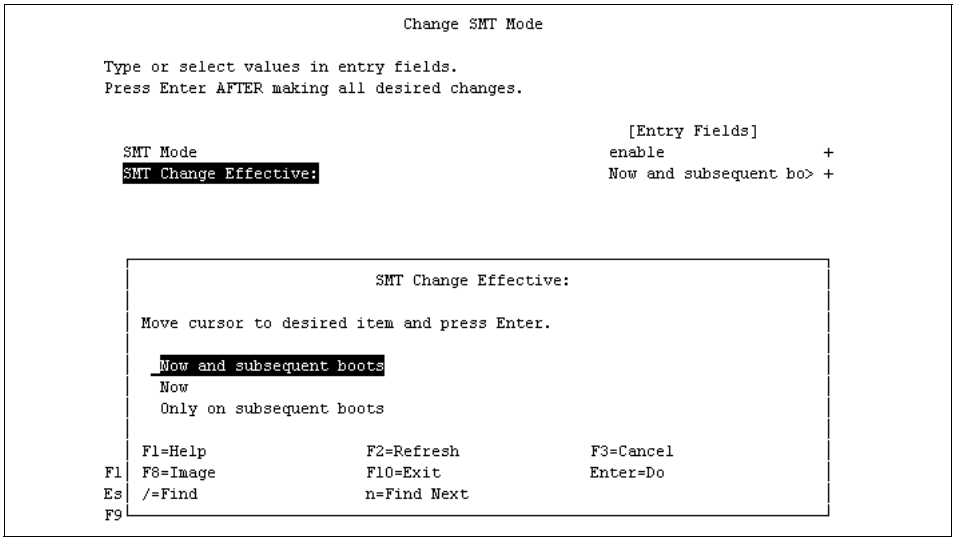


Figure 2-7 SMIT SMT panel with options

SMT performance monitor and tuning

AIX 5L Version 5.3 includes commands or extended options to existing commands for the monitoring and tuning of system parameters in SMT mode.

SMT monitoring

The SMT behavior requires the operating system to provide statistics on the use of the particular logical processors. The **mpstat** command is used to display performance statistics for all logical processors operating in the logical partition. The **mpstat** command is described in “Logical processor tools” on page 326.

2.5.3 SMT control in Linux

To enable or disable SMT at boot, use the following boot option at the boot prompt:

```
boot: linux smt-enabled=on
```

Change the on to off to disable SMT at boot time. The default is SMT on.

2.6 Introduction to the POWER Hypervisor

The POWER Hypervisor is the foundation of the IBM Virtualization Engine system technologies implemented in the POWER5 processor-based family of products. Combined with features designed into the POWER5 processor, the POWER Hypervisor delivers functions that enable other system technologies including Micro-Partitioning, virtual processors, IEEE VLAN compatible virtual switch, virtual SCSI adapters, and virtual consoles.

The POWER Hypervisor is a firmware layer sitting between the hosted operating systems and the server hardware, as shown in Figure 2-8. The POWER Hypervisor is always installed and activated, regardless of system configuration. The Hypervisor has no processor resources assigned to it.

The POWER Hypervisor performs the following tasks:

- Enforces partition integrity by providing a security layer between logical partitions.
- Provides an abstraction layer between the physical hardware resources and the logical partitions using them. It controls the dispatch of virtual processors to physical processors. It saves and restores all processor state information during logical processor context switch.
- Controls hardware I/O interrupts management facilities for logical partitions.

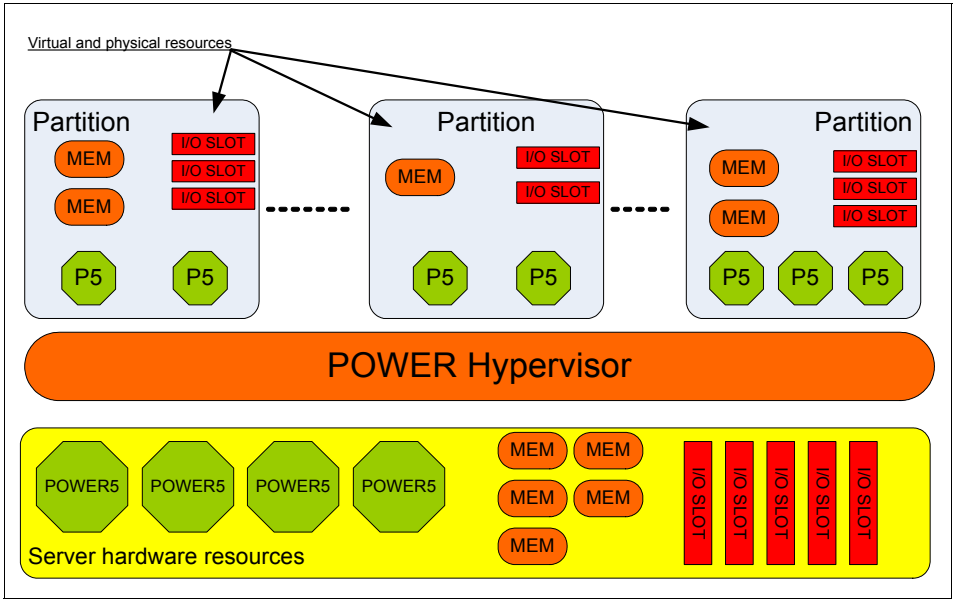


Figure 2-8 The POWER Hypervisor abstracts the physical server hardware

The Hypervisor firmware and the hosted operating systems communicate with each other through Hypervisor calls (hcalls).

The POWER Hypervisor allows multiple instances of operating systems to run on POWER5 servers concurrently. The supported operating systems are listed in 1.4, “Operating system support” on page 9.

2.6.1 POWER Hypervisor virtual processor dispatch

Shared-processor partitions are given one or more virtual processors to run their workload on. The number of virtual processors in any partition and in all partitions does not necessarily have any correlation to the number of physical processors in the shared-processor pool except that each physical processor can support, at most, ten virtual processors.

The POWER Hypervisor manages the distribution of available physical processor cycles to all the processors in the shared pool. The POWER Hypervisor uses a 10 ms dispatch cycle; each virtual processor is guaranteed to get its entitled share of processor cycles during each 10 ms dispatch window.

To optimize physical processor usage, a virtual processor will yield a physical processor if it has no work to run or enters a wait-state, such as waiting for a lock or for I/O to complete. A virtual processor may yield a physical processor through a Hypervisor call.

Dispatch mechanism

To illustrate the mechanism, consider three partitions with two, one, and three virtual processors. These six virtual processors are mapped to two physical POWER5 cores, as shown in Figure 2-9 on page 51.

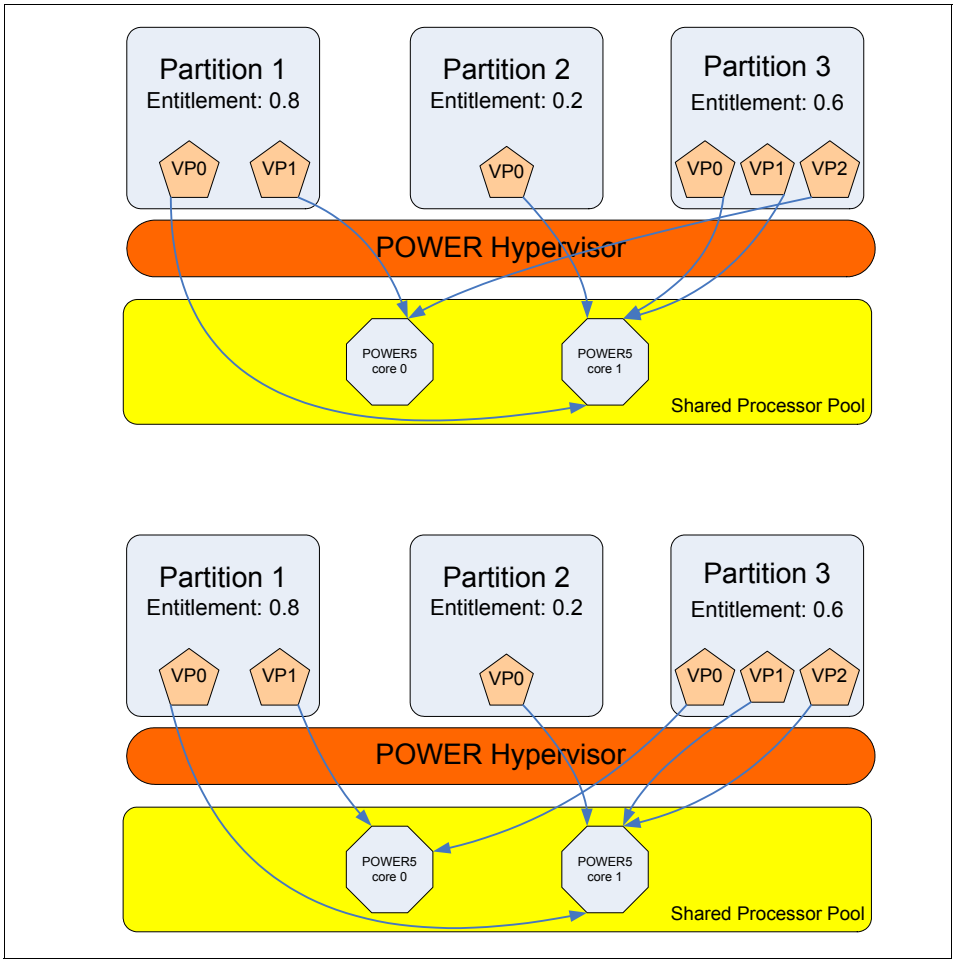


Figure 2-9 Virtual processor to physical processor mapping: pass 1 and pass 2

Figure 2-10 on page 52 shows two POWER Hypervisor dispatch cycles for two partitions with a total of six virtual processors dispatched on to two physical CPUs.

Partition 1 is defined with an entitlement capacity of 0.8 processing units, with two virtual processors. This allows the partition the equivalent of 80 percent of one physical processor for each 10 ms dispatch window for the shared processor pool. The workload uses 40 percent of each physical processor during each dispatch interval.

Partition 2 is configured with one virtual processor and a capacity of 0.2 processing units, entitling it to 20 percent usage of a physical processor during each dispatch interval. In this example, a worst case dispatch latency is shown for this virtual processor, where the 2 ms are used in the beginning of dispatch interval 1 and the last 2 ms of dispatch interval 2, leaving 16 ms between processor allocation.

Note: It is possible for a virtual processor to be dispatched more than one time during a dispatch interval. In the first dispatch interval, the workload executing on virtual processor 1 in LPAR 1 is discontinuous on the physical processor resource. This can happen if the operating system confers cycles, and is reactivated by a prod hcall.

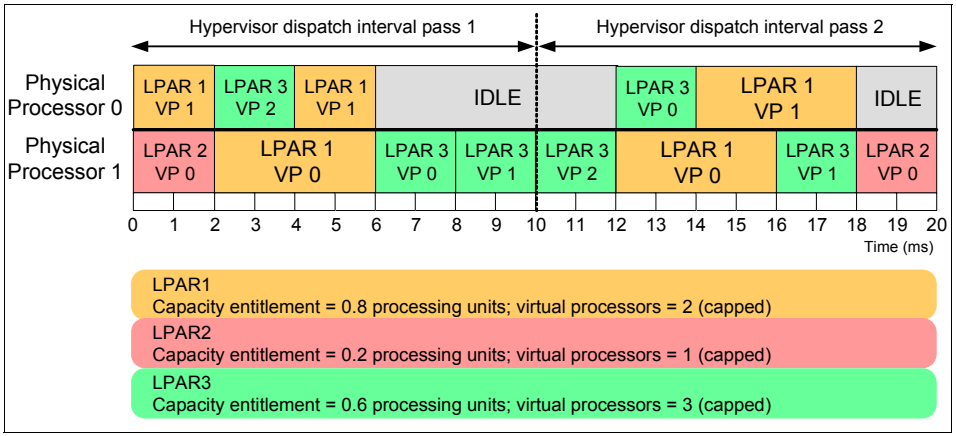


Figure 2-10 Micro-Partitioning processor dispatch

Partition 3 contains three virtual processors, with an entitled capacity of 0.6 processing units. Each of the partition's three virtual processors consumes 20 percent of a physical processor in each dispatch interval, but in the case of virtual processor 0 and 2, the physical processor they run on changes between dispatch intervals.

Processor affinity

The POWER Hypervisor is designed to dispatch threads on the same physical processor as it ran on in the previous dispatch cycle. This is called processor affinity. The POWER Hypervisor will always first try to dispatch the virtual processor on the same physical processor as it last ran on, and, depending on resource utilization, will broaden its search out to the other processor on the POWER5 chip, then to another chip on the same MCM, then to a chip on another

MCM. The goal of processor affinity is to try to keep a thread close to its data and to optimize the use of the caches.

System monitoring and statistics

The sharing of systems resources, such as with micro-partitions and SMT challenges the traditional AIX 5L performance collection and reporting tools. The POWER5 architecture introduced a new register (in each core): the Processor Utilization Resource Register (PURR). This register provides the partition with an accurate cycle count to measure activity during time slices dispatched on a physical processor.

The PURR and the performance collection and reporting tools are discussed in 5.5, “Monitoring a virtualized environment” on page 321.

Monitoring Hypervisor hcalls

AIX 5L Version 5.3, provides the **`lparstat`** and **`mpstat`** commands to display the Hypervisor and virtual processor affinity statistics. These commands are discussed in detail in 5.5, “Monitoring a virtualized environment” on page 321.

2.6.2 POWER Hypervisor and virtual I/O

The POWER Hypervisor does not own any physical I/O devices nor does it provide virtual interfaces to them. All physical I/O devices in the system are owned by logical partitions.

Note: Shared I/O devices are owned by the Virtual I/O Server, which provides access to the real hardware upon which the virtual device is based.

To support virtual I/O, the POWER Hypervisor provides:

- Control and configuration structures for virtual adapters
- Controlled and secure access to physical I/O adapters between partitions
- Interrupt virtualization and management

I/O types supported

Three types of virtual I/O adapters are supported by the POWER Hypervisor:

- SCSI
- Ethernet
- System Port (virtual console)

Note: The Virtual I/O Server supports optical devices. These are presented to client partitions as a virtual SCSI device.

Virtual I/O adapters are defined by system administrators during logical partition definition. Configuration information for the virtual adapters is presented to the partition operating system by the system firmware.

Virtual SCSI is covered in detail in 2.9, “Virtual SCSI introduction” on page 89; virtual Ethernet and the shared Ethernet adapter is discussed in 2.8, “Virtual and Shared Ethernet introduction” on page 70.

2.6.3 System port (virtual TTY/console support)

Each partition needs to have access to a system console. Tasks such as operating system install, network setup, and some problem analysis activities require a dedicated system console. The POWER Hypervisor provides virtual console using a virtual TTY or serial adapter and a set of Hypervisor calls to operate on them.

Depending on the system configuration, the operating system console can be provided by the Hardware Management Console (HMC) virtual TTY, or from a terminal emulator connected to physical system ports on the system’s service processor.

2.7 Software licensing in a virtualized environment

In an effort to expand on demand offerings, being consistent with the deployment and adoption of tools to virtualize the IBM System p platforms, IBM and several independent software vendors (ISVs) have considered new licensing methods to better fit client needs when consolidating business applications along with required middleware in virtualized environments.

2.7.1 IBM i5/OS licensing

Clients who want to run i5/OS on IBM System p must acquire license entitlements for i5/OS, which is priced and licensed on a per processor basis. Program licensing (such as processor counting methodology, aggregation, and transferability) terms for i5/OS are the same on System p servers as on System i servers.

There is an upper limit to the number of i5/OS processor licenses to share between partitions that the client can entitle and run on a System p platform. An

p5-570 can run up to one processor license of i5/OS and Models 590 and 595 can run up to two processor licenses of i5/OS.

The rest of this section covers the main aspects about software licensing for System p systems configured with IBM AIX 5L and Linux operating systems. For more information about software licensing for i5/OS partitions on an IBM System p system, contact your IBM sales representative.

2.7.2 Software licensing methods for UNIX operating systems

Although a discussion of software licensing on a per-server basis is not part of this redbook, we should mention that the term *server* is defined by the ISV for licensing purposes. In most cases, since each partition has its own operating system and hardware resources (either partition with dedicated processors or micro-partition), the partition where the software is installed is considered the server. In this case, the software is charged one time for each partition where it is installed and running, independently of the processors in the partition.

Many of the new licensing methods are offered on a per-processor basis, so it is important to determine the quantity of processors in which the software is running to determine the licensing requirements of such software, either physical or virtual processors, to determine software charges. The rest of this section applies only for per-processor based licensing methods.

Clients should use the quantity of active processors declared by IBM in an IBM System p5 server as the cores to license in a per-processor basis. For example, if IBM configures a p5-570 to have eight installed processors, six of them active, the p5-570 should have six active cores for licensing purposes from a total of eight installed cores, regardless the quantity of chips or processors cards.

2.7.3 Licensing factors in a virtualized system

Clients planning for the purchase of software for a partitioned IBM System p platform should understand the drivers for licensing, since charges depend on the way the processors in the system are used by the operating systems.

Active processors and hardware boundaries

In a per-processor basis, the boundary for licensing is the quantity of active processors in the system (assigned and unassigned), since only active processors can be real engines for software. It works for any type of per-processor based software.

Most ISVs consider partitions with dedicated processors on an System p platform as independent servers. In this case, software licenses must be obtained for all processors in a partition and for all partitions where the software is installed. IBM uses this classification for partitions with dedicated processors for selected IBM software.

The quantity of processors for a certain partition can vary over the time because of dynamic LPAR operations, but the overall licenses should equal or exceed the total number of processors using the software at a time.

Unassigned processors and dedicated partitions

In a system with only dedicated partitions (no shared processor pool because no APV is activated in the system), active processors in the unassigned pool can increase the quantity of processors for software in a partition if they are added, even temporarily, when assigned with dynamic LPAR operations. Clients should note that ISVs can require licenses for the maximum number of processors for each of the partitions where the software is installed (the maximum quantity of processors in the partition profile).

IBM charges for the quantity of processors in a partition with dedicated processors, even temporary ones with dynamic LPAR operations. Since the licenses are purchased as one-time charges, IBM software licensing is incremental. For example, a client can install an AIX 5L partition on a system with three processors for DB2® from a pool of eight active processors (no more partitions with AIX 5L and DB2), later increases that partition with two more processors, and later releases one processor from the partition; in this case, the client has incremental licensing starting with three processors for AIX 5L and DB2, then needs to add licenses for two more processors for both AIX 5L and DB2, for a total of five processor licenses.

Processors with Capacity Upgrade on Demand

Processors in the CUoD pool do not count for licensing purposes until:

- They become temporarily or permanently active as part of the shared processor pool in systems with Advanced POWER Virtualization.
- They become temporarily or permanently active and assigned in systems with no Advanced POWER Virtualization.

Clients can provision licenses of selected IBM software for temporary use on their systems. Such licenses can be used on a per-processor/day basis to align with the possible temporary use of CUoD processors in existing or new AIX 5L or Linux partitions. For example, temporary AIX 5L based software licenses can be used either for active processors (unassigned processors in systems with no APV), new partitions (created from unassigned processors or from the shared

processor pool in systems with APV), permanent activated CUoD processors, or temporary activated On/Off processors.

For more information about processors on demand On/Off, refer to 2.4.3, “Capacity Upgrade on Demand” on page 40.

Processors in the shared processor pool

All the processors that become active and non-dedicated are part of the shared processor pool; thus, the quantity of processors in the shared processor pool equals the quantity of active non-dedicated processors in the system.

Entitled capacity for micro-partitions

The entitled capacity is the real capacity in terms of computing power that a partition is given even when it starts up or by executing dynamic LPAR operations when running. The entitled capacity applies only at runtime and is the guaranteed amount of processing units a micro-partition can consume. There are two ways to define a shared processor partition: capped and uncapped mode.

For a capped micro-partition, the entitled capacity is also the maximum processing power the partition can use and its first value was given at start up. Keep in mind that with dynamic LPAR operations, it is possible to add processing units to the entitled capacity depending on both system resources in the shared processor pool and the maximum quantity of processing units allowed for the partition.

For an uncapped micro-partition, the entitlement capacity given to the partition is not limiting the access to processing power. An uncapped micro-partition can use more than the entitled capacity when there are free resources in the shared processor pool. The limiting factor for uncapped micro-partition is the number of defined virtual processors. The micro-partition can use as many as there are physical processors in the shared processor pool, since each virtual processor is dispatched to a physical processor at a time.

Virtual processors for micro-partitions

The virtual processors are created for the operating systems of micro-partitions to enable the mechanism that shares physical processors in the shared processing pool between such micro-partitions. The operating system in a micro-partition handles virtual processors as discrete entities (system objects) and the hardware dispatches the virtual processors to physical processors in a time-share basis.

When a partition is running as uncapped and exceeds the maximum quantity of processing units, there is a higher probability that their virtual processors get dispatched to physical processors simultaneously. So, for a single uncapped micro-partition, the maximum quantity of virtual processors running on the physical processors of the shared pool at a point of time is the lowest between the number of virtual processors and physical processors in the shared pool.

Entitled capacity versus virtual processors

From the definitions, the maximum computing power given to the software that should be licensed in a capped micro-partition is always the entitlement capacity. It is expandable to the maximum quantity of processing units defined in the micro-partition profile. The entitled capacity becomes the driver to license software for a capped micro-partition and can be measured for auditing purposes to determine the real use of the system and calculate possible requirements of additional licensing.

For uncapped micro-partitions, the maximum computing power given to a software program depends on the number of virtual processors in the operating system and the number of physical processors in the shared processor pool. So, the number of virtual processors becomes the driver to license software for uncapped micro-partitions, the maximum being the number of physical processors in the shared pool.

Figure 2-11 on page 59 shows the boundaries for per-processor software licensing.

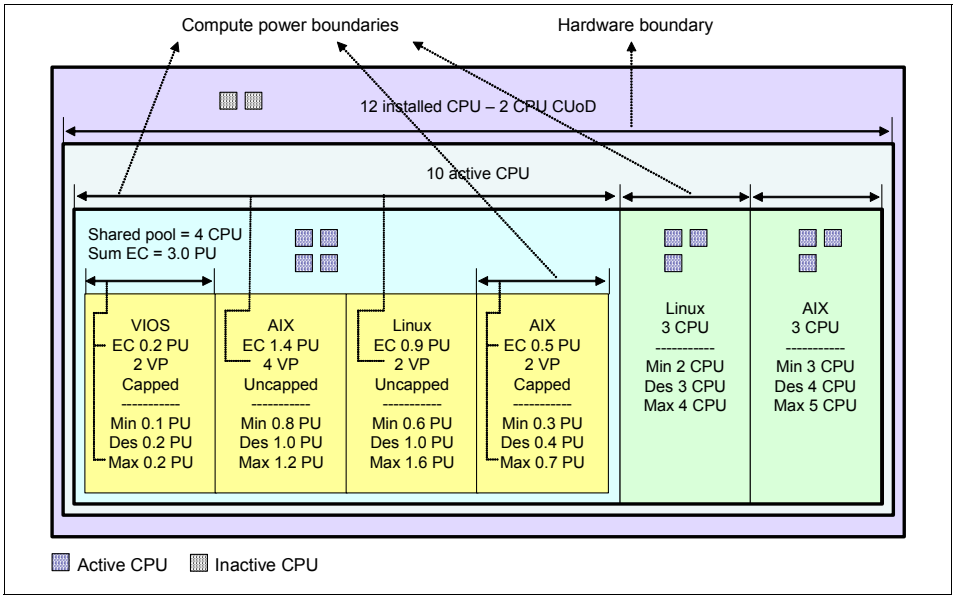


Figure 2-11 Boundaries for software licensing on a per-processor basis

2.7.4 License planning and license provisioning of IBM software

Clients acquiring IBM software licensed on a per-processor basis should consider the following licensing criteria (for non-IBM software, users should contact the respective ISV sales representative):

- IBM AIX 5L and selected pSeries software programs are licensed per-processor and licenses are required only for those partitions where the AIX operating system and IBM programs are installed.
- IBM AIX 5L and selected pSeries software programs may offer per-processor/day licensing for temporary use.
- Linux distributions have their own licensing methods and licenses are required only for those partitions where the operating system is installed.
- The Advanced POWER Virtualization software (VIO and PLM) is licensed per-processor and licenses are required for all the systems.
- The Advanced POWER Virtualization software (VIO and PLM) apply for per-processor/day licensing for temporary use and licenses are required in the activation of CUoD On/Off processors.

- Selected IBM software programs eligible under IBM Passport Advantage® and licensed on a per-processor basis may qualify for Sub-Capacity terms, so licenses are required only for those partitions where the programs are installed. To be eligible for Sub-Capacity terms, the client must agree to the terms of the IBM International Passport Advantage Agreement Attachment for Sub-Capacity Terms.
- At the time of writing, selected IBM System p software programs and IBM software programs are eligible for temporary On/Off. To be eligible for On/Off Capacity Upgrade on Demand pricing, clients must be enabled for temporary capacity on the corresponding hardware, and the required contract, Amendment for iSeries and pSeries Temporary Capacity Upgrade on Demand Software, must be signed prior to use.
- For IBM AIX 5L and selected IBM software, program licenses can be shared between capped micro-partitions (the only type of partitions with fractions of a processor for licensing); thus, several micro-partitions using the IBM software with an aggregate capacity in terms of processing units (planned capacity at software installation, entitled capacity at runtime) can use less processor licenses than if considered separately.

Note: At the time of writing, processor license sharing applies only to AIX 5L, HACMP™, and selected IBM programs, and only for capped micro-partitions in the shared processor pool. Other AIX 5L related IBM programs may apply for license sharing. Contact your IBM sales representative for the current status of selected AIX 5L-related IBM products that apply for license sharing between capped micro-partitions

Only selected IBM software for the System p are eligible for on demand licensing. When planning for software charge in a per-processor basis for the systems, the client should also differentiate between:

Initial planned licensing
: The client calculates the base license entitlements based on the licensing rules and the drivers for licensing (everything except entitlement capacity for operating systems). The client purchases processor licenses based on the planned needs, and the maximum is the number of active processors in the system. The client can also purchase temporary On/Off licenses of selected pSeries related software.

Additional licensing
: The client checks the real usage of software licenses and planned needs and calculates the additional license entitlements (temporary On/Off licenses also) based on

the licensing rules and the drivers for licensing (including entitlement capacity for operating systems).

On demand licensing

The client contacts IBM or a Business Partner for the submission of a Passport Advantage Program enrollment. The client follows the procedures of the licensing method (sub-capacity licensing for selected IBM Passport Advantage eligible programs) and establishes a monitoring system with IBM Tivoli® License Manager for IBM software. IBM is notified about the usage of software licenses and the client is notified by IBM to adjust the license entitlements when apply.

For the initial license plan, the client can use the following approach, as summarized in Figure 2-12 on page 62:

1. Determine the quantity of processors that need software licenses for dedicated partitions (plan between desired and maximum).
2. Determine the quantity of processor units that need software licenses for capped micro-partitions (plan between desired and maximum). Round processor units to next integer value. Verify if the software program allows for processor license sharing.
3. Determine the quantity of virtual processors that need software licenses for uncapped micro-partitions (plan between desired and maximum number of VPs and processors in the shared pool).
4. Sum individual processor quantities (integer processor units for capped micro-partitions, number of virtual processors for uncapped micro-partitions) and take the lowest between the sum and processors in the shared pool.
5. Sum planned licenses for dedicated partitions and planned licenses for the shared pool and take the lowest between the sum and planned active processors (active at installation plus CUoD activations).

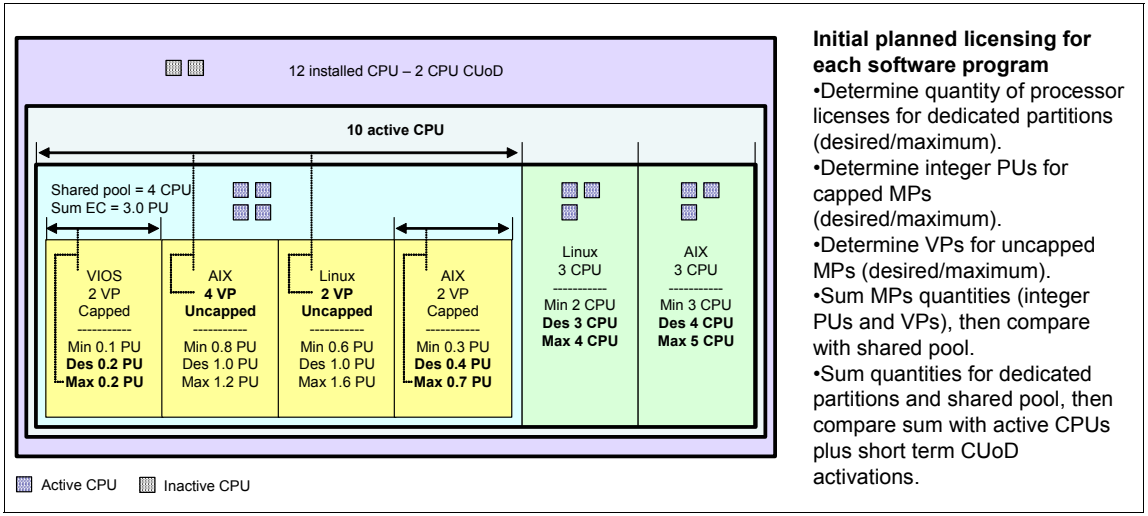


Figure 2-12 Example of initial licensing planning

2.7.5 Sub-capacity licensing for IBM software

Sub-capacity licensing allows for the licensing of a software program for use on less than the full processor capacity of the System p5 systems when the software program is used within two or more partitions.

Sub-capacity licensing allows a client to benefit from hardware partitioning that includes advanced IBM virtualization capabilities, such as shared processor pools, Micro-Partitioning (allowing for processor license sharing), and dynamic reallocation of resources with flexible software licensing support.

The following are important considerations for sub-capacity licensing of IBM software:

- The sub-capacity licensing program applies to selected IBM Passport Advantage programs licensed on a per-processor basis.
- The client agrees to the terms of an attachment to their International Passport Advantage Agreement and submits a Passport Advantage Enrollment Form to IBM or Business Partner.
- The client must use IBM Tivoli License Manager for IBM Software to monitor program use and submit to IBM an IBM a *use report* each calendar quarter.
- At the time of this writing, System p5 software is not required to be monitored by ITLM.

IBM Tivoli License Manager (ITLM) allows the program to monitor the use of processor licenses in all the partitions where the monitored IBM software program is installed (partitions with dedicated processors, and capped and uncapped micro-partitions).

For the partitions in the System p5 server where it is installed, ITLM monitors the overall usage of processor licenses, detects changes in system configuration from dynamic LPAR operations and CUoD and CUoD On/Off processor activations, and notifies IBM periodically about such changes and differences in licensing. Since IBM software program licenses are incremental, the client is required to purchase additional IBM software licenses when the use has exceeded the overall entitlement license capacity (Figure 2-13).

For capped micro-partitions, ITLM becomes the tool that constantly measures the entitlement capacity of the operating systems and allows you to match the initial licensing provisioning versus the real consumption in the shared processing pool.

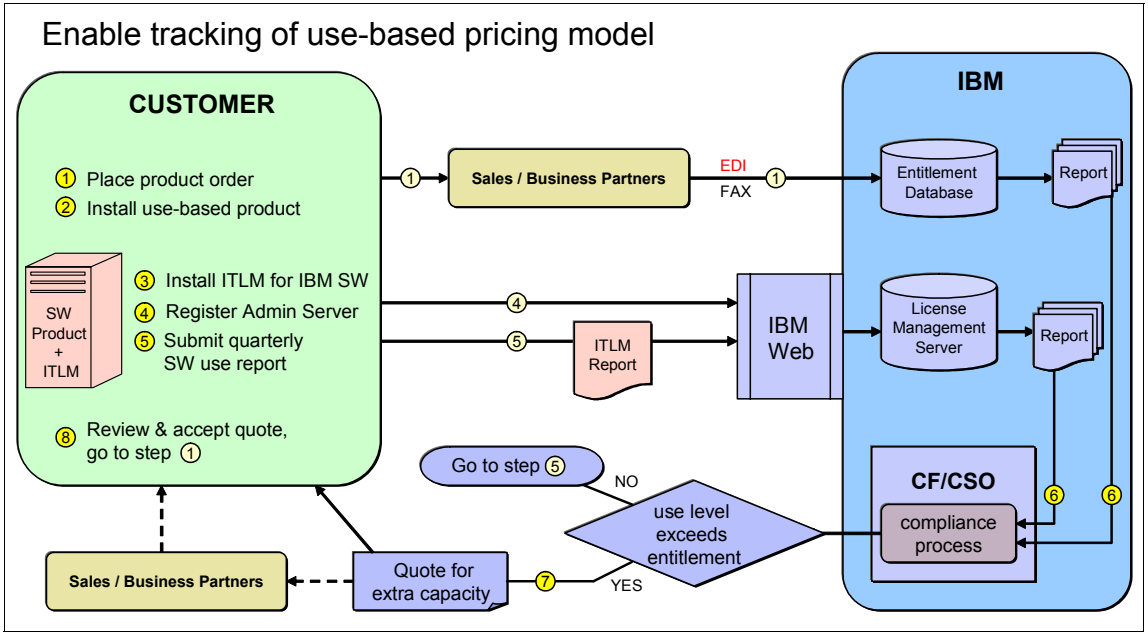


Figure 2-13 IBM Tivoli License Manager role in compliance

For additional information about terms and conditions for sub-capacity licensing of selected IBM software for your geography, contact your IBM representative or visit:

http://www.ibm.com/software/passportadvantage

For more information about the use of IBM Tivoli License Manager to monitor IBM software, refer to *Introducing IBM Tivoli License Manager,* SG24-6888.

Table 2-4 has a summary of the main features of selected IBM software for System p5 systems.

Table 2-4 Selected IBM software programs and licensing features

IBM software program	Entire system (active CPUs)	CUoD On/Off (temporary)	Shared processor licenses	Subcapacity licensing contract
VIO V1.1, V1.2, V1.3	X	X	X	-
PLM V1	X	X	X	-
AIX 5L V5.2 and V5.3	-	X	X	-
Performance AIDE V3	-	X	-	-
GPFS™ V21	-	X	-	-
HACMP V5.2 and V5.3	-	X	X	-
LoadLeveler® V3	-	X	-	-
Parallel Environment for AIX V4	-	X	-	-
Parallel ESSL for Linux pSeries V31	-	X	-	-
Parallel ESSL for AIX V3	-	X	-	-
CSM V1	-	X	-	-
ESSL V4	-	X	-	-
Lotus® Domino® V6.5	-	X	-	-
TXSeries® V5.0	-	-	X	X
WebSphere® MQ V5.3 and V6.0	-	X	X	X
WebSphere MQSeries® Workflow V3.5 and V3.6	-	X	X	X
WebSphere Application Server V5.0, V5.1, V6.0	-	X	X	X
WebSphere Data Interchange V3.2	-	-	X	X
WebSphere Everyplace® Connection Manager without WAP V5.1	-	-	X	X
WebSphere Business Integration Event Broker V5.0	-	-	X	X
WebSphere Business Integration Message Broker V5.0	-	-	X	X
WebSphere Business Integration Message Broker with Rules and Formatter Extension V5.0	-	-	X	X
WebSphere Business Integration Server Foundation, V5.1	-	X	-	-
WebSphere Portal Enable for Multiplatforms V5.0, V5.1	-	X	X	X

IBM software program	Entire system (active CPUs)	CUoD On/Off (temporary)	Shared processor licenses	Subcapacity licensing contract
WebSphere Portal Extend for Multiplatforms V5.0, V5.1	-	X	X	X
WebSphere InterChange Server V4.3	-	X	X	X
DB2 Data Links Manager V8.2	-	X	X	X
DB2 Net Search Extender V8.2	-	X	X	X
DB2 UDB Data Warehouse Enterprise Edition V8.2	-	X	X	X
DB2 UDB Enterprise Server Edition V8.2	-	X	X	X

2.7.6 IBM software licensing

The following scenarios can help you understand how to license System p5 software based on your needs. It should be noted that software licensing is a client responsibility.

Table 2-5 on page 66 shows license planning for a 16-core, with 14 active processors system. In this case, the partitions DLPAR1, DLPAR2, DLPAR5, and DLPAR6 have direct estimations for planned processor licenses. The partitions DLPAR3 and DLPAR4 are capped micro-partitions and their licensing depend on runtime processing units usage; thus, the client plans licenses to exceed real future needs and to take advantage of sharing processor licenses for selected IBM software (AIX 5L, HACMP, and IBM software under sub-capacity licensing contract).

To qualify for this license plan, the client agrees to the IBM International Passport Advantage Agreement Attachment for Sub-Capacity Terms, signs the Sub-capacity licensing program, and installs ITLM to monitor the use of software licenses.

In the example, there is a high probability of running out of software licenses for capped micro-partition DLPAR3 because the estimation of one processor license and a maximum of two processor licenses based on the maximum number of processing units in the partition profile.

Table 2-5 Licensing estimation for initial purchasing of processor licenses

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (inactive)
Operating system	AIX 5L V5.3	Linux	AIX 5L V5.3	AIX 5L V5.3	Linux	AIX 5L V5.3	N/A
Additional IBM System p5 software	HACMP		HACMP	HACMP			
Additional IBM software	DB2	DB2	DB2 / WebSphere Application Server	Domino	WebSphere Application Serve		
Partition type	Dedicated	Dedicated	Micro	Micro	Micro	Micro	
Physical processors	4	3	7				2
Maximum virtual processors used	N/A	N/A	2	3	5	8	
Capped / uncapped	N/A	N/A	Capped	Capped	Uncapped	Uncapped	
Maximum processing units	N/A	N/A	2.0	3.0	5.0 (VPs)	7.0 (Pool)	
Estimated entitled capacity	N/A	N/A	0.8	1.4	2.4	2.4	
Estimated processors for licenses	4	3	1-2	2-3	5	7	
Client planned processor licenses	4	3	1	2	5	7	
AIX processor licenses = 11+1	4		Roundup(0.8+1.4)=3			7	1 On/Off
HACMP processor licenses = 7+1	4		Roundup(0.8+1.4)=3				1 On/Off
DB2 processor licenses = 8	4	3	1				

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (inactive)
WebSphere Application Server processor licenses = 6 +1			1		5		1 On/Off
Domino processor licenses = 14				14 (system)			

Table 2-6 shows the same pSeries configuration six months later. At that time, several events occur:

- The system dynamically adjusts entitlement capacities for micro-partitions.
- The client changes several partition system profiles.
- Dynamic LPAR operations move processing units between partitions and On/Off software licenses are not exhausted.
- The client decides to permanently activate one processor to improve overall performance for micro-partitions. One inactive processor still has On/Off licenses.
- IBM Tivoli License Manager monitors and reports licensing changes.
- The client installs WebSphere Application Server on partition DLPAR4.

Table 2-6 Example of licensing for an installed system

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (inactive)
Operating system	AIX 5L V5.3	Linux	AIX 5L V5.3	AIX 5L V5.3	Linux	AIX 5L V5.3	N/A
Additional System p5 software	HACMP		HACMP	HACMP			
Additional IBM software	DB2	DB2	DB2 / WebSphere Application Server	Domino / WebSphere Application Server	WebSphere Application Server		
Partition type	Dedicated	Dedicated	Micro	Micro	Micro	Micro	
Physical processors	4	3	8				1
Real use of physical processors	4+1 On/Off	3	8+1 On/Off				

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (inactive)
Maximum virtual processors used	N/A	N/A	2	3	5	10	
Capped / uncapped	N/A	N/A	Capped	Capped	Uncapped	Uncapped	
Maximum processing units	N/A	N/A	2.0	3.0	5.0 (VPs)	9.0 (Pool)	
Maximum real entitled capacity	N/A	N/A	1.7	2.2	4.2	3.4	
Used processors for non shared processor licenses	5	3	2	3	5	9	
Required AIX shared processor licenses = 14	5		Roundup(1.7+2.2)=4			9	
Client AIX processor licenses = 13+1	4		4			8	1 On/Off
Required HACMP processor licenses = 9	5		Roundup(1.7+2.2)=4				
Client HACMP processor licenses = 8 +1	4		4				1 On/Off
Reported shared processor licenses ITLM for DB2 = 10, Client DB2 licenses = 10	5	3	2				

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (inactive)
Reported shared processor licenses ITLM for WAS = 9, Client WAS licenses = 9+1			Roundup(1.7+2.2)=4		5		1 On/Off
Client Domino processor licenses = 15				15 (system)			

2.7.7 Linux operating system licensing

License terms and conditions of Linux operating system distributions are provided by the Linux distributor, but all base Linux operating systems are licensed under the GPL. Distributor pricing for Linux includes media, packaging/shipping, and documentation costs, and they may offer additional programs under other licenses as well as bundled service and support.

IBM offers the ability to accept orders and payment for Novell SUSE LINUX and Red Hat, Inc. Linux distributions for the System p5 systems. This includes shipping program media with initial System p5 system orders. Clients or authorized business partners are responsible for the installation of the Linux OS, with orders handled pursuant to license agreements between the client and the Linux distributor.

Clients should consider the quantity of virtual processors in micro-partitions for scalability and licensing purposes (uncapped partitions) when installing Linux in a virtualized System p5 system.

Each Linux distributor sets its own pricing method for their distribution, service, and support. Consult the distributor's Web site for information or visit:

http://www.novell.com/products/server/
https://www.redhat.com/software/rhel/compare/server/

2.8 Virtual and Shared Ethernet introduction

Virtual Ethernet enables inter-partition communication without the need for physical network adapters assigned to each partition. Virtual Ethernet allows the administrator to define in-memory connections between partitions handled at system level (POWER Hypervisor and operating systems interaction). These connections exhibit characteristics similar to physical high-bandwidth Ethernet connections and support the industry standard protocols (such as IPv4, IPv6, ICMP, or ARP). Shared Ethernet enables multiple partitions to share physical adapters for access to external networks.

Virtual Ethernet requires an IBM System p5 or IBM eServer p5 with either AIX 5L Version 5.3 or the appropriate level of Linux and a Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) to define the virtual Ethernet devices. Virtual Ethernet does not require the purchase of any additional features or software, such as the Advanced POWER Virtualization feature, which is needed for Shared Ethernet Adapters and Virtual I/O Servers.

The concepts of Virtual and Shared Ethernet on System p5 are introduced in the following sections:

- A general overview of Virtual LAN concepts and its use with AIX 5L is given.
- Inter-partition networking with virtual Ethernet on System p5 is introduced.
- Sharing of physical Ethernet adapters on System p5 to allow multiple partitions to access external networks is explained.

This completes the introduction of basic Virtual and Shared Ethernet concepts and is applied to an example.

2.8.1 Virtual LAN

This section discusses the general concepts of Virtual LAN (VLAN) technology. Specific reference to its implementation within AIX 5L is given after emphasizing the benefits of VLANs.

Virtual LAN overview

Virtual LAN is a technology used for establishing virtual network segments, also called network partitions, on top of physical switch devices. A Virtual LAN is a layer-2 (L2) concept, so it operates below TCP/IP. If configured appropriately, a single switch can support multiple VLANs, and a VLAN definition can also straddle multiple switches. VLANs on a switch can be disjunct or overlapping regarding the switch-ports assigned to them.

Typically, a VLAN is a single broadcast domain that enables all nodes in the VLAN to communicate with each other without any routing (L3 forwarding) or inter-VLAN bridging (L2 forwarding). For TCP/IP, this means that all node's interfaces in the same VLAN typically share the same IP subnet/netmask and can resolve all IP addresses on this VLAN to MAC addresses by using the Address Resolution Protocol (ARP). Even if a VLAN spans multiple switches, from the TCP/IP point-of-view, all nodes on the same VLAN can be reached with a single hop. This is in contrast to communication with nodes in other VLANs: their IP addresses cannot (and need not) be resolved by ARP, because these nodes are reached by making an additional hop through an L3 router (which UNIX administrators sometimes refer to as a gateway).

In Figure 2-14 on page 72, two VLANs (VLAN 1 and 2) are defined on three switches (Switch A, B, and C). There are seven hosts (A-1, A-2, B-1, B-2, B-3, C-1, and C-2) connected to the three switches. The physical network topology of the LAN forms a tree, which is typical for a non-redundant LAN:

- Switch A
  - Node A-1
  - Node A-2
  - Switch B
    - Node B-1
    - Node B-2
    - Node B-3
  - Switch C
    - Node C-1
    - Node C-2

In many situations, the physical network topology has to take into account the physical constraints of the environment, such as rooms, walls, floors, buildings, and campuses, to name a few. But VLANs can be independent of the physical topology:

- VLAN 1
  - Node A-1
  - Node B-1
  - Node B-2
  - Node C-1

- VLAN 2
  - Node A-2
  - Node B-3
  - Node C-2

Although nodes C-1 and C-2 are physically connected to the same switch C, traffic between two nodes can be blocked. To enable communication between VLAN 1 and 2, L3 routing or inter-VLAN bridging would have to be established between them; this would typically be provided by an L3 device, for example, a router or firewall plugged into switch A.

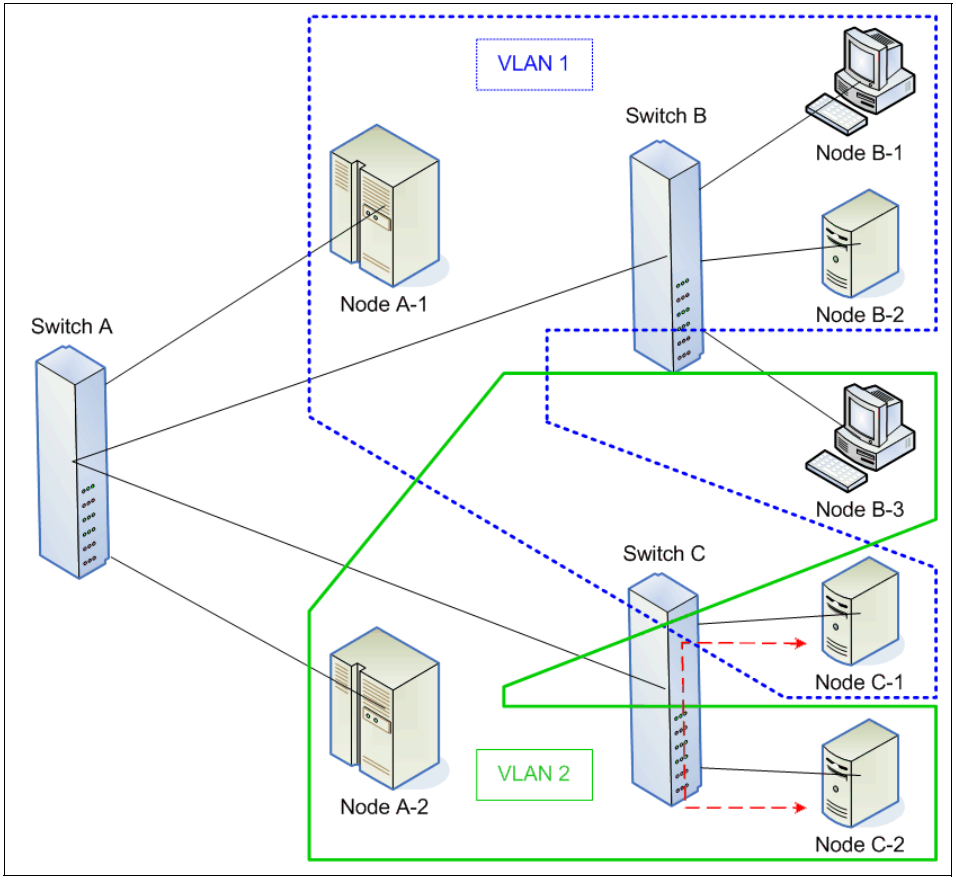


Figure 2-14 Example of a VLAN

Consider the uplinks between the switches: they carry traffic for both VLANs, 1 and 2. Thus, there has to be only one physical uplink from B to A, not one per VLAN. The switches will not be confused and will not mix-up the different VLANs' traffic, because packets travelling through the trunk ports over the uplink will have been tagged appropriately.

Virtual LAN benefits

The use of VLAN technology provides more flexible network deployment over traditional network technology. It can help overcome physical constraints of the environment and help reduce the number of required switches, ports, adapters, cabling, and uplinks. This simplification in physical deployment does not come for free: the configuration of switches and hosts becomes more complex when using VLANs. But the overall complexity is not increased; it is just shifted from physical to virtual.

VLANs also have the potential to improve network performance. By splitting up a network into different VLANs, you also split up broadcast domains. Thus, when a node sends a broadcast, only the nodes on the same VLAN will be interrupted by receiving the broadcast. The reason is that normally broadcasts are not forwarded by routers. You have to keep this in mind, if you implement VLANs and want to use protocols that rely on broadcasting, such as BOOTP or DHCP for IP auto-configuration.

It is also common practice to use VLANs if Gigabit Ethernet's Jumbo Frames are implemented in an environment, where not all nodes or switches are able to use or compatible with Jumbo Frames. Jumbo Frames allow for a MTU size of 9000 instead of Ethernet's default 1500. This may improve throughput and reduce processor load on the receiving node in a heavy loaded scenario, such as backing up files over the network.

VLANs can provide additional security by allowing an administrator to block packets from a domain to another domain on the same switch, therefore providing an additional control on what LAN traffic is visible to specific Ethernet ports on the switch. Packet filters and firewalls can be placed between VLANs, and Network Address Translation (NAT) could be implemented between VLANs. VLANs can make the system less vulnerable to attacks.

AIX 5L virtual LAN support

Some of the technologies for implementing VLANs include:

- Port-based VLAN
- Layer-2 VLAN
- Policy-based VLAN
- IEEE 802.1Q VLAN

VLAN support in AIX 5L is based on the IEEE 802.1Q VLAN implementation. AIX 5L can be used with port-based VLAN too, but this is completely transparent to AIX 5L. VLAN support is not special to Advanced POWER Virtualization on IBM System p5, but available on all IBM System p servers with the appropriate level of AIX 5L.

The IEEE 802.1Q VLAN support is achieved by letting the AIX 5L VLAN device driver add a VLAN ID tag to every Ethernet frame, as shown in Figure 2-15, and the Ethernet switches restricting the frames to ports that are authorized to receive frames with that VLAN ID.

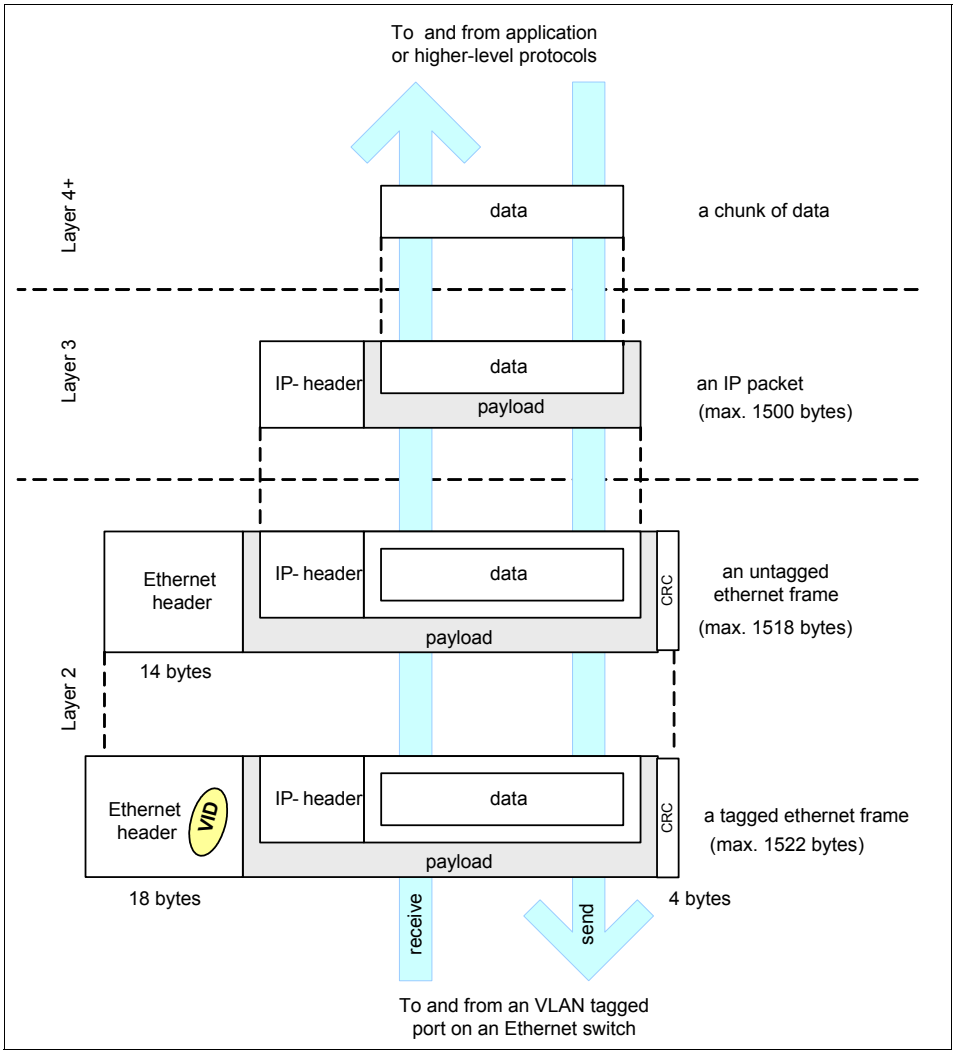


Figure 2-15 The VID is placed in the extended Ethernet header

The VLAN ID is placed in the Ethernet header and constitutes no additional header. To be able to do this, the Ethernet frame size for tagged frames was increased from 1518 bytes to 1522 bytes and the Ethernet header format was slightly modified with the introduction of IEEE802.1Q. Thus, in contrast to, for example, Point-to-Point-Protocol-over-Ethernet (PPPoE), which is commonly used for xDSL with a MTU of 1492, you do not have to care about reducing the TCP/IP's MTU of 1500 with respect to VLAN ID tagging.

Note: You do not have to reduce the TCP/IP default MTU size of 1500 for Ethernet due to the additional 4 bytes introduced by IEEE 802.1Q VLANs.

Attention: If you increase the TCP/IP's MTU size for virtual Ethernet that are implemented by the POWER Hypervisor, as introduced in 2.8.2, "Inter-partition networking with virtual Ethernet" on page 79, you must take the additional 4 bytes introduced by IEEE 802.1Q VLANs into account: the maximum MTU is 65394 without VLANs and 65390 bytes with VLANs. This is due to a limit of 65408 bytes for virtual Ethernet frames transmitted through the POWER Hypervisor. (The Ethernet headers are 14 and 18 bytes respectively, but there is no need for the 4 byte CRC in the POWER Hypervisor).

A port on a VLAN-capable switch has a default Port virtual LAN ID (PVID) that indicates the default VLAN the port belongs to. The switch adds the PVID tag to untagged packets that are received by that port. In addition to a PVID, a port may belong to additional VLANs and have those VLAN IDs assigned to it that indicate the additional VLANs the port belongs to.

- A switch port with a PVID only is called an *untagged port.*
  Untagged ports are used to connect *VLAN-unaware* hosts.
- A port with a PVID and additional VIDs is called a *tagged port*.
  Tagged ports are used to connect *VLAN-aware* hosts.

VLAN-aware means that the host is IEEE 802.1Q compatible and can interpret VLAN tags, and thus can interpret them, add them, and remove them from Ethernet frames. A VLAN-unaware host could be confused by receiving a tagged Ethernet frame. It would drop the frame and indicate a frame error.

Receiving packets on a tagged port

A tagged port uses the following rules when receiving a packet from a host:

1. Tagged port receives an untagged packet:

   The packet will be tagged with the PVID, then forwarded.

2. Tagged port receives a packet tagged with the PVID or one of the assigned VIDs:

   The packet will be forwarded without modification.

3. Tagged port receives a packet tagged with any VLAN ID other than the PVID or assigned additional VIDs:

   The packet will be discarded.

Thus, a tagged port will only accept untagged packets and packets with a VLAN ID (PVID or additional VIDs) tag of these VLANs that the port has been assigned to. The second case is the most typical.

Receiving packets on an untagged port

A switch port configured in the untagged mode is only allowed to have a PVID and will receive untagged packets or packets tagged with the PVID. The untagged port feature helps systems that do not understand VLAN tagging (VLAN unaware hosts) to communicate with other systems using standard Ethernet.

An untagged port uses the following rules when receiving a packet from a host:

1. Untagged port receives an untagged packet:

   The packet is tagged with the PVID, then forwarded.

2. Untagged port receives a packet tagged with the PVID:

   The packet is forwarded without modification.

3. Untagged port receives a packet tagged with any VLAN ID other than the PVID:

   The packet is discarded.

The first case is the most typical; the other two should not occur in a properly configured system.

After having successfully received a packet over a tagged or untagged port, the switch internally does not need to handle untagged packets any more, just tagged packets. This is the reason why multiple VLANs can easily share one physical uplink to a neighbor switch. The physical uplink is being made through trunk ports that have all the appropriate VLANs assigned.

Sending packets on a tagged or untagged port

Before sending a packet out, the destination ports of the packet must be determined by the switch based on the destination MAC address in the packet. The destination port must have a PVID or VID matching the VLAN ID of the packet. If the packet is a broadcast (or multicast), it is forwarded to all (or many) ports in the VLAN, even using uplinks to other switches. If no valid destination port can be determined, the packet is simply discarded. Finally, after internally forwarding the packet to the destination switch ports, before sending the packet out to the receiving host, the VLAN ID may be stripped-off or not, depending on the port-type:

- Tagged port sends out a packet:

  The PVID or VID remains tagged to the packet.

- Untagged port sends out a packet:

  The PVID is stripped from the packet.

Therefore, tagged and untagged switch ports behave similar with respect to receiving packets, but they behave different with respect to sending packets out.

Ethernet adapters and interfaces in AIX 5L

AIX 5L differentiates between a network adapter and network interface:

Network adapter Represents the layer-2 device, for example, the Ethernet adapter ent0 has a MAC address, such as 06:56:C0:00:20:03.

Network interface Represents the layer-3 device, for example the Ethernet interface en0 has an IP address, such as 9.3.5.195.

Typically, a network interface is attached to a network adapter, for example, an Ethernet interface en0 is attached to an Ethernet adapter ent0. There are also some network interfaces in AIX 5L that are not attached to a network adapter, for example, the loopback interface lo0 or a Virtual IP Address (VIPA) interface, such as vi0, if defined.

Note: Linux does not distinguish between a network adapter and a network interface with respect to device naming: there is just one device name for both. In Linux, a network device eth0 represents the network adapter and the network interface, and the device has attributes from layer-2 and layer-3, such as a MAC address and an IP address.

When using VLAN, EtherChannel (EC), Link Aggregation, (LA) or Network Interface Backup (NIB) with AIX 5L, the general concept is that Ethernet adapters are being associated with other Ethernet adapters, as shown in Figure 2-16 on

page 78. EtherChannel and Link Aggregation will be explained in more detail in 4.1.2, “Using Link Aggregation or EtherChannel to external networks” on page 187.

By configuring VLANs on a physical Ethernet adapter in AIX 5L, for each VLAN ID being configured by the administrator, another Ethernet adapter representing this VLAN will be created automatically. There are some slight difference with regard to what happens to the original adapters: with EC, LA, and NIB, the member adapters will not be available for any other use, for example, to be attached to an interface. Contrary to this, when creating a VLAN adapter, the attached Ethernet adapter will remain in the available state and an interface can still be attached to it in addition to the VLAN adapter.

If you have one real Ethernet adapter with device name ent0, which is connected to a tagged switch port with PVID=1 and VID=100, the administrator will generate an additional device name ent1 for the VLAN with VID=100. The original device name ent0 will represent the untagged Port VLAN with PVID=1. Ethernet interfaces can be put on both adapters: en0 would be stacked on ent0 and en1 on ent1, and different IP addresses will be configured to en0 and en1. This is shown in Figure 2-16.

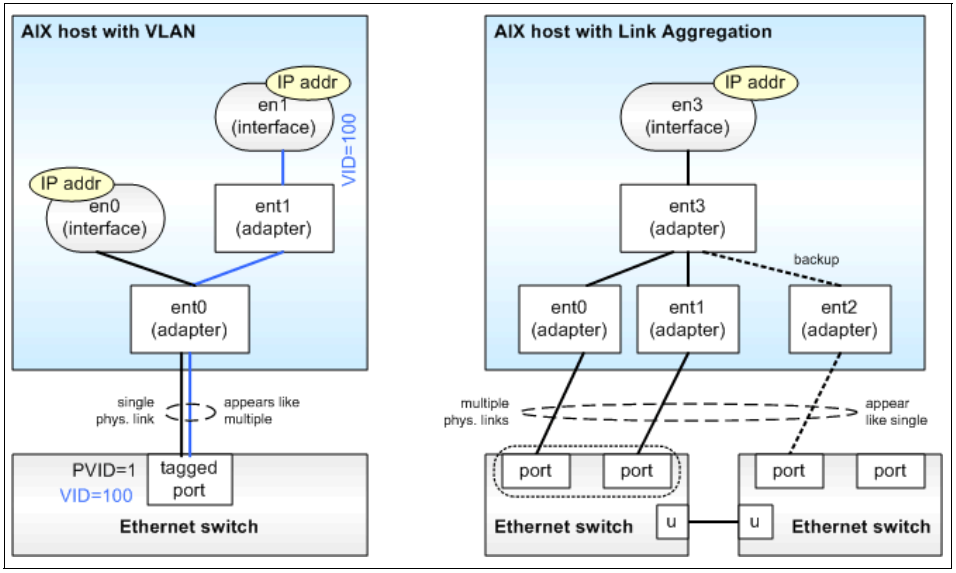


Figure 2-16 adapters and interfaces with VLANs (left) and LA (right)

2.8.2 Inter-partition networking with virtual Ethernet

The POWER Hypervisor firmware implements a IEEE 802.1Q VLAN style virtual Ethernet switch. Similar to a physical IEEE 802.1Q Ethernet switch it can support tagged and untagged ports. Because a virtual switch, does not really need ports, the virtual ports correspond directly to virtual Ethernet adapters that can be assigned to LPARs from the HMC or IVM. There is no need to explicitly attach a virtual Ethernet adapter to a virtual Ethernet switch port. But to draw on the analogy of physical Ethernet switches, a virtual Ethernet switch port is configured when you configure the virtual Ethernet adapter on the HMC or IVM.

For AIX 5L, a virtual Ethernet adapter is not much different from a real Ethernet adapter. It can be used:

- To configure an Ethernet interface with an IP address onto it
- To configure VLAN adapters (one per VID) onto it
- As a member of a Network Interface Backup adapter

But it cannot be used for EtherChannel or Link Aggregation

The POWER Hypervisor's virtual Ethernet switch can support virtual Ethernet frames of up to 65408 bytes size, which is much larger than what physical switches support: 1522 bytes is standard and 9000 bytes are supported with Gigabit Ethernet Jumbo Frames. Thus, with the POWER Hypervisor's virtual Ethernet, you can increase TCP/IP's MTU size to 65394 (= 65408 - 14 for the header, no CRC) in the non-VLAN-case and to 65390 (= 65408 - 14 - 4 for the VLAN, again no CRC) if you use VLAN. Increasing the MTU size is good for performance because it reduces processing due to headers and reduces the number of interrupts that the device driver has to react on.

2.8.3 Sharing physical Ethernet adapters

There are two approaches to connect a virtual Ethernet, that enables inter-partition communication on the same server, to an external network:

Routing	Layer-3 IP packet forwarding
Bridging	Layer-2 Ethernet frame forwarding

Routing

By enabling the IP forwarding capabilities of an AIX 5L or Linux partition with virtual and physical Ethernet adapters, the partition can act as router. Figure 2-17 on page 80 shows a sample configuration. The client partitions would have their default routes set to the partition, which routes the traffic to the external network.

Note: In this type of configuration, the partition that routes the traffic to the external network cannot be the Virtual I/O Server (VIOS), because you cannot enable IP forwarding from the VIOS command line interface.

The routing approach has the following characteristics:

- It does not require the purchase of the Advanced POWER Virtualization feature and use of a Virtual I/O Server.
- IP filtering, firewalling, or Quality of Service (QoS) could be implemented on these routing partitions.
- The routing partitions could also act as endpoints for IPsec tunnels, thus providing for encrypted communication over external networks for all partitions, without having to configure IPSec on all partitions.
- High availability can be addressed by implementing more than one such routing partition and by configuring IP multipathing on the clients, or by implementing IP address fail over on routing partitions, as discussed in 4.1.3, “High availability for communication with external networks” on page 189.

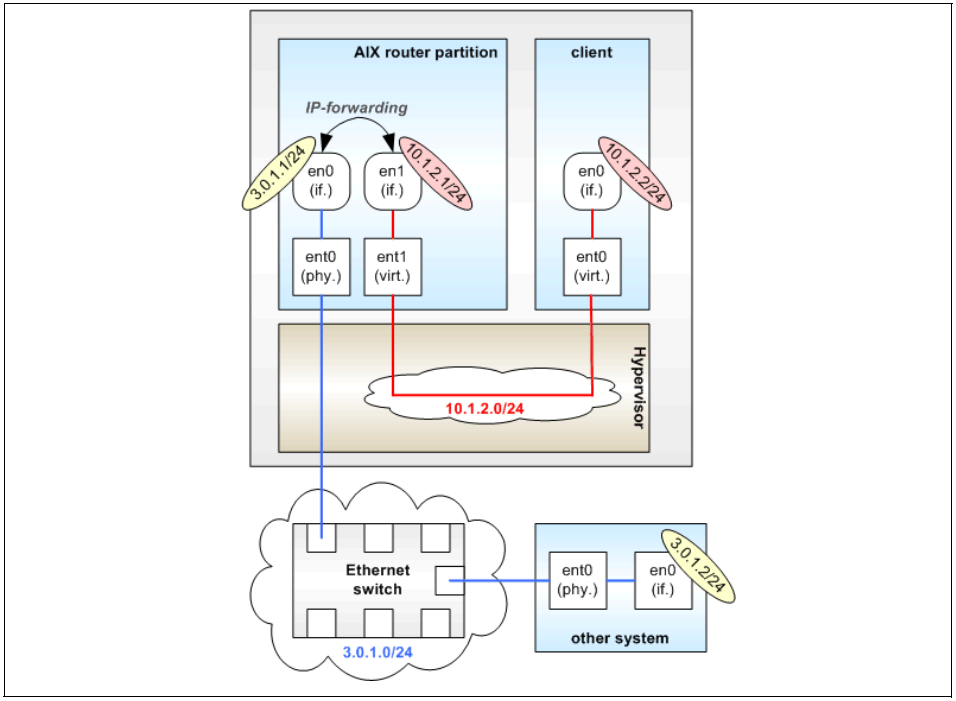


Figure 2-17 Connection to external network using routing

Shared Ethernet Adapter

A Shared Ethernet Adapter (SEA) can be used to connect a physical Ethernet network to a virtual Ethernet network. It also provides the ability for several client partitions to share one physical adapter. Using a SEA, you can connect internal and external VLANs using a physical adapter. The SEA hosted in the Virtual I/O Server (VIOS) acts as a layer-2 bridge between the internal and external network.

A SEA is a layer-2 network bridge to securely transport network traffic between virtual Ethernet networks and real network adapters. The Shared Ethernet Adapter service runs in the Virtual I/O Server. It cannot be run in a general purpose AIX 5L partition.

Tip: A Linux partition can provide bridging function too by use of the **`brctl`** command.

There are some things to consider on the use of SEA:

- SEA requires the POWER Hypervisor and Advanced POWER Virtualization feature and the installation of an Virtual I/O Server.
- SEA cannot be used prior to AIX 5L Version 5.3, because the device drivers for virtual Ethernet are only available for AIX 5L Version 5.3 and Linux. Thus, an AIX 5L Version 5.2 partition will need a physical Ethernet adapter.

The Shared Ethernet Adapter allows partitions to communicate outside the system without having to dedicate a physical I/O slot and a physical network adapter to a client partition. The Shared Ethernet Adapter has the following characteristics:

- Virtual Ethernet MAC addresses of virtual Ethernet adapters are visible to outside systems (using the **`arp -a`** command).
- Unicast, broadcast, and multicast is supported, so protocols that rely on broadcast or multicast, such as Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP), Boot Protocol (BOOTP), and Neighbor Discovery Protocol (NDP) can work across a SEA.

In order to bridge network traffic between the virtual Ethernet and external networks, the Virtual I/O Server has to be configured with at least one physical Ethernet adapter. One Shared Ethernet Adapter can be shared by multiple virtual Ethernet adapters and each can support multiple VLANs. Figure 2-18 on page 82 shows a configuration example of a SEA with one physical and two virtual Ethernet adapters. A Shared Ethernet Adapter can include up to 16 virtual Ethernet adapters that share the physical access.

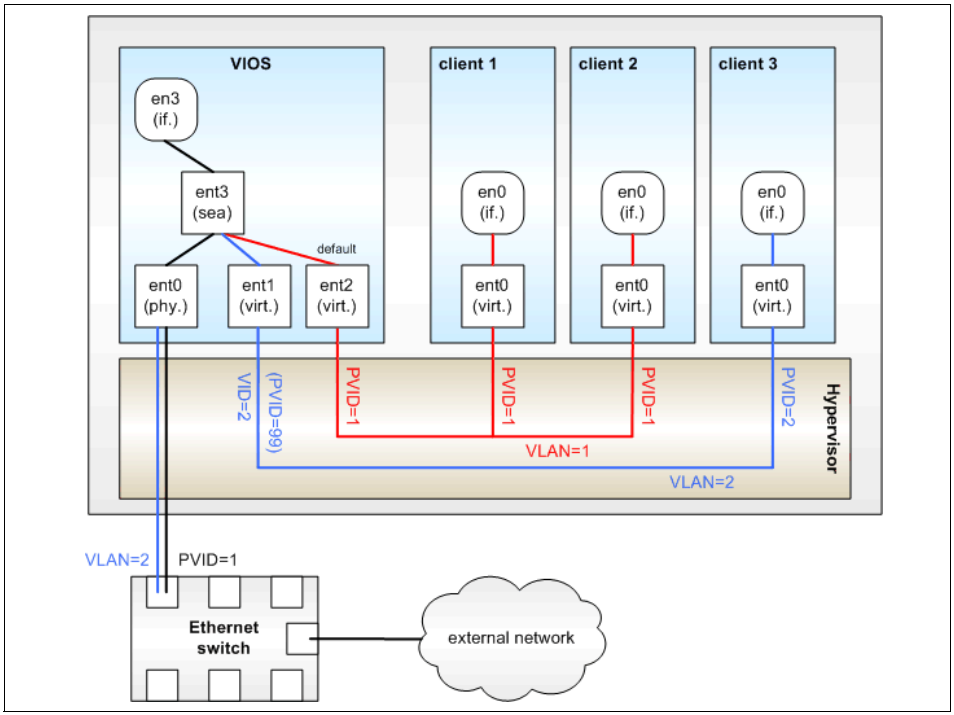


Figure 2-18 Shared Ethernet Adapter

A virtual Ethernet adapter connected to the Shared Ethernet Adapter must have the **Access External Networks** check box (named the *trunk flag* in some earlier releases of the HMC) enabled. Once an Ethernet frame is sent from a virtual Ethernet adapter on a client partition to the POWER Hypervisor, the POWER Hypervisor searches for the destination MAC address within the VLAN. If no such MAC address exists within the VLAN, it forwards the frame to the virtual Ethernet adapter on the VLAN that has the Access External Networks option enabled. This virtual Ethernet adapter corresponds to a port of a layer-2 bridge, while the physical Ethernet adapter constitutes another port of the same bridge.

Note: A Shared Virtual Adapter does not need to have IP configured to be able to perform the Ethernet bridging functionality. But it is very convenient to configure IP on the Virtual I/O Server, because then the Virtual I/O Server can be reached by TCP/IP, for example, to perform dynamic LPAR operations or to enable remote login. This can be done either by configuring an IP address directly on the SEA device, but it is sometimes more convenient to define an additional virtual Ethernet adapter into the Virtual I/O Server carrying the IP address and leave the SEA without the IP address, allowing for maintenance on the SEA without loosing IP connectivity in case SEA failover is configured. Neither has a remarkable impact on Ethernet performance.

The SEA directs packets based on the VLAN ID tags. One of the virtual adapters in the Shared Ethernet Adapter on the Virtual I/O Server must be designated as the *default* PVID adapter. Ethernet frames without any VLAN ID tags that the SEA receives from the external network are forwarded to this adapter and assigned the default PVID. In Figure 2-18 on page 82, it is ent2 that is designated as the default adapter, so all untagged frames received by ent0 from the external network will be forwarded to ent2. Since ent1 is not the default PVID adapter, only VID=2 will be used on this adapter, and the PVID=99 of ent1 is not important. It could be set to any unused VLAN ID. Alternatively, ent1 and ent2 could also be merged into a single virtual adapter ent1 with PVID=1 and VID=2, being flagged as the default adapter.

When the SEA receives or sends IP (IPv4 or IPv6) packets that are larger than the MTU of the adapter that the packet is forwarded through, either IP fragmentation is performed, or an `ICMP packet too big message` is returned to the sender, if the **Do not fragment** flag is set in the IP header. This is used, for example, with Path MTU discovery.

Theoretically, one adapter can act as the only contact with external networks for all client partitions. Depending on the number of client partitions and the network load they produce, performance can become a critical issue. Because the Shared Ethernet Adapter is dependent on Virtual I/O, it consumes processor time for all communications. A significant amount of CPU load can be generated by the use of virtual Ethernet and Shared Ethernet Adapter.

There are several different ways to configure physical and virtual Ethernet adapters into Shared Ethernet Adapters to maximize throughput:

- Using Link Aggregation (EtherChannel), several physical network adapters can be aggregated. Refer to 4.1.2, “Using Link Aggregation or EtherChannel to external networks” on page 187 for more details.
- Using several Shared Ethernet Adapters provides more queues and more performance.

Other aspects that have to be taken into consideration are availability (refer to 4.1.3, “High availability for communication with external networks” on page 189) and the ability to connect to different networks.

When to use routing or bridging

In a consolidation scenario, where multiple existing servers are being consolidated on a few systems, or if LPARs are often relocated from one system to another, bridging is often the preferred choice, because the network topology does not have to be changed and IP subnets and IP addresses of the consolidated servers can stay unmodified. Even an existing multiple VLAN scheme can be bridged.

Routing may be worth a consideration, if, in addition to basic packet forwarding, additional functions, such as IP filtering, firewalling, QoS Routing, or IPsec tunneling, should be performed in a central place. Also, if the external network is a layer-3-switched Ethernet with the dynamic routing protocol OSPF, as found in many IBM System z9 environments, routing may also be the preferred approach. For some environments, it may be a consideration, too, that the routing approach does not require the use of the Virtual I/O Server and the purchase of the Advanced POWER Virtualization feature.

To summarize, in most typical environments, bridging will be the most appropriate and even simpler to configure option, so it should be considered as the default approach.

2.8.4 Virtual and Shared Ethernet configuration example

After having introduced the basic concepts of VLANs, virtual Ethernet, and Shared Ethernet Adapters in the previous sections, this section discusses in more detail how communication between partitions and with external networks works, using the sample configuration in Figure 2-19 on page 85.

The configuration is using four client partitions (Partition 1 through Partition 4) running AIX 5L and one Virtual I/O Server (VIOS). Each of the client partitions is defined with one virtual Ethernet adapter. The Virtual I/O Server has a Shared Ethernet Adapter (SEA) that bridges traffic to the external network.

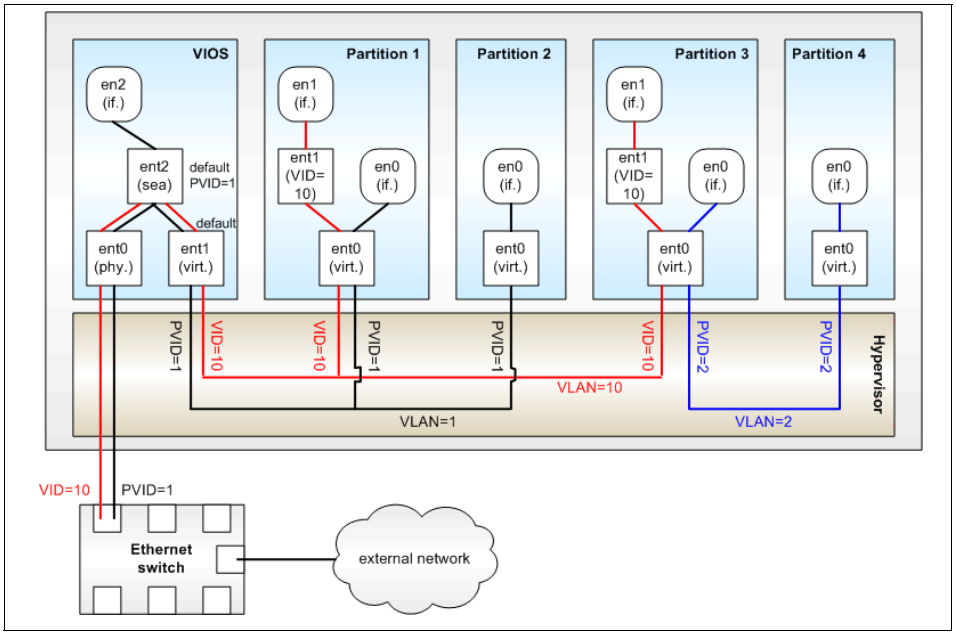


Figure 2-19 VLAN configuration example

Inter-partition networking

Partition 2 and Partition 4 are using the Port virtual LAN ID (PVID) only. This means that:

- The operating system running in such a partition is not aware of the VLANs.
- Only packets for the VLAN specified as PVID are received.
- Packets have their VLAN tagged removed by the POWER Hypervisor before the partitions receive them.
- Packets sent by these partitions have a VLAN tag attached for the VLAN specified as PVID by the POWER Hypervisor.

In addition to the PVID, the virtual Ethernet adapters in Partition 1 and Partition 3 are also configured for VLAN 10 using a VLAN Ethernet adapter (ent1) and network interface (en1) created through the **`smitty vlan`** command on AIX 5L (using the **`vconfig`** command on Linux). This means that:

- Packets sent through network interfaces en1 are tagged for VLAN 10 by the VLAN Ethernet adapter ent1 in AIX 5L.
- Only packets for VLAN 10 are received by the network interfaces en1.
- Packets sent through en0 are not tagged by AIX 5L, but are automatically tagged for the VLAN specified as PVID by the POWER Hypervisor.

- Only packets for the VLAN specified as PVID are received by the network interfaces en0.

In the configuration shown in Figure 2-19 on page 85, the Virtual I/O Server (VIOS) bridges both VLAN 1 and VLAN 10 through the Shared Ethernet Adapter (SEA) to the external Ethernet switch. But the VIOS itself can only communicate with VLAN 1 through its network interface en2 attached to the SEA. Because this is associated with the PVID, VLAN tags are automatically added and removed by the POWER Hypervisor when sending and receiving packets to other internal partitions through interface en2.

Table 2-7 summarizes which partitions in the virtual Ethernet configuration from Figure 2-19 on page 85 can communicate with each other internally through which network interfaces.

Table 2-7 Inter-partition VLAN communication

Internal VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / en0 VIOS / en2
2	Partition 3 / en0 Partition 4 / en0
10	Partition 1 / en1 Partition 3 / en1

If the VIOS should be able to communicate with VLAN 10 too, then it would need to have an additional Ethernet adapter and network interface with an IP address for VLAN 10, as shown on the left of Figure 2-20 on page 87. A VLAN-unaware virtual Ethernet adapter with a PVID only, as shown in the left of Figure 2-20 on page 87, would be sufficient; there is no need for a VLAN-aware Ethernet adapter (ent4), as shown in the center of Figure 2-20 on page 87. The simpler configuration with a PVID only would do the job, since the VIOS already has access to VLAN 1 through the network interface (en2) attached to the SEA (ent2). Alternatively, you could associate an additional VLAN Ethernet adapter (ent3) to the SEA (ent2), as shown on the right in Figure 2-20 on page 87.

Note: Although it is possible to configure multiple IP addresses on a VIOS, it is recommended to have no more than one, because some commands of the command line interface make this assumption. Thus, a Virtual I/O Server should have one IP address or no IP address.

An IP address is necessary on a Virtual I/O Server to allow communication with the HMC through RMC, which is a prerequisite to perform dynamic LPAR operations. Thus, we recommend having exactly one IP address on a Virtual I/O Server, if you want to be able to use dynamic LPAR with the Virtual I/O Server.

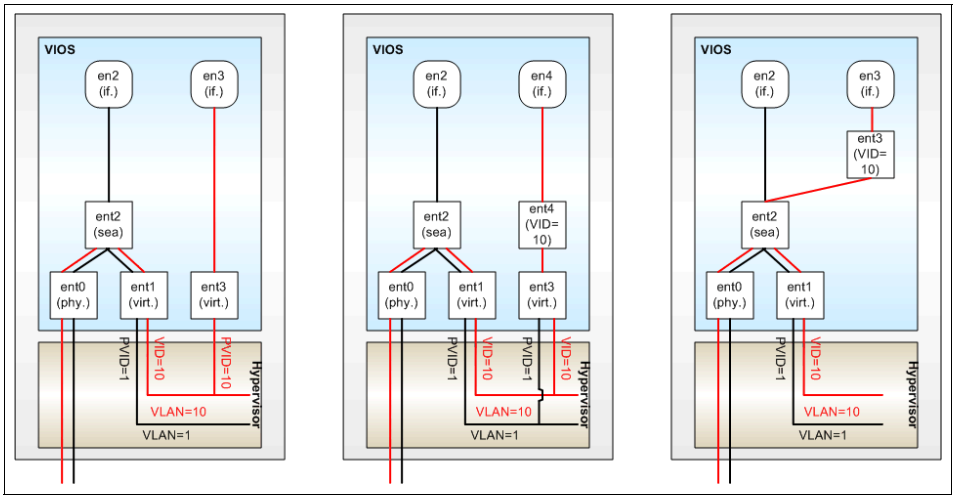


Figure 2-20 Adding virtual Ethernet adapters on the VIOS for VLANs

Communication with external networks

The Shared Ethernet Adapter (SEA) of Figure 2-19 on page 85 is configured with default PVID 1 and default adapter ent1. This means that untagged packets or packets with VID=1 that are received by the SEA from the external network are forwarded to adapter ent1. The virtual Ethernet adapter ent1 has the additional VID 10. Thus, packets tagged with VID 10 will be forwarded to ent1 as well.

The handling of outgoing traffic to the external network depends on the VLAN tag of the outgoing packets:

- Packets tagged with VLAN 1, which matches the PVID of the virtual Ethernet Adapter ent1, are untagged by the POWER Hypervisor before they are received by ent1, bridged to ent0 by the SEA, and sent out to the external network.

- Packets tagged with a VLAN other than the PVID 1 of the virtual Ethernet adapter ent1, such as VID 10, are processed with the VLAN tag unmodified.

In the virtual Ethernet and VLAN configuration example of Figure 2-19 on page 85, partition 1 and partition 2 have access to the external Ethernet through network interface ent0 using PVID 1.

- Since packets with VLAN 1 are using the PVID, the POWER Hypervisor will remove the VLAN tags before these packets are received by ent0 of partition 1 and 2.
- Since VLAN 1 is also the PVID of ent1 of the SEA in the Virtual I/O Server, these packets will be processed by the SEA without VLAN tags and will be send out untagged to the external network.
- Therefore, VLAN-unaware destination devices on the external network will be able to receive the packets as well.

Partition 1 and Partition 3 have access to the external Ethernet through network interface en1 and VLAN 10.

- These packets are sent out by the VLAN Ethernet adapter ent1, tagged with VLAN 10, through the physical Ethernet adapter ent0.
- The virtual Ethernet adapter ent1 of the SEA in the Virtual I/O Server also uses VID 10 and will receive the packet from the POWER Hypervisor with the VLAN tag unmodified. The packet will then be sent out through ent0 with the VLAN tag unmodified.
- Therefore, only VLAN-capable destination devices will be able to receive these.

Partition 4 has no access to the external Ethernet.

Table 2-8 summarizes which partitions in the virtual Ethernet configuration from Figure 2-19 on page 85 can communicate with which external VLANs through which network interface.

Table 2-8 VLAN communication to external network

External VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / en0 VIOS / en2
10	Partition 1 / en1 Partition 3 / en1

If this configuration must be extended to enable Partition 4 to communicate with devices on the external network, but without making Partition 4 VLAN-aware, the following alternatives could be considered:

- An additional physical Ethernet adapter could be added to partition 4.
- An additional virtual Ethernet adapter ent1 with PVID=1 could be added to Partition 4:

  Then Partition 4 would be able to communicate with devices on the external network using the default VLAN=1.
- An additional virtual Ethernet adapter ent1 with PVID=10 could be added to Partition 4:

  Then Partition 4 would be able to communicate with devices on the external network using VLAN=10.
- VLAN 2 could be added as additional VID to ent1 of the VIOS partition, thus bridging VLAN 2 to the external Ethernet, just like VLAN 10:

  Then Partition 4 would be able to communicate with devices on the external network using VLAN=2. This would work only if VLAN 2 is also known to the external Ethernet and there are some devices on the external network in VLAN 2.
- Partition 3 could act as a router between VLAN 2 and VLAN 10 by enabling IP forwarding on Partition 3 and adding a default route via Partition 3 to Partition 4.

2.8.5 Considerations

For considerations for virtual Ethernet and Shared Ethernet Adapters, refer to 4.1.7, “Considerations” on page 206 at the end of the discussion of advanced virtual Ethernet topics.

2.9 Virtual SCSI introduction

Virtual I/O pertains to a virtualized implementation of the SCSI protocol. Virtual SCSI requires POWER5 hardware with the Advanced POWER Virtualization feature activated. It provides virtual SCSI support for AIX 5L Version 5.3 and Linux (refer to 1.1.8, “Multiple operating system support” on page 4).

The driving forces behind virtual I/O are:

- The advanced technological capabilities of today’s hardware and operating systems, such as POWER5 and IBM AIX 5L Version 5.3.

- The value proposition enabling on demand computing and server consolidation. Virtual I/O also provides a more economic I/O model by using physical resources more efficiently through sharing.

At the time of writing, the virtualization features of the IBM System p platform support up to 254 partitions, while the server hardware provides up to 240 I/O slots and 192 internal SCSI disks per machine. With each partition typically requiring one I/O slot for disk attachment and another one for network attachment, this puts a constraint on the number of partitions. To overcome these physical requirements, I/O resources have to be shared. Virtual SCSI provides the means to do this for SCSI storage devices.

IBM supports up to ten Virtual I/O Servers within a single CEC managed by an HMC. Though architecturally up to 254 LPARS are supported, more than ten Virtual I/O Server LPARs within a single CEC have not been tested and therefore are not recommended.

Note: You will see different terms in this redbook that refer to the various components involved with virtual SCSI. Depending on the context, these terms may vary. With SCSI, usually the terms *initiator* and *target* are used, so you may see terms such as *virtual SCSI initiator* and *virtual SCSI target*. On the HMC, the terms *virtual SCSI server adapter* and *virtual SCSI client adapter* are used. Basically, they refer to the same thing. When describing the client/server relationship between the partitions involved in virtual SCSI, the terms *hosting partition* (meaning the Virtual I/O Server) and *hosted partition* (meaning the client partition) are used.

2.9.1 Partition access to virtual SCSI devices

The following sections describe the virtual SCSI architecture and the protocols used.

Virtual SCSI client and server architecture overview

Virtual SCSI is based on a client/server relationship. The Virtual I/O Server owns the physical resources and acts as server or, in SCSI terms, target device. The logical partitions access the virtual SCSI resources provided by the Virtual I/O Server as clients.

The virtual I/O adapters are configured using an HMC or through Integrated Virtualization Manager on smaller systems. The provisioning of virtual disk resources is provided by the Virtual I/O Server.

Often the Virtual I/O Server is also referred to as a hosting partition and the client partitions as hosted partitions.

Physical disks owned by the Virtual I/O Server can either be exported and assigned to a client partition whole, or can be partitioned into several logical volumes. The logical volumes can then be assigned to different partitions. Therefore, virtual SCSI enables sharing of adapters as well as disk devices.

To make a physical or a logical volume available to a client partition, it is assigned to a virtual SCSI server adapter in the Virtual I/O Server. The virtual SCSI adapter is represented by a vhost device as follows:

```
vhost0          Available  Virtual SCSI Server Adapter
```

The disk or logical volume being virtualized is represented by the standard AIX 5L device type of hdisk or logical volume type.

Note: A physical SCSI disk or LUN is assigned to a VSCSI adapter in the same procedure as a logical volume. A single VSCSI adapter can have multiple physical disks or LUNs assigned to it, creating multiple VSCSI target devices.

The client partition accesses its assigned disks through a virtual SCSI client adapter. The virtual SCSI client adapter sees standard SCSI devices and LUNs through this virtual adapter. The commands in the following example show how the disks appear on an AIX 5L client partition:

```
# lsdev -Cc disk -s vscsi
hdisk2 Available Virtual SCSI Disk Drive

# lscfg -vpl hdisk2
hdisk2 111.520.10DDEDC-V3-C5-T1-L810000000000 Virtual SCSI Disk Drive
```

The vhost SCSI adapter is the same as a normal SCSI adapter, because you can have multiple disks available from it. However, this vhost adapter can only be mapped to one virtual SCSI client adapter; any disks associated with that vhost adapter will only be visible to the client partition that has a VSCSI client adapter associated with the `vhost` adapter.

The mapping of VIOS virtual SCSI adapters to virtual SCSI client adapters is performed on the HMC. See 3.4, “Basic Virtual I/O Server scenario” on page 146 for more details.

Figure 2-21 on page 92 shows an example where one physical disk is partitioned into two logical volumes inside the Virtual I/O Server. Each of the two client partitions is assigned one logical volume, which it accesses through a virtual I/O adapter (VSCSI Client Adapter). Inside the partition, the disk is seen as a normal hdisk.

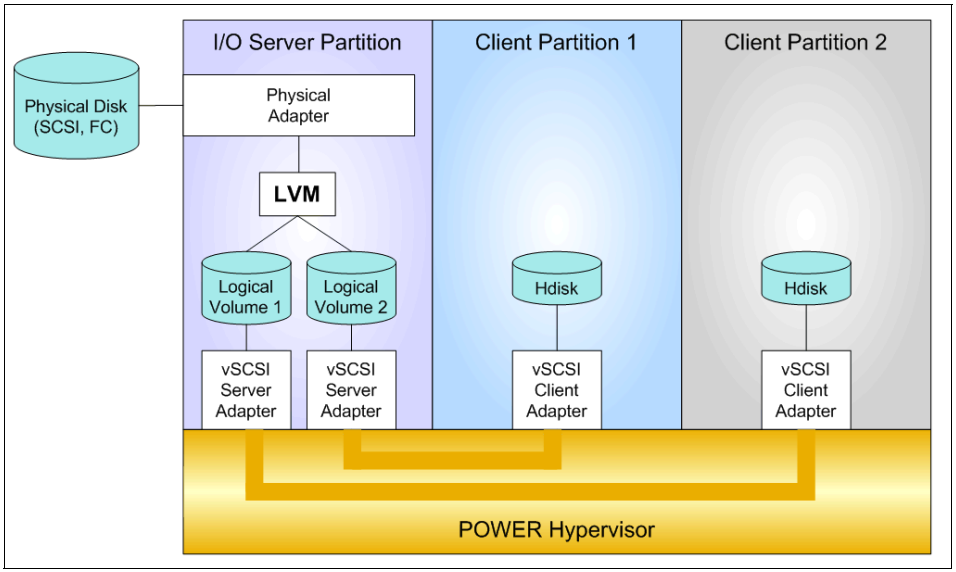


Figure 2-21 Virtual SCSI architecture overview

SCSI Remote Direct Memory Access

The SCSI family of standards provides many different transport protocols that define the rules for exchanging information between SCSI initiators and targets. Virtual SCSI uses the SCSI RDMA Protocol (SRP), which defines the rules for exchanging SCSI information in an environment where the SCSI initiators and targets have the ability to directly transfer information between their respective address spaces.

SCSI requests and responses are sent using the virtual SCSI adapters that communicate through the POWER Hypervisor.

The actual data transfer, however, is done directly between a data buffer in the client partition and the physical adapter in the Virtual I/O Server by using the Logical Remote Direct Memory Access (LRDMA) protocol.

Figure 2-22 on page 93 demonstrates data transfer using LRDMA. The VSCSI initiator of the client partition uses the Hypervisor to request data access from the VSCSI target device.

The Virtual I/O Server then determines which physical adapter this data is to be transferred from and sends its address to the Hypervisor. The Hypervisor maps this physical adapter address to the client partition's data buffer address to set up the data transfer directly from the physical adapter of the Virtual I/O Server to the client partition's data buffer.

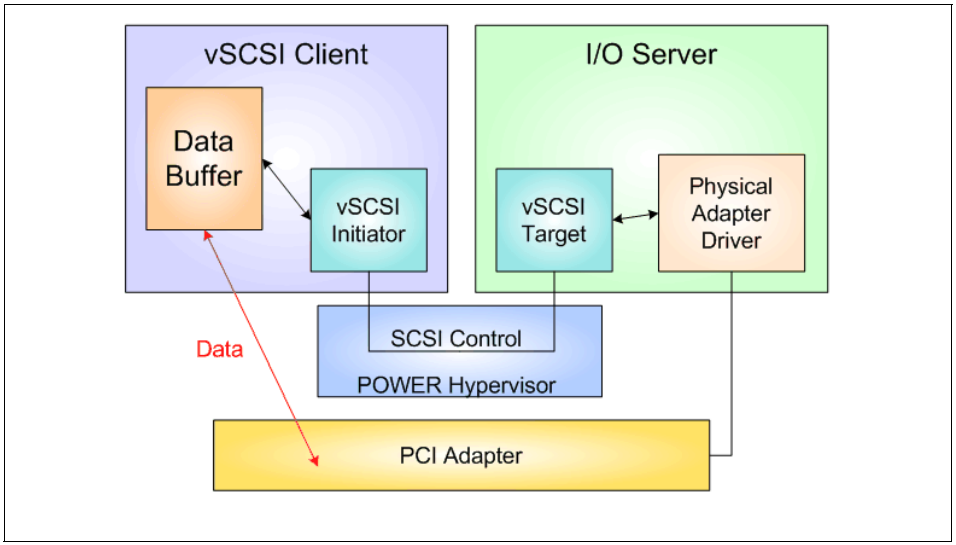


Figure 2-22 Logical Remote Direct Memory Access

AIX 5L device configuration for virtual SCSI

The virtual I/O adapters are connected to a virtual host bridge, which AIX 5L treats similar to a PCI host bridge. It is represented in the ODM as a bus device whose parent is sysplanar0. The virtual I/O adapters are represented as adapter devices with the virtual host bridge as their parent.

On the Virtual I/O Server, each logical volume or physical volume that is exported to a client partition is represented by a virtual target device that is a child of a virtual SCSI server adapter.

On the client partition, the exported disks are visible as normal hdisks, but they are defined in subclass vscsi. They have a virtual SCSI client adapter as a parent.

Figure 2-23 and Figure 2-24 show the relationship of the devices used by AIX 5L and Virtual I/O Server for virtual SCSI and their physical counterparts.

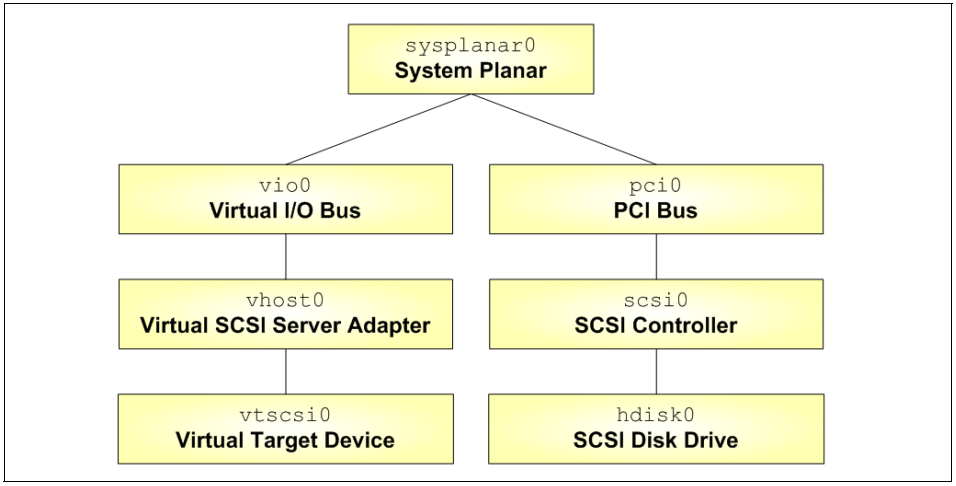


Figure 2-23 Virtual SCSI device relationship on Virtual I/O Server

Figure 2-23 shows the relationship between they physical disk and the target SCSI device on the VIOS. The SCSI controller is on the same hierarchal level as the virtual SCSI server adapter, which means that the VSCSI adapter can be considered the same as a SCSI adapter.

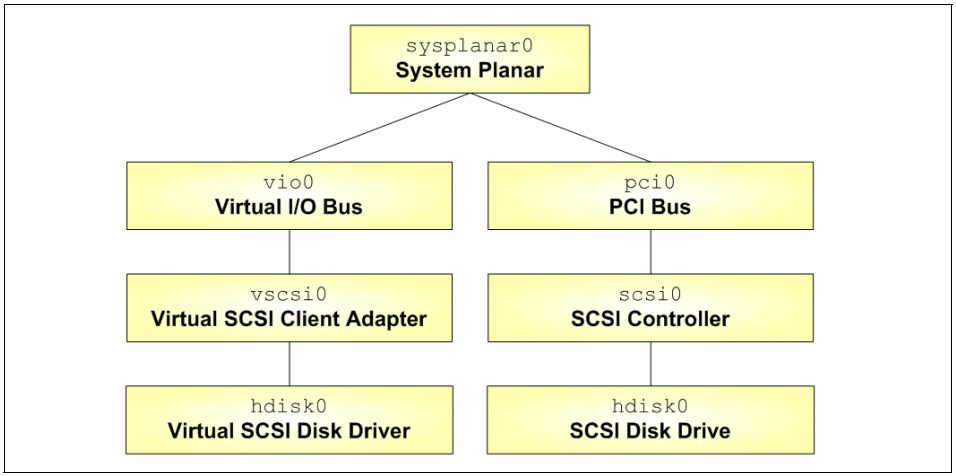


Figure 2-24 Virtual SCSI device relationship on AIX 5L client partition

Figure 2-24 on page 94 shows that the client partition's virtual SCSI client adapter is the same as a SCSI controller, each with a SCSI disk as a child device. The VSCSI client adapter can have multiple virtual SCSI disks as children, which is similar to a normal SCSI adapter. And as with physical SCSI adapters, there is a one to one relationship between adapter connections, which means only one virtual SCSI client adapter can connect to one virtual SCSI server adapter.

Dynamic partitioning for virtual SCSI devices

Virtual SCSI resources can be assigned and removed dynamically on the HMC. Virtual SCSI server and client adapters can be assigned and removed from a partition using dynamic logical partitioning after the resources have been set up in the Virtual I/O Server and varied off in AIX 5L.

With a physical device mapped to a virtual host adapter, moving the virtual SCSI disk between client partitions is similar to moving a physical adapter. Reassigning the VSCSI adapter on the server partition to a new client partition is followed by creating a new client partition VSCSI client adapter. On the client, the **cfgmgr** command is run and the new VSCSI disk is available.

Virtual SCSI optical devices

A DVD or CD device can be virtualized and assigned to Virtual I/O Clients. Only one virtual I/O client can have access to the drive at a time. The advantage of a virtual optical device is that you do not have to move the parent SCSI adapter between virtual I/O clients.

Note: The virtual drive cannot be moved to another Virtual I/O Server since client SCSI adapters cannot be created in a Virtual I/O Server. If you want the CD or DVD drive in another Virtual I/O Server, the virtual device must be unconfigured and the parent SCSI adapter must be unconfigured and moved, as described later in this section.

The following are the setup steps:

1. Let the DVD drive be assigned to a Virtual I/O Server.
2. Create a server SCSI adapter using the HMC where any partition can connect.

   Note: This should not be an adapter already used for disks since it will be removed or unconfigured when not holding the optical drive.

3. Run the **cfgdev** command to get the new vhost adapter. You can find the new adapter number with the **lsdev -virtual** command.

4. In the Virtual I/O Server, vios1, you create the virtual device with the following command:

   ```
   $ mkvdev -vdev <DVD drive> -vadapter vhostn -dev <name you want >
   ```

   where n is the number of the vhost adapter. See Example 2-1.

Example 2-1 Making the virtual device for the DVD drive

```
$ mkvdev -vdev cd0 -vadapter vhost3 -dev vcd
```

5. Create a client SCSI adapter in each LPAR using the HMC. The client adapter should point to the server adapter created in the previous step.

   Tip: It is useful to use the same slot number for all the clients.

6. In the client, run the **cfgmgr** command that will assign the drive to the LPAR. If the drive is already assigned to another LPAR you will get an error message and you will have to release the drive from the LPAR holding it.

Moving the drive:

If your documentation does not provide the vscsi adapter number, you can find it with the **lscfg|grep C***n* command, where *n* is the slot number of the virtual client adapter from the HMC.

1. Use the **rmdev -Rl vscsi***n* command to change the vscsi adapter and the optical drive to a defined state.

   Note: Adding the -d option also removes the adapter from the ODM.

2. The **cfgmgr** command in the target LPAR will make the drive available.

Note: You can also use the virtual CD or DVD to install an LPAR when selected in the SMS startup menu, provided the drive is not assigned to another LPAR.

Note: In IVM, the optical device is moved using the graphical user interface.

You can use the **dsh** command to find the LPAR currently holding the drive, as shown in Example 2-2 on page 97.

Note: Set the DSH_REMOTE_CMD=/usr/bin/ssh variable if you use SSH for authentication:

```
# export DSH_REMOTE_CMD=/usr/bin/ssh
# export DSH_LIST=<file listing lpars>
# dsh lsdev -Cc cdrom|dshbak
```

Example 2-2 Finding which LPAR is holding the optical drive using dsh

```
# dsh lsdev -Cc cdrom|dshbak
HOST: appserver
---------------
cd0 Available  Virtual SCSI Optical Served by VIO Server
HOST: dbserver
--------------
cd0 Defined  Virtual SCSI Optical Served by VIO Server
HOST: nim
---------
cd0 Defined  Virtual SCSI Optical Served by VIO Server
```

Tip: Put the DSH_LIST and DSH_REMOTE_CMD definitions in .profile on your admin server. You can change the file containing names of target LPARs without redefining DSH_LIST.

Note: If some partitions do not appear in the list, it is usually because the drive has never been assigned to the partition or completely removed with the -d option.

Or use the **ssh** command. See Example 2-3.

Example 2-3 Finding which LPAR is holding the optical drive using ssh.

```
# for i in nim dbserver appserver
> do
> echo $i; ssh $i lsdev -Cc cdrom
> done
nim
cd0 Defined  Virtual SCSI Optical Served by VIO Server
dbserver
cd0 Defined  Virtual SCSI Optical Served by VIO Server
appserver
cd0 Available  Virtual SCSI Optical Served by VIO Server
```

The following are the steps to unconfigure the virtual optical device when it is going to be used in the Virtual I/O Server for local backups:

1. Release the drive from the partition holding it.
2. Unconfigure the virtual device in the Virtual I/O Server.

> **Note:** If any media is in the drive, it will not unconfigure since it is then allocated.

3. When finished using the drive locally, use the **cfgdev** command in the Virtual I/O Server to restore the drive as a virtual drive

The following are the steps to unconfigure the virtual optical device in one Virtual I/O Server when it is going to be moved *physically* to another partition and to get it back.

1. Release the drive from the partition holding it.
2. Unconfigure the virtual device in the Virtual I/O Server.
3. Unconfigure the PCI adapter recursively.
4. Use the HMC to move the adapter to the target partition.
5. Run the **cfgmgr** command (or the **cfgdev** command for a Virtual I/O Server partition) to configure the drive.
6. When finished, remove the PCI adapter recursively
7. Use the HMC to move the adapter back.
8. Run the **cfgmgr** command (or the **cfgdev** command for a Virtual I/O Server partition) to configure the drive.

Use the **cfgdev** command on the Virtual I/O Server to reconfigure the drive when it is reassigned to the original partition in order to make it available as a virtual optical drive again. See Example 2-4 for unconfiguring and configuring the drive (disregard the error message from our test system).

Example 2-4 Unconfiguring and reconfiguring the DVD drive

```
$ rmdev -dev vcd -ucfg
vcd Defined
$ lsdev -slots
# Slot                     Description       Device(s)
U787B.001.DNW108F-P1-C1    Logical I/O Slot  pci3 ent0
U787B.001.DNW108F-P1-C3    Logical I/O Slot  pci4 fcs0
U787B.001.DNW108F-P1-C4    Logical I/O Slot  pci2 sisioa0
U787B.001.DNW108F-P1-T16   Logical I/O Slot  pci5 ide0
U9113.550.105E9DE-V1-C0    Virtual I/O Slot  vsa0
```

```
U9113.550.105E9DE-V1-C2   Virtual I/O Slot  ent1
U9113.550.105E9DE-V1-C3   Virtual I/O Slot  ent2
U9113.550.105E9DE-V1-C4   Virtual I/O Slot  vhost0
U9113.550.105E9DE-V1-C20  Virtual I/O Slot  vhost1
U9113.550.105E9DE-V1-C22  Virtual I/O Slot  vhost6
U9113.550.105E9DE-V1-C30  Virtual I/O Slot  vhost2
U9113.550.105E9DE-V1-C40  Virtual I/O Slot  vhost3
U9113.550.105E9DE-V1-C50  Virtual I/O Slot  vhost4
$ rmdev -dev pci5 -recursive -ucfg
cd0 Defined
ide0 Defined
pci5 Defined

$ cfgdev
Method error (/usr/lib/methods/cfg_vt_optical -l vcd ):
$ lsdev -virtual
name            status
description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vhost3          Available  Virtual SCSI Server Adapter
vhost4          Available  Virtual SCSI Server Adapter
vhost6          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
apps_rootvg     Available  Virtual Target Device - Disk
db_rootvg       Available  Virtual Target Device - Disk
linux_lvm       Available  Virtual Target Device - Disk
nim_rootvg      Available  Virtual Target Device - Disk
vcd             Available  Virtual Target Device - Optical Media
vtscsi0         Available  Virtual Target Device - Logical Volume
ent3            Available  Shared Ethernet Adapter
```

2.9.2 General virtual SCSI considerations

The following areas should be considered when implementing virtual SCSI:

- At the time of writing, virtual SCSI supports Fibre Channel, parallel SCSI, SCSI RAID devices, and optical devices, including DVD-RAM and DVD-ROM. Other protocols, such as SSA and tape devices, are not supported.

- A logical volume on the Virtual I/O Server used as a virtual SCSI disk cannot exceed 1 TB in size.
- The SCSI protocol defines mandatory and optional commands. While virtual SCSI supports all the mandatory commands, not all optional commands are supported.

Important: Although logical volumes that span multiple physical volumes are supported, for optimum performance, a logical volume should reside wholly on a single physical volume. To guarantee this, volume groups can be composed of single physical volumes.

Keeping an exported storage pool backing device or logical volume on a single hdisk results in optimized performance.

Bad Block Relocation on Virtual I/O Server Version 1.3 is supported as long as a virtual SCSI device is not striped, mirrored, and when the LV spans multiple PV.

To verify that a logical volume does not span multiple disks, run the following:

```
$ lslv -pv app_vg
app_vg:N/A
PV                COPIES        IN BAND       DISTRIBUTION
hdisk5            320:000:000   99%           000:319:001:000:000
```

Only one disk should appear in the resulting list.

Installation and migration considerations

The following are the major installation and migration considerations:

- Planning client partition root volume group sizes prior to creating logical volumes is recommended. Increasing a rootvg by extending its associated Virtual I/O Server logical volume is not supported. For information about extending volume groups, see “Increasing a client partition volume group” on page 311.
- Migrating from a physical SCSI disk to a virtual SCSI device is not supported at this time. All virtual SCSI devices created on the Virtual I/O Server are treated as new devices. A backup and restore of the data would be required when migrating from a physical device to a virtual device.
- The Virtual I/O Server uses several methods to uniquely tag a disk for use in as a virtual SCSI disk. They are:
  - Unique device identifier (UDID)
  - IEEE volume identifier

– Physical Volume Identifier (PVID)

Regardless of which method is used, the virtual device will always appear the same to the VSCSI client. The method used does have a bearing on the layout of the physical storage that is managed by the Virtual I/O Server.

The preferred disk identification method for virtual disks is UDID. MPIO devices use the UDID method. Ideally, sometime in the future, all devices will have converged such that all devices may use the UDID method.

Today, most non-MPIO disk storage multi-pathing software products use the PVID method instead of the UDID method. When all devices have converged to use the UDID method, existing devices created using the old methods (such as PVID or IEEE volume ID) will continue to be supported as is. Clients should be aware that in order to take advantage of certain future actions or function performed in the Virtual I/O Server LPAR, older devices may first require data migration, that is, some type of backup and restore of the attached disks. These actions may include, but are not limited to the following:

– Conversion from a Non-MPIO environment to MPIO.

– Conversion from the PVID to the UDID method of disk identification.

– Removal and rediscovery of the Disk Storage ODM entries.

– Updating non-MPIO multi-pathing software under certain circumstances.

– Possible future enhancements to VIO.

- Virtual SCSI itself does not have any maximums in terms of number of supported devices or adapters. At the time of writing, the Virtual I/O Server supports a maximum of 1024 virtual I/O slots per server. A maximum of 256 virtual I/O slots can be assigned to a single partition.

  Every I/O slot needs some resources to be instantiated. Therefore, the size of the Virtual I/O Server puts a limit on the number of virtual adapters that can be configured.

Performance considerations

There are performance implications when using virtual SCSI devices. It is important to understand that, with the use of SCSI Remote DMA and POWER Hypervisor calls, virtual SCSI may use additional CPU cycles when processing I/O requests. When putting heavy I/O load on virtual SCSI devices, this means more CPU cycles on the Virtual I/O Server will be used.

Provided that there is sufficient CPU processing capacity available, the performance of virtual SCSI should be comparable to dedicated I/O devices.

Suitable applications for virtual SCSI include boot disks for the operating system or Web servers, which will typically cache a lot of data. When designing a virtual

I/O configuration, performance is an important aspect that should be given careful consideration.

Virtual SCSI runs at low priority interrupt levels, while virtual Ethernet runs on high priority interrupts due to the latency differences between disks and LAN adapters. A client that produces extensive network activity has the potential to impact performance of a client that requires extensive disk activity as the virtual Ethernet has higher priority interrupts. Either a larger Virtual I/O Server with more CPU resources will be required or a second Virtual I/O Server can be considered to split the two high throughput clients.

2.10 Partition Load Manager introduction

The Partition Load Manager (PLM) software is part of the Advanced POWER Virtualization feature and helps clients maximize the utilization of processor and memory resources of dynamic LPAR-capable logical partitions running AIX 5L.

The Partition Load Manager is a resource manager that assigns and moves resources based on defined policies and utilization of the resources. PLM manages memory, both dedicated processors, and partitions using Micro-Partitioning technology to readjust the resources. This adds additional flexibility on top of the micro-partitions flexibility added by the POWER Hypervisor.

PLM, however, has no knowledge about the importance of a workload running in the partitions and cannot readjust priority based on the changes of types of workloads. PLM does not manage Linux and i5/OS partitions.

Partition Load Manager is set up in a partition or on another system running AIX 5L Version 5.2 ML4 or AIX 5L Version 5.3. Linux or i5OS support for PLM and the clients is not available. You can have other installed applications on the partition or system running the Partition Load Manager as well. A single instance of the Partition Load Manager can only manage a single server.

Partition Load Manager uses a client/server model to report and manage resource utilization. The clients (managed partitions) notify the PLM server when resources are either under- or over-utilized. Upon notification of one of these events, the PLM server makes resource allocation decisions based on a policy file defined by the administrator.

Figure 2-25 on page 103 shows an overview of the components of Partition Load Manager. In this figure, the PLM server would be notified by the database partition that it requires additional processor resources. The Web server partition would also have notified the PLM server that it has an abundance of processor

resources. Using the Policy File, the PLM server would determine that taking processor resources from the Web server partition and assigning them to the database partition is acceptable and then proceed to carry out this reallocation of resources using the HMC.

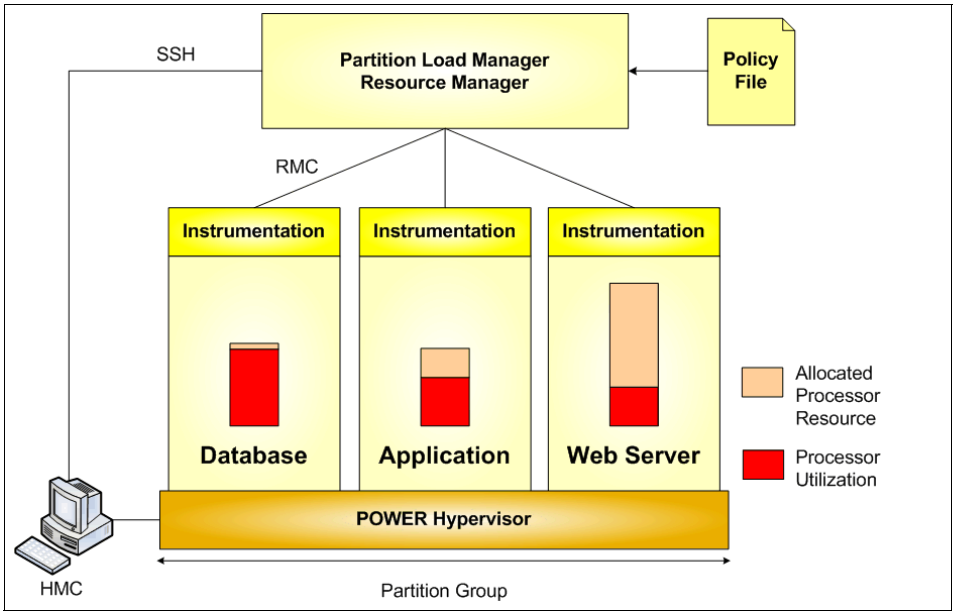


Figure 2-25 Partition Load Manager overview

When the Database partition returns to a normal workload, the PLM server would determine if resource reallocation is required to return resources back to the Web server partition.

2.11 Integrated Virtualization Manager

This section provides a short introduction about the Integrated Virtualization Manager. For further information and detailed configuration steps refer to *Integrated Virtualization Manager on IBM System p5*, REDP-4061.

The primary hardware management solution IBM has developed relies on an appliance server called Hardware Management Console (HMC), packaged as an external tower or rack-mounted personal computer.

The HMC is a centralized point of hardware control. A single HMC can handle multiple POWER5 systems, and two HMCs may manage the same set of servers in a dual-active configuration.

Hardware management is done by an HMC using a standard Ethernet connection to the service processor of each POWER5 system. Interacting with the service processor, the HMC is capable to create, manage, and modify logical partitions, modify the hardware configuration of the managed system, and manage service calls.

For a smaller or distributed environment, not all functions of an HMC are required, and the deployment of an additional personal computer may not be suitable.

IBM has developed the IVM, a hardware management solution that inherits most of the HMC features, but it is limited to managing a single server, avoiding the need for an dedicated personal computer. It provides a solution that enables the administrator to reduce system setup time. The IVM is integrated within the Virtual I/O Server product, which enables I/O and processor virtualization in IBM System p systems.

2.11.1 IVM setup guidelines

Since one of the goals of IVM is management, some implicit rules apply to the server configuration and setup. To manage a system using the IVM, the following guidelines are designed to assist you:

- The system is configured in the *Factory Default* mode, which means that a single partition with service authority predefined owns all the hardware resources. If the system is not configured in Factory Default mode because it is already partitioned or attached to an HMC, you can reset the system to the Factory Default mode using the ASMI.
- The predefined partition is started automatically at system power on. The physical control panel and the serial ports are attached to this partition.
- The APV feature has to be enabled. When ordering the feature with the system, it should be enabled by default; otherwise, it can be enabled using the ASMI.
- Virtual I/O Server Version 1.2 or higher has to be installed on the predefined partition.

The Virtual I/O Server then automatically allocates all I/O resources. All other LPARs are configured using the built-in IVM on the Virtual I/O Server. The client partitions have no physical I/O devices configured. They access disks, network, and optical devices only through the Virtual I/O Server as virtual devices. The configuration can be done by a GUI or by using the command line interface on the Virtual I/O Server. The administrator can use a browser to connect to IVM to set up the system configuration.

Figure 2-26 shows a sample configuration using IVM. The Virtual I/O Server owns all physical adapters, while the other two partitions are configured to use only virtual devices.

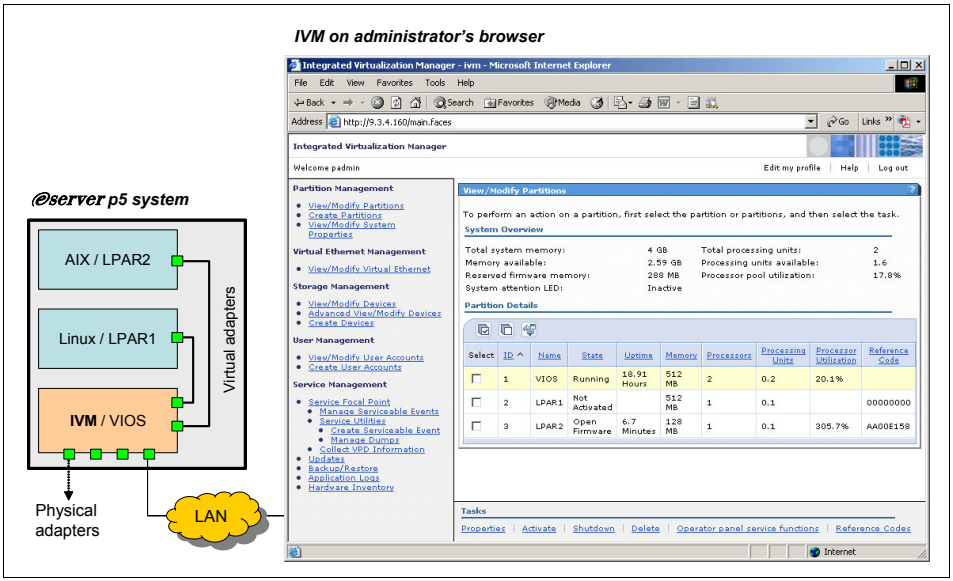


Figure 2-26 Integrated Virtualization Manager configuration

The tight relationship between the Virtual I/O Server and IVM enables the administrator to manage a partitioned system without the HMC. The software that is normally running on the HMC has been reworked to fit inside the Virtual I/O Server, reducing its functions to those required by the IVM configuration model. Since IVM is running using system resources, the design has been developed to have minimal impact on their consumption.

IVM does not interact with the system's service processor. A specific device named the *Virtual Management Channel* (VMC) has been developed on Virtual I/O Server to enable a direct Hypervisor configuration without requiring additional network connections to be set up. This device is activated by default when Virtual I/O Server is installed as the first partition.

VMC allows IVM to provide basic logical partitioning functions:

- Logical partitioning configuration, including dynamic LPAR
- Boot, start, and stop actions for individual partitions
- Displaying partition status
- Manage virtual Ethernet
- Manage virtual storage
- Provide basic system management

Since IVM is executing in an LPAR, it has limited service functions and ASMI must be used. For example, system power on must be done by physically pushing the system power on button or remotely accessing ASMI, since IVM is not executing while the system is powered off. ASMI and IVM together provide a basic but effective solution for a single partitioned server.

LPAR management with IVM is made through a Web interface developed to make administration tasks easier and quicker compared to a full HMC solution. Being integrated within the Virtual I/O Server code, IVM also handles all virtualization tasks that normally require Virtual I/O Server commands to be run.

IVM manages the system in a different way than the HMC. A new POWER5 administrator will quickly learn the required skills, while an HMC expert should study the differences before using IVM.

2.11.2 Partition configuration with IVM

LPAR configuration is made by assigning processors, memory, and virtual I/O using a Web GUI wizard. In each step of the process, simple questions are asked of the administrator, and the range of possible answers are provided. Most of the parameters related to LPAR setup are hidden during creation time to ease the setup and can be changed after the creation in the partition properties if needed.

Resources that are assigned to an LPAR are immediately allocated and are no longer available to other partitions, regardless of the fact that the LPAR is activated or powered down. This behavior makes management more direct and different than an HMC managed system, where resource over commitment is allowed.

LPARs in a IVM managed system are isolated exactly as in all POWER5 systems and cannot interact except using the virtual devices. Only IVM has been enabled to perform limited actions on the other LPARs, such as:

- Power on and power off
- Shut down gracefully the operating system

- Create and delete
- View and change configuration

Starting with Virtual I/O Server Version 1.3, dynamic LPAR is now supported with IVM.

Processors

An LPAR can be defined either with dedicated or with shared processors.

When shared processors are selected for a partition, the wizard lets the administrator choose only the number of virtual processors to be activated. For each virtual processor, 0.1 processing units are implicitly assigned and the LPAR is created in uncapped mode, with a weight of 128.

Processing unit value, uncapped mode, and the weight can be changed, modifying the LPAR configuration after it has been created.

Virtual Ethernet

The IVM managed system is configured with four predefined virtual Ethernet networks, each having a virtual Ethernet ID ranging from 1 to 4. Every LPAR can have up to two virtual Ethernet adapters that can be connected to any of the four virtual networks in the system.

Each virtual Ethernet network can be bridged by Virtual I/O Server to a physical network using only one physical adapter. The same physical adapter cannot bridge more than one virtual network.

The virtual Ethernet network is a bootable device and can be used to install the LPAR's operating system.

Virtual storage

Every LPAR is equipped with one or more virtual SCSI disks using a single virtual SCSI adapter. The virtual disks are bootable devices and treated by the operating system as normal SCSI disks.

Virtual optical device

Any optical device equipped on the Virtual I/O Server partition (either CD-ROM, DVD-ROM, or DVD-RAM) can be virtualized and assigned at any logical partition, one at a time, using the same virtual SCSI adapter provided to virtual disks. Virtual optical devices can be used to install the operating system and, if DVD-RAM, to make backups.

Virtual TTY

In order to allow LPAR installation and management, IVM provides a virtual terminal environment for LPAR console handling. When a new LPAR is defined, it is automatically assigned a client virtual serial adapter to be used as the default console device. With IVM, a matching server virtual terminal adapter is created and linked to the LPAR's client virtual client.

2.12 Dynamic LPAR operations

There are a few things you need to know when doing dynamic LPAR operations that pertain to both virtualized and non-virtualized server resources:

- Make sure resources such as a physical and virtual adapters being added and moved between partitions are not being used by other partitions. This means doing the appropriate cleanup on the client side by deleting them out of the system or taking them offline by executing PCI hot-plug procedures through SMIT if they are physical adapters.
- You will not be able to dynamically add additional memory to a running partition up to the maximum setting that you had defined in the profile.
- The HMC should be able to communicate to the logical partitions over the network for RMC connections.
- Be aware of performance implications when removing memory from logical partitions.
- Running applications must be dynamic LPAR-aware when doing dynamic resource allocation and deallocation so it is able to resize itself and accommodate changes in hardware resources.

5.1, "Dynamic LPAR operations" on page 258 shows you how to do dynamic LPAR operations on a running system.

2.13 Linux virtual I/O concepts

In addition to AIX 5L, Linux can also be used on IBM System p5. Linux can be installed in a dedicated or shared processor partition. Linux running in a partition can use physical devices and virtual devices. It can also participate in virtual Ethernet and can access external networks through Shared Ethernet Adapters (SEA). A Linux partition can use virtual SCSI disks. Linux can also provide some of the virtualization services to other Linux partitions that the IBM Virtual I/O Server typically provides to AIX 5L and Linux partitions.

The following terms and definitions are general:

Virtual I/O client Any partition that is using virtualized devices provided by other partitions.

Virtual I/O Server Any server of I/O partition (VIOS) that is providing virtualized devices to be used by other partitions.

Specifically, there are two different types of Virtual I/O Servers available for System p5:

APV VIOS The Advanced POWER Virtualization Virtual I/O Server from IBM for pSeries p5. This is a special-function appliance that can only be used as an Virtual I/O Server, but is not intended to run general applications.

Linux VIOS The implementation of Virtual I/O Server functionality on Linux.

VIO client and VIO server are *roles*. By this definition, a system could be a VIO client and a VIO server at the same time. In most cases, this should be avoided, because it complicates administration through complex dependencies.

Important: Throughout the rest of this redbook, except for this section, *Virtual I/O Server* or *VIOS*, when written without further qualifications, means the APV VIOS. In this section, we explicitly call it the APV VIOS to distinguish it from the Linux VIOS.

A partition of a System p5 can host four different types of systems:

- AIX 5L: Can only be a VIO client.
- Linux: Can be a VIO client and VIO server.
- APV VIOS: A VIO server only.
- i5/OS on select systems.

We will now explain the basic concepts of Linux VIO clients and servers. For a more detailed explanation and hands-on guide, you can refer to the publications listed in 2.13.5, “Further reading” on page 115.

2.13.1 Linux device drivers for IBM System p5 virtual devices

IBM worked with Linux developers to create device drivers for the Linux 2.6 kernel that enable Linux to use the IBM System p5 virtualization features.

Table 2-9 shows all the kernel modules for IBM System p5 virtual devices.

Table 2-9 Kernel modules for IBM System p5 virtual devices

Linux 2.6 kernel module	Supported virtual device	Source file locations, relative to /usr/src/linux/drivers/
hvcs	virtual console server	char/hvc*
ibmveth	virtual Ethernet	net/ibmveth*
ibmvscsic	virtual SCSI - client/initiator	scsi/ibmvscsi*
ibmvscsis	virtual SCSI - server/target	scsi/ibmvscsi*

The Linux 2.6 kernel source can be downloaded from:

`ftp://ftp.kernel.org/pub/linux/kernel/v2.6/`

Precompiled Linux kernel modules are included with some Linux distributions.

2.13.2 Linux as a VIO client

Linux running in a partition of a System p5 can use virtual Ethernet adapters and use virtual devices provided by Virtual I/O Servers. A Linux VIO client can use both APV VIOS and Linux VIOS at the same time.

Virtual console

The System p5 provides a virtual console /dev/hvc0 to each Linux partition.

Virtual Ethernet

To use virtual Ethernet adapters with Linux, the Linux kernel module ibmveth must be loaded. If IEEE 802.1Q VLANs are used, then, in addition, the Linux kernel module 8021q must be available. Virtual Ethernet adapters use the same naming scheme like physical Ethernet adapters, such as eth0 for the first adapter. VLANs are configured by the **`vconfig`** command.

Linux can use inter-partition networking with other partitions and share access to external networks with other Linux and AIX 5L partitions, for example, through a Shared Ethernet Adapter (SEA) of an APV Virtual I/O Server.

Virtual SCSI Client

The IBM virtual SCSI client for Linux is implemented by the ibmvscsic Linux kernel module. When this kernel module is loaded, it will scan and auto-discover

any virtual SCSI disks provided by the Virtual I/O Servers. Discovery can also be triggered manually after additional virtual SCSI disks have been added to the virtual SCSI adapter at the Virtual I/O Server.

Virtual SCSI disks will be named just as regular SCSI disks, for example, /dev/sda for the first SCSI disk or /dev/sdb3 for the third partition on the second SCSI disk.

MPIO

Linux has support for generic and some vendor-specific implementations of Multi-Path I/O (MPIO), and some vendors provide additional MPIO-capable device drivers for Linux.

Attention: Currently, a Linux VIO client does not support MPIO to access the same disks using two Virtual I/O Servers.

Remember that MPIO can also be implemented in the Virtual I/O Server to provide redundant access to external disks for the VIO client. But implementing MPIO in the Virtual I/O Server instead of the VIO client does not provide the same degree of high availability to the VIO client, because the VIO client has to be shutdown when the single Virtual I/O Server is brought down, for example, when the Virtual I/O Server is upgraded. The difference between MPIO in the VIO client and in the Virtual I/O Server is shown in Figure 2-27.

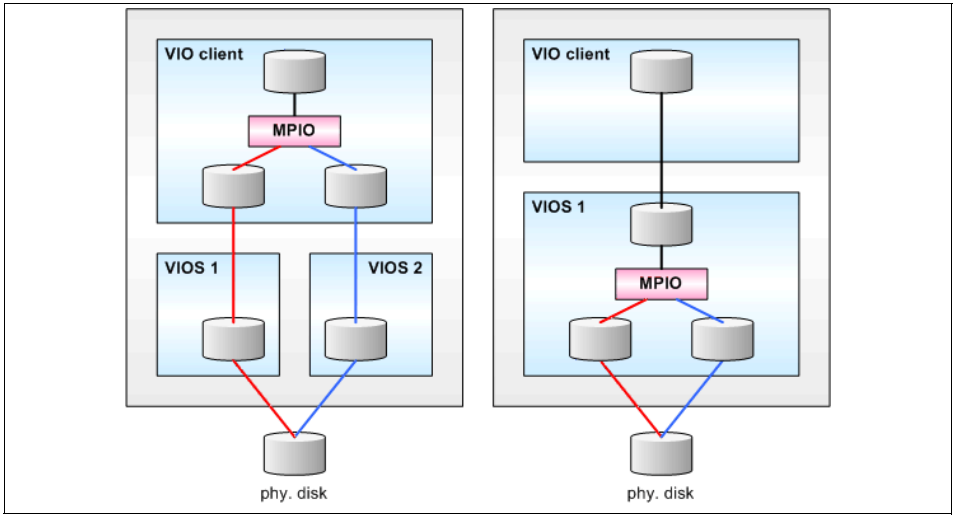


Figure 2-27 Implementing MPIO in the VIO client or VIO server

Mirroring

Linux can mirror disks by use of the RAID-Tools. Thus, for redundancy, you may mirror each virtual disk provided by one Virtual I/O Server to another virtual disk provided by a different Virtual I/O Server.

Attention: Be aware that using Linux's RAID-Tools to mirror the boot- and root-partitions, and enabling the Linux system to boot from mirrored disks, may require modification of the default boot scheme.

Remember that mirroring could also be implemented in the Virtual I/O Server to provide redundant access to external disks for the VIO client. But implementing mirroring in the Virtual I/O Server instead of the VIO client would not provide the same degree of high availability to the VIO client, because the VIO client would have to be shut down when the single Virtual I/O Server were brought down, for example, when the Virtual I/O Server is upgraded. The difference between mirroring in the VIO client and in the Virtual I/O Server is shown in Figure 2-28.

Note: Currently LVM mirroring of virtual disks on the APV Virtual I/O Server is not recommended.

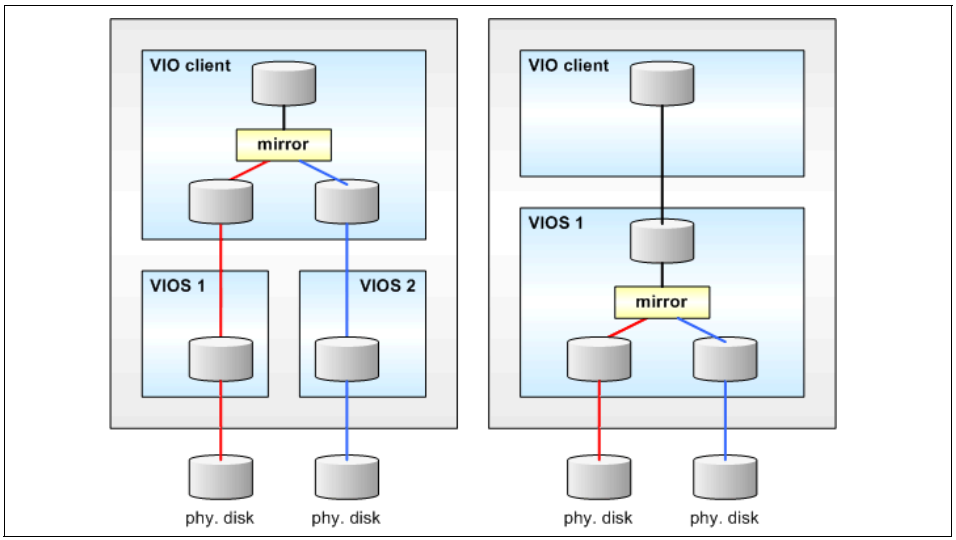


Figure 2-28 Implementing Mirroring in the VIO client or VIO server

LVM

The Linux OS Logical Volume Manager can use any mix of virtual and physical disks.

Attention: Be aware that using Linux's LVM for the root-partition, and enabling the Linux system to boot with the root file system on a logical volume, may require modification of the default boot scheme.

2.13.3 Linux as a VIO server

Linux can also provide some of the virtualization services to other Linux partitions that the IBM System p5 APV VIOS typically provides to AIX 5L and Linux partitions.

Consideration: The Linux VIO Server is not supported by AIX 5L. For VIO clients running AIX 5L, only the APV VIOS is supported.

Ethernet Bridging

To provide layer-2 bridging functionality on Linux, for example, between virtual and physical Ethernet adapters, bridging functionality must have been enabled at kernel build-time. The bridge-utils RPM must be installed to make the **`brctl`** command available, which is used to set up and configure a bridge. The **`ipfilt`** command can be used to restrict access.

Bridging between a physical and a virtual Ethernet adapter with Linux is shown in Figure 2-29 on page 114. Note that the IP-address for the bridging Linux partition is defined on br0, not on eth0, which is now member of the bridge.

Routing

Linux can act as a router to external networks, too. IP-forwarding has to be enabled. The **`ipfilt`** command can be used to restrict access.

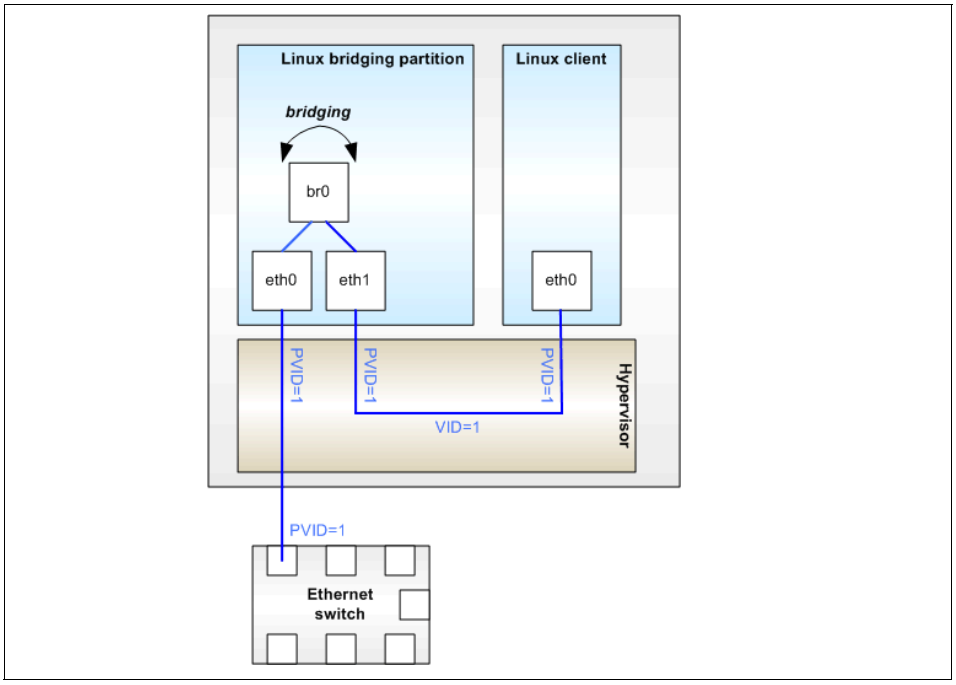


Figure 2-29 Bridging a virtual and a physical Ethernet adapter with Linux

Virtual SCSI server

The virtual SCSI server is implemented by the Linux kernel module ibmvscsis. First, the virtual SCSI server adapters have to be configured. Then the following can be added to be used as virtual disks for Linux VIO clients:

- Physical disks, such as /dev/sdc,
- Partitions of physical disks, such as /dev/sdd2,
- Logical volumes, such as /dev/datavg/lv01,
- Loopback-mounted files

The use of loopback-mounted files has the advantage that cloning and backing-up of virtual disks can be easily performed with the **cp** command. The virtual disks should not be active when this is done.

2.13.4 Considerations

The use of Linux as a VIO client or server is subject to the following considerations:

- Supported VIO clients are AIX 5L V5.3 and specific Linux distributions.
- For AIX 5L V5.3 VIO clients, only the APV VIOS is supported by IBM.
- Linux VIO clients may use the APV VIOS and the Linux VIO Server.
- Use of the APV VIOS may require the purchase of an additional feature, depending on the IBM System p5 model.
- The Linux VIO Server is currently only available with SUSE Linux Enterprise Server 9 or 10 (SLES V9 and V10).
- MPIO on Linux clients using virtual disks is currently not supported.
- Dynamic LPAR on Linux partitions uses the Hot-Plugging subsystem and behaves differently from AIX 5L. Some operations require a reboot of the Linux system, for example, resizing of memory.

2.13.5 Further reading

The following are sources of additional information, that can help you to configure and use Advanced POWER Virtualization features with Linux VIO clients and servers:

- *Linux for pSeries installation and administration (SLES 9)*, by Chris Walden <cmwalden@us.ibm.com>, published at IBM DeveloperWorks, found at:

  http://www-128.ibm.com/developerworks/linux/library/l-pow-pinstall/

- *Linux virtualization on POWER5: A hands-on setup guide*, by John Engel <engel@us.ibm.com>, published at IBM DeveloperWorks, found at:

  http://www-128.ibm.com/developerworks/edu/l-dw-linux-pow-virutal.html

- *POWER5 Virtualization: How to set up the SUSE Linux Virtual I/O Server*, by Nigel Griffiths, <nag@uk.ibm.com>, found at:

  http://www-128.ibm.com/developerworks/eserver/library/es-susevio/

3

Setting up the Virtual I/O Server: the basics

This chapter introduces the basics of configuring a virtual environment on an IBM System p5. The Virtual I/O Server build is covered in its entirety and a basic scenario of configuring a Virtual I/O Server along with client partitions is also demonstrated.

The basic topics include:

- Getting started
- Creating a Virtual I/O Server partition
- Virtual I/O Server software installation
- Basic Virtual I/O Server scenario
- Interaction with UNIX client partitions

3.1 Getting started

This section provides the following information about the operating environment of the Virtual I/O Server:

- The Command line interface of the VIOS, also named the IOSCLI
- Hardware resources managed
- Software packaging and support

3.1.1 Command line interface

The Virtual I/O Server provides a restricted scriptable command line interface (IOSCLI). All VIOS configurations should be made on this IOSCLI using the restricted shell provided.

Important: No configuration of volume group and logical volume creation should be done under the oem_setup_env shell environment.

The following Virtual I/O Server administration is made through the command line interface:

- Device management (physical, virtual, and LVM)
- Network configuration
- Software installation and update
- Security
- User management
- Installation of OEM software
- Maintenance tasks

For the initial login to the Virtual I/O Server, use the padmin user ID, which is the main administrator. Upon login, a password change is required. There is no default password to remember.

Upon logging into the Virtual I/O Server, you will be placed into a restricted Korn shell. The restricted Korn shell works the same way as a regular Korn shell with some restrictions. Specifically, users cannot do the following:

- Change the current working directory.
- Set the value of the SHELL, ENV, or PATH variable.
- Specify the path name of the command that contains a redirect output of a command with a >, >|, <>, or >.

As a result of these restrictions, you are not able to run commands that are not accessible to your PATH variable. These restrictions prevent you from directly sending the output of the command to a file, requiring you to pipe the output to the **tee** command instead.

After you are logged on, you can enter the **help** command to get an overview of the supported commands, as in Example 3-1.

Example 3-1 Supported commands on Virtual I/O Server Version 1.3

```
$ help
Install Commands
 ioslevel
 license
 lssw
 oem_platform_level
 oem_setup_env
 remote_management
 updateios

LAN Commands
 cfglnagg
 cfgnamesrv
 entstat
 hostmap
 hostname
 lsnetsvc
 lstcpip
 mktcpip
 chtcpip
 netstat
 optimizenet
 ping
 rmtcpip
 startnetsvc
 stopnetsvc
 traceroute
 vasistat

Device Commands
 chdev
 chpath
 cfgdev
 lsdev
 lsmap

Security Commands
 lsfailedlogin
 lsgcl
 viosecure

UserID Commands
 chuser
 lsuser
 mkuser
 passwd
 rmuser

Maintenance Commands
 backupios
 bootlist
 cattracerpt
 chdate
 chlang
 diagmenu
 errlog
 fsck
 invscout
 ldfware
 loginmsg
 lsfware
 lslparinfo
 motd
 mount
 pdump
 restorevgstruct
 savevgstruct
 showmount
 shutdown
 snap
```

```
 lspath
 mkpath
 mkvdev
 mkvt
 rmdev
 rmpath
 rmvdev
 rmvt

Physical Volume Commands
 lspv
 migratepv

Logical Volume Commands
 chlv
 cplv
 extendlv
 lslv
 mklv
 mklvcopy
 rmlv
 rmlvcopy

Volume Group Commands
 activatevg
 chvg
 deactivatevg
 exportvg
 extendvg
 importvg
 lsvg
 mirrorios
 mkvg
 redefvg
 reducevg
 syncvg
 unmirrorios

Storage Pool Commands
 chsp
 lssp
 mkbdsp
 mksp
 rmbdsp
 startsysdump
 starttrace
 stoptrace
 sysstat
 topas
 unmount
 viostat
 wkldmgr
 wkldagent
 wkldout

Shell Commands
 awk
 cat
 chmod
 clear
 cp
 crontab
 date
 ftp
 grep
 head
 ls
 man
 mkdir
 more
 mv
 rm
 sed
 stty
 tail
 tee
 vi
 wall
 wc
 who
```

To receive further help on these commands, use the **help** command, as shown in Example 3-2.

Example 3-2 Help command

```
$ help errlog
Usage: errlog [[ -ls][-seq Sequence_number] | -rm Days]]

       Displays or clears the error log.

       -ls         Displays information about errors in the error log file
                   in a detailed format.

       -seq        Displays information about a specific error in the error log file
                   by the sequence number.

       -rm         Deletes all entries from the error log older than the
                   number of days specified by the Days parameter.
```

The Virtual I/O Server command line interface supports two execution modes:

- Traditional mode
- Interactive mode

The traditional mode is for single command execution. In this mode, you run one command at a time from the shell prompt. For example, to list all virtual devices, enter the following:

```
#ioscli lsdev -virtual
```

To reduce the amount of typing required in traditional shell level mode, an alias has been created for each sub-command. With the aliases set, you are not required to type the **ioscli** command. For example, to list all devices of type adapter, you can enter the following:

```
#lsdev -type adapter
```

In interactive mode, the user is presented with the **ioscli** command prompt by executing the **ioscli** command without any sub-commands or arguments. From this point on, **ioscli** commands are run one after the other without having to retype **ioscli**. For example, to enter interactive mode, enter:

```
#ioscli
```

Once in interactive mode, to list all virtual devices, enter:

```
#lsdev -virtual
```

External commands, such as **grep** or **sed**, cannot be run from the interactive mode command prompt. You must first exit interactive mode by entering **quit** or **exit**.

3.1.2 Hardware resources managed

The Advanced POWER Virtualization feature that enables Micro-Partitioning on a server provides the Virtual I/O Server installation media. A logical partition with enough resources to share to other partitions is also required. The following minimum hardware requirements must be available to create the Virtual I/O Server:

POWER5 server The Virtual I/O capable machine.

Hardware Management Console

The HMC is needed to create the partition and assign resources or an installed IVM either on the predefined partition or pre-installed.

Storage adapter The server partition needs at least one storage adapter.

Physical disk If you want to share your disk to client partitions, you need a disk large enough to make *sufficient-sized* logical volumes on it.

Ethernet adapter This adapter is needed if you want to allow securely routed network traffic from a virtual Ethernet to a real network adapter.

Memory At least 512 MB of memory is needed. Similar to an operating system, the complexity of the I/O subsystem and the number of the virtual devices have a bearing on the amount of memory required, such as when using multiple paths to SAN storage devices.

Virtual I/O Server Version 1.3 is designed for selected configurations that include specific models of IBM and other vendor storage products.

Consult your IBM representative or Business Partner for the latest information and included configurations.

Virtual devices exported to client partitions by the Virtual I/O Server must be attached through one of the following physical adapters:

- PCI 4-Channel Ultra3 SCSI RAID Adapter (FC 2498)
- PCI-X Dual Channel Ultra320 SCSI RAID Adapter (FC 5703)
- Dual Channel SCSI RAID Enablement Card (FC 5709)
- PCI-X Dual Channel Ultra320 SCSI Adapter (FC 5712)
- 2 Gigabit Fibre Channel PCI-X Adapter (FC 5716)
- 2 Gigabit Fibre Channel Adapter for 64-bit PCI Bus (FC 6228)
- 2 Gigabit Fibre Channel PCI-X Adapter (FC 6239)

Careful planning is recommended before you begin the configuration and installation of your Virtual I/O Server and client partitions. Depending on the type of workload and needs of an application, consider mixing virtual and physical devices. For example, if your application benefits from fast disk access, then plan a physical adapter dedicated to this partition.

3.1.3 Software packaging and support

Installation of the Virtual I/O Server partition is performed from a special **mksysb** DVD-ROM that is provided to clients that order the Advanced POWER Virtualization feature, at an additional charge. For the p5-590 and p5-595, this feature is already included. The Virtual I/O Server software is only supported in Virtual I/O Server partitions.

The Virtual I/O Server Version 1.3 DVD-ROM installation media can be installed in the following ways:

- Media (assigning the DVD-ROM drive to the partition and booting from the media).
- The HMC (inserting the media in the DVD-ROM drive on the HMC and using the **installios** command).
- Using the DVD-ROM media together with the NIM server and executing the **smitty installios** command (the secure shell needs to be working between NIM and HMC).

Important:

- For IVM, there is no need to assign the DVD-ROM for the partition because VIOS installs on a predefined partition.
- The **installios** command is not applicable for IVM managed systems.

For more information about the installation of the Virtual I/O Server, refer to 3.3, “Virtual I/O Server software installation” on page 142.

3.2 Creating a Virtual I/O Server partition

This section provides the steps to create the Virtual I/O Server logical partition and to install the VIOS software.

3.2.1 Defining the Virtual I/O Server partition

This section shows you how to create the logical partition and install the Virtual I/O Server named VIO_Server1 from the HMC.

Experience has shown that a shared, uncapped processor is adequate in most cases to use for the Virtual I/O Server partition.

Tip: If you plan on using SAN disks and you do not have a NIM server, you can use one internal disk or disk pack for a NIM server partition for your Virtual I/O Server and also client partitions. All systems are shipped with at least one internal disk. The NIM server could use a virtual Ethernet adapter for communication since virtual networks do not require a Virtual I/O Server between partitions within a single server.

The following section describes how to allocate a shared processor to our Virtual I/O Server partition. Follow the steps below to create the Virtual I/O Server:

1. Figure 3-1 shows the HMC with four attached managed system. For our basic VIOS configuration setup, we will use the managed system named P550_ITSO.

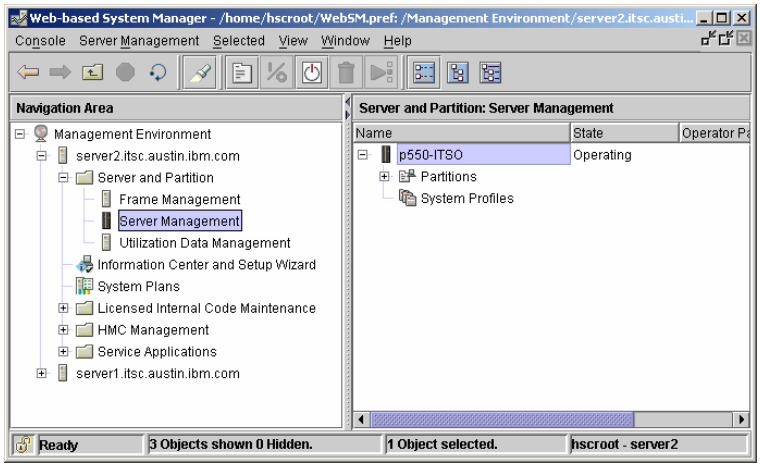


Figure 3-1 Hardware Management Console view

The following windows will take us through creating our first Virtual I/O Server partition.

2. Right-click the managed system **P520_ITSO**, then select **Create** → **Logical Partition**, as shown in Figure 3-2, to start the Create Logical Partition Wizard.

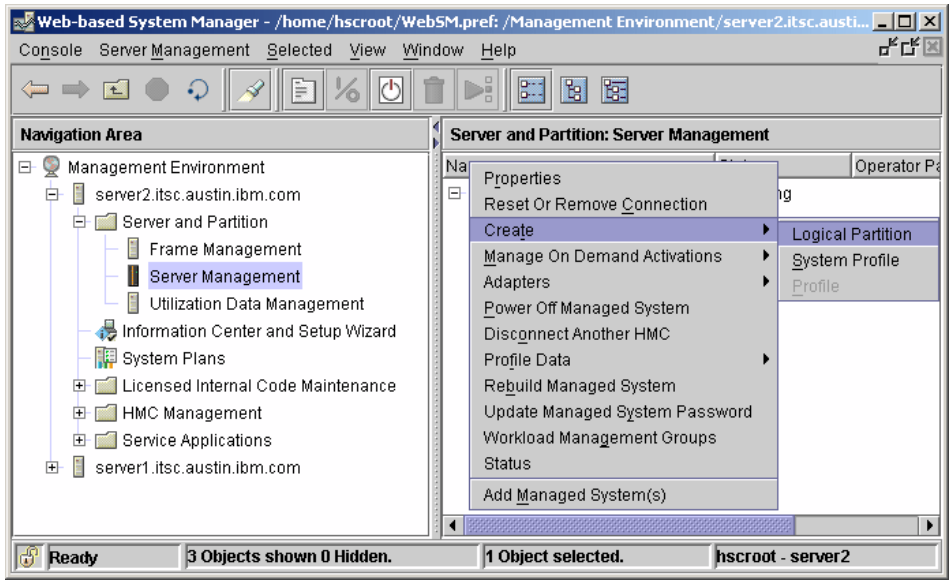


Figure 3-2 Starting the Create Logical Partition Wizard

3. Enter the partition name and ID (or keep the ID selected by the system. IDs must be unique), then select the **Virtual I/O Server** button, as shown in Figure 3-3.

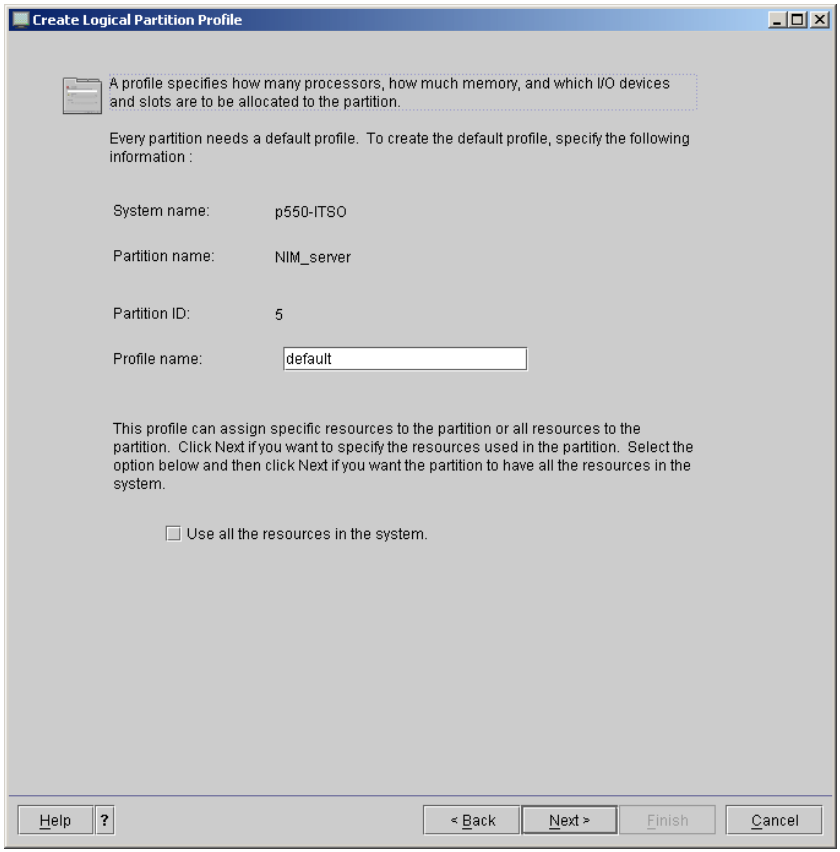


Figure 3-3 Defining the partition ID and partition name

4. Skip the definition of a workload management group by selecting the **No** button and then click **Next** (Figure 3-4).

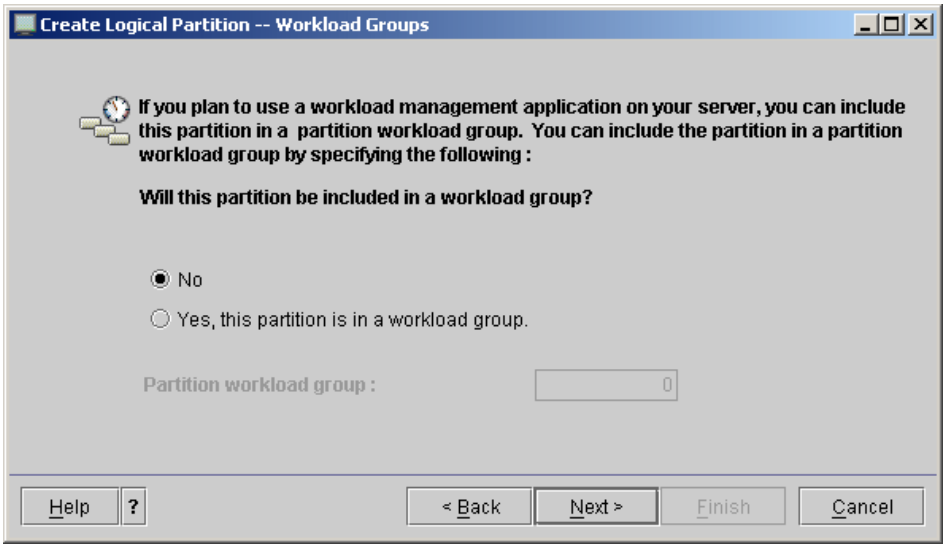


Figure 3-4 Skipping the workload management group

5. You have the option to change the Profile name, as shown in Figure 3-5. Otherwise, you can leave it as the default. In our case, we used the profile name the same as our partition name.

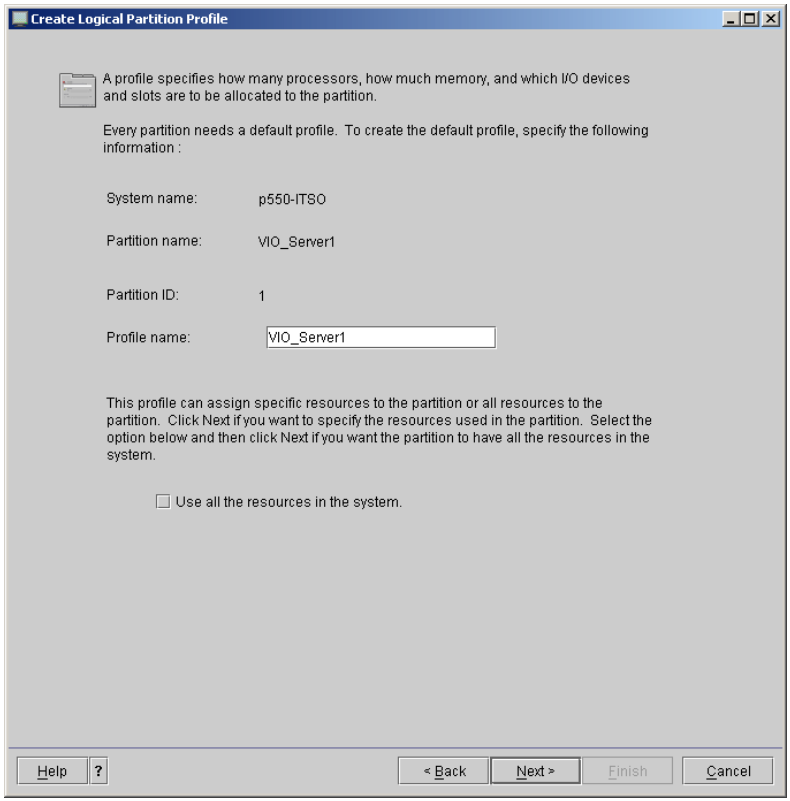


Figure 3-5 Specifying a name to the partition profile

Note: If the check box **Use all resources in the system** is checked, the logical partition being defined will get all the resources in the managed system.

6. Choose the memory settings, as shown in Figure 3-6.

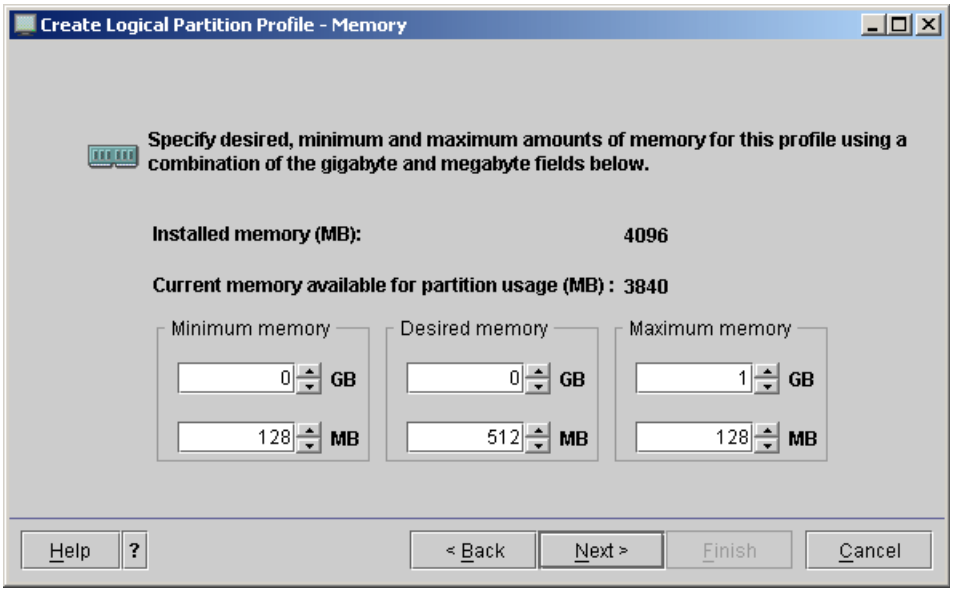


Figure 3-6 Partitions memory settings

Note: The following rules apply to Figure 3-6:

- If the managed system is not able to provide the minimum amount of memory, the partition will not start.
- You cannot dynamically increase the amount of memory in a partition to more than the defined maximum. If you want more memory than the maximum, the partition needs to be brought down and the profile updated and then restarted.
- The ratio between minimum amount of memory and maximum cannot be more than 1/64.

Important: The minimum recommended memory for a VIOS is 512 MB. Similar to an operating system, the complexity of the I/O subsystem and the number of the virtual devices have a bearing on the amount of memory required, such as when using multiple paths to SAN storage devices.

7. Select the **Shared** check box for processor allocation, as shown in Figure 3-7.

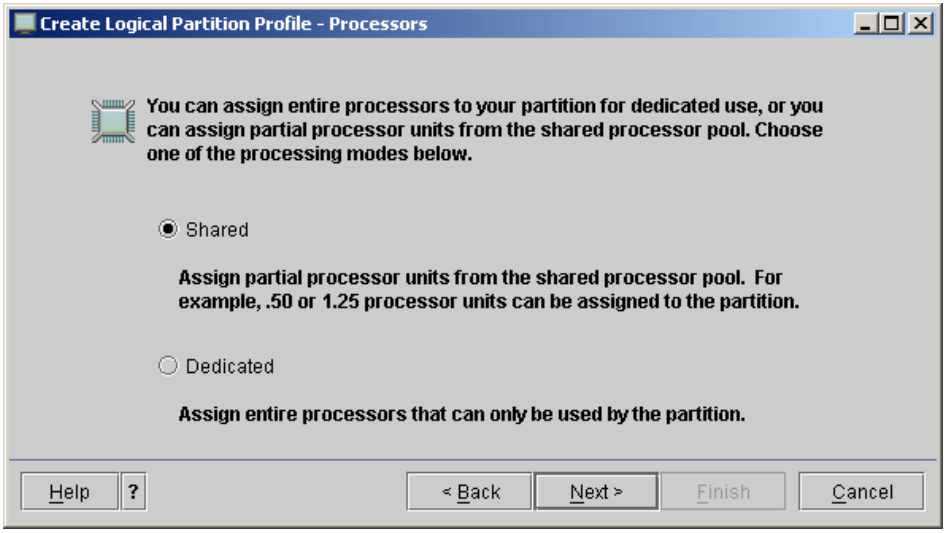


Figure 3-7 Using shared processor allocation

Note: For high Virtual I/O Server workload, we recommend having a dedicated processor allocated to the partition. We used shared processor allocation due to the limited amount physical processors in our managed system and the simplicity of setup we are trying to achieve at this point.

8. Choose the shared processor settings and specify the processing units, as shown in Figure 3-8.

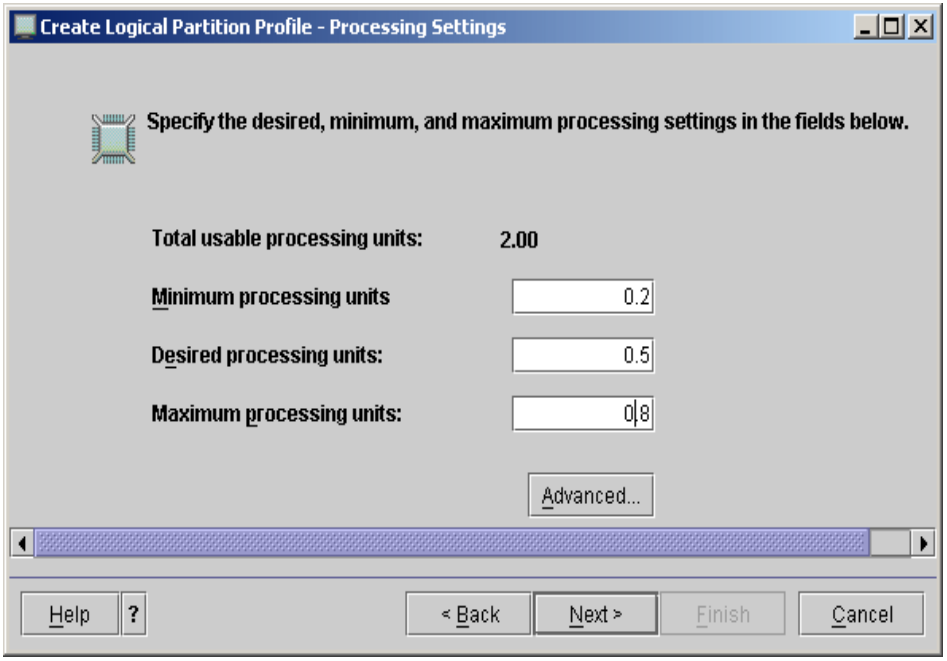


Figure 3-8 Shared processor settings

Note: The following rules apply to the processor settings:

- The partition will not start if the managed system cannot provide the minimum amount of processing units.
- You cannot dynamically increase the amount of processing units to more than the defined maximum. If you want more processing units, the partition needs to be brought down, update the profile, and then reactivate the partition.
- The maximum number of processing units cannot exceed the total Managed System processing units.

9. Specify the processing sharing mode and the virtual processor settings, as shown in Figure 3-9, by clicking the **Advanced** button. Click **OK** when you have completed the settings.

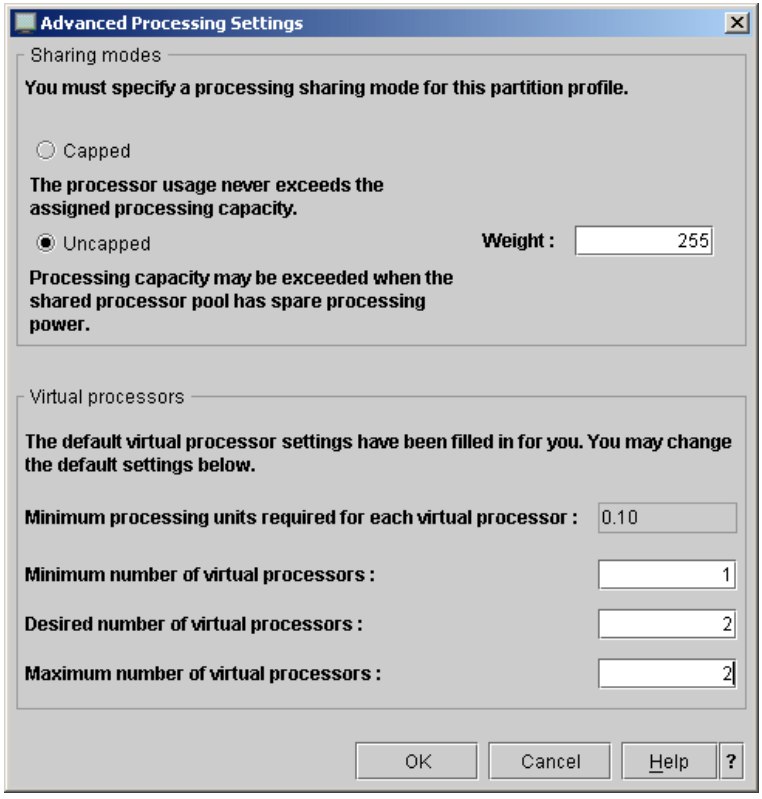


Figure 3-9 Processing sharing mode and the virtual processor settings

Note: Refer to 2.4, “Micro-Partitioning introduction” on page 32 for more information about processing units, capped and uncapped mode, and virtual processors.

Note: The system cannot utilize more virtual processors than the number of physical processors.

Tip: Consider giving the Virtual I/O Server a high weight to increase its priority. The value must be in the range 0-255 where 255 is the highest weight.

10. Select the physical resources you want to allocate to the Virtual I/O Server. For our basic configuration setup, we used a storage controller with local disks, one DVD drive, and an Ethernet adapter. The Fibre Channel adapter is for SAN disks used later. Figure 3-10 shows the selection for our basic setup. Click on **Next** when you have completed your selections.

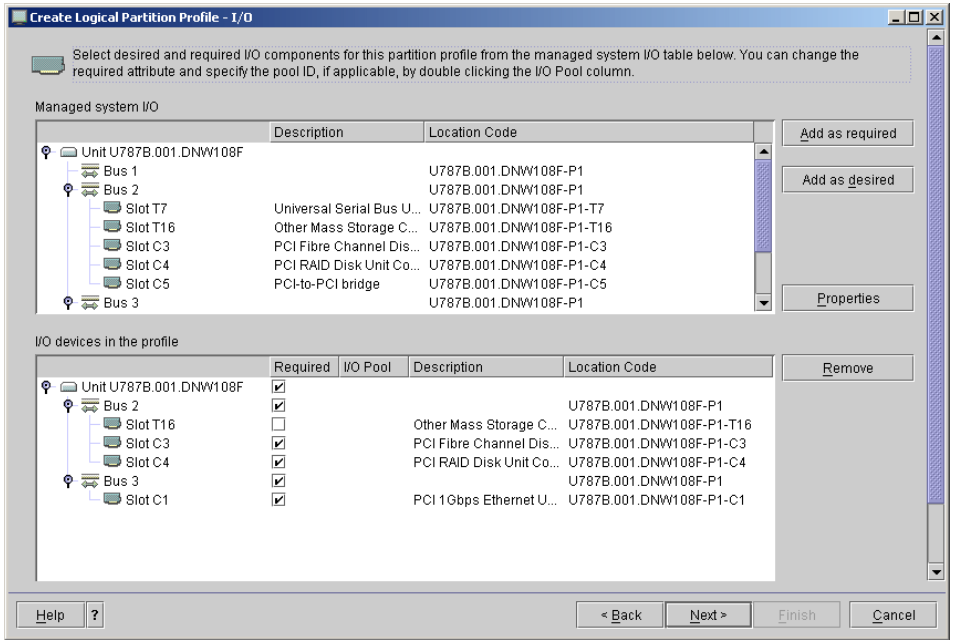


Figure 3-10 Physical I/O component selection

Note: Do not set T16 (the IDE adapter that holds the DVD) to **Required**, as it may be moved in a dynamic LPAR operation later.

Tip: If possible use hot-plug Ethernet adapters for the Virtual I/O Server for increased serviceability.

11. Skip setting the I/O pools, as shown in Figure 3-11, by clicking **Next**.

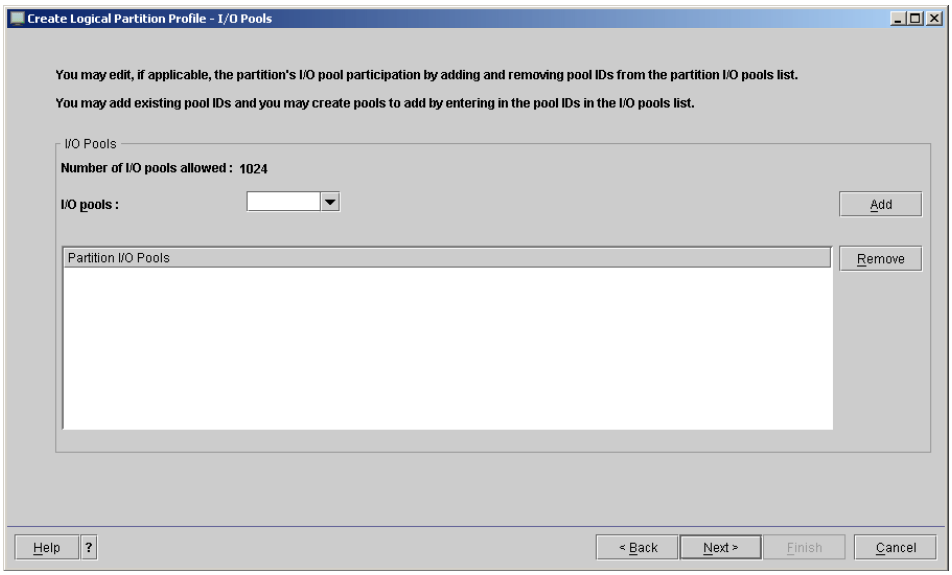


Figure 3-11 I/O Pool settings

12. Specify virtual I/O adapters by selecting the **Yes** button (see Figure 3-12).

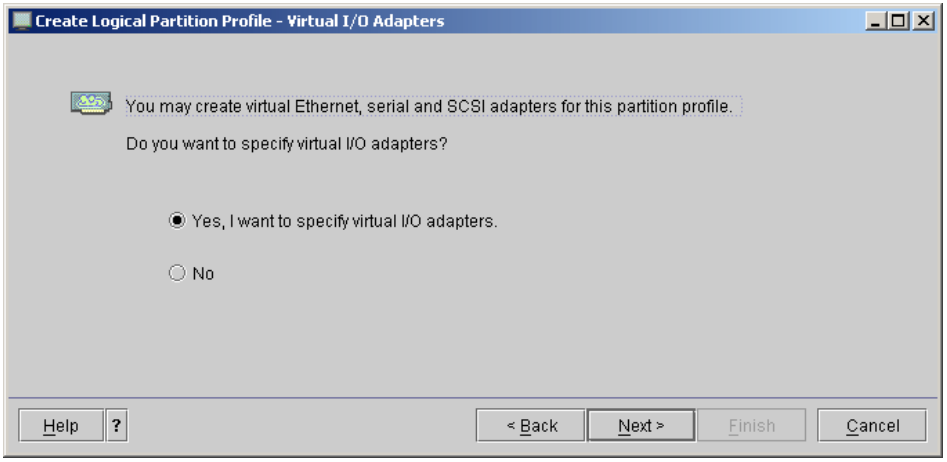


Figure 3-12 Skipping virtual I/O adapter definitions

A virtual Ethernet adapter is a logical adapter that emulates the function of a physical I/O adapter on a logical partition. Virtual Ethernet adapters enable

communication to other logical partitions within the managed system without using real hardware and cabling.

To create the adapter, perform the following steps:

1. Click the **Virtual I/O Adapters** tab and then click the **Ethernet** tab, as shown in Figure 3-13.

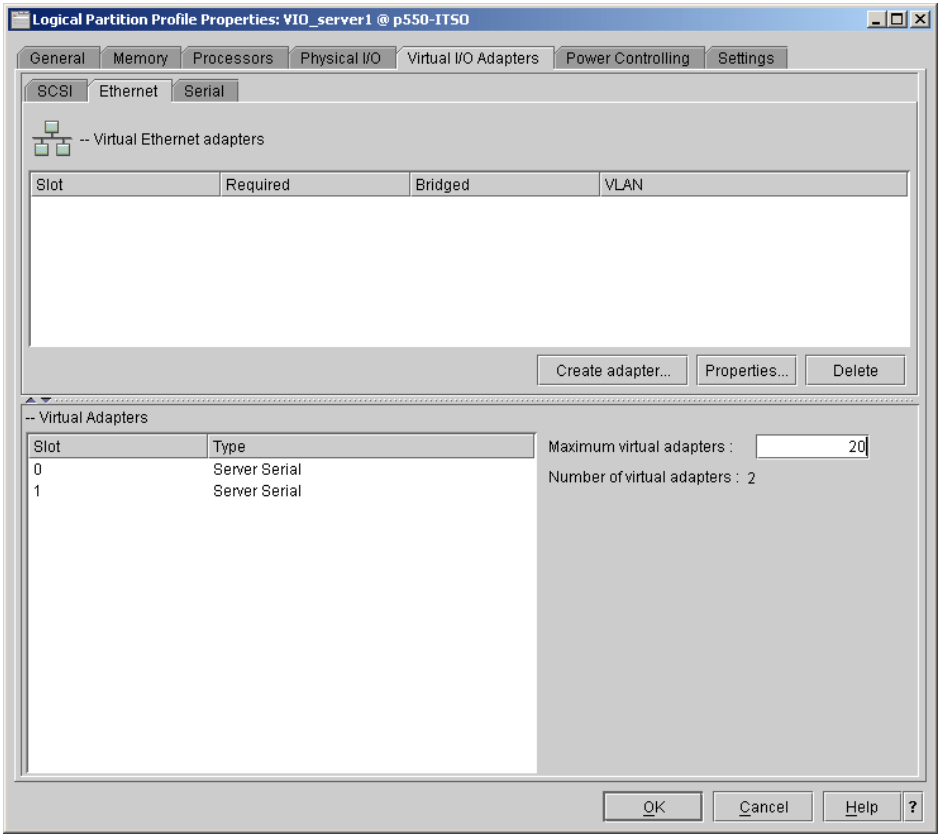


Figure 3-13 Virtual Ethernet tab

2. Click the **Create adapter** button and enter the settings. In the virtual Ethernet adapter properties, choose the slot number for the virtual adapter and virtual LAN ID, and then select the **Access External network** check box to use this adapter as a gateway between VLANs and an external network. This virtual Ethernet will be configured as a shared Ethernet adapter (see Figure 3-14).

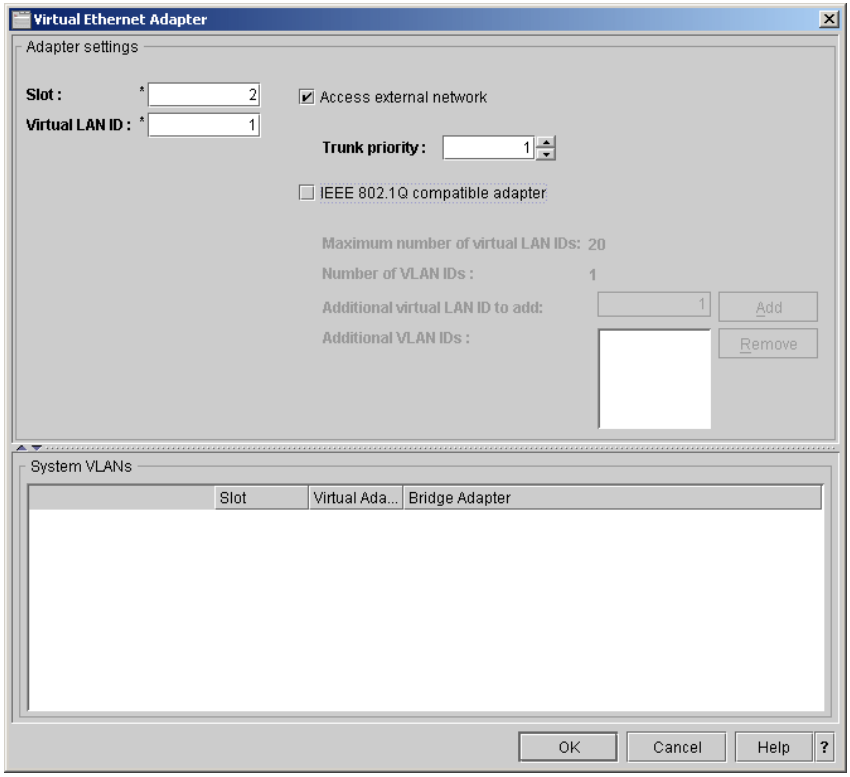


Figure 3-14 Virtual Ethernet properties

3. You can select the **IEEE 802.1Q compatible adapter** check box if you want to add additional virtual LAN IDs. In our case, we did not in order to keep our configuration basic.

Note: Selecting the **Access External Networks** check box makes sense only for a Virtual I/O Server partition. Do not select this flag when configuring the client partitions virtual Ethernet adapters. Do not create more than one Ethernet adapter with the Access External Networks check box within one VLAN.

4. Click **OK** and the virtual Ethernet adapter is ready for configuration from the command-line interface (CLI) of the Virtual I/O Server. Refer to Figure 3-15.

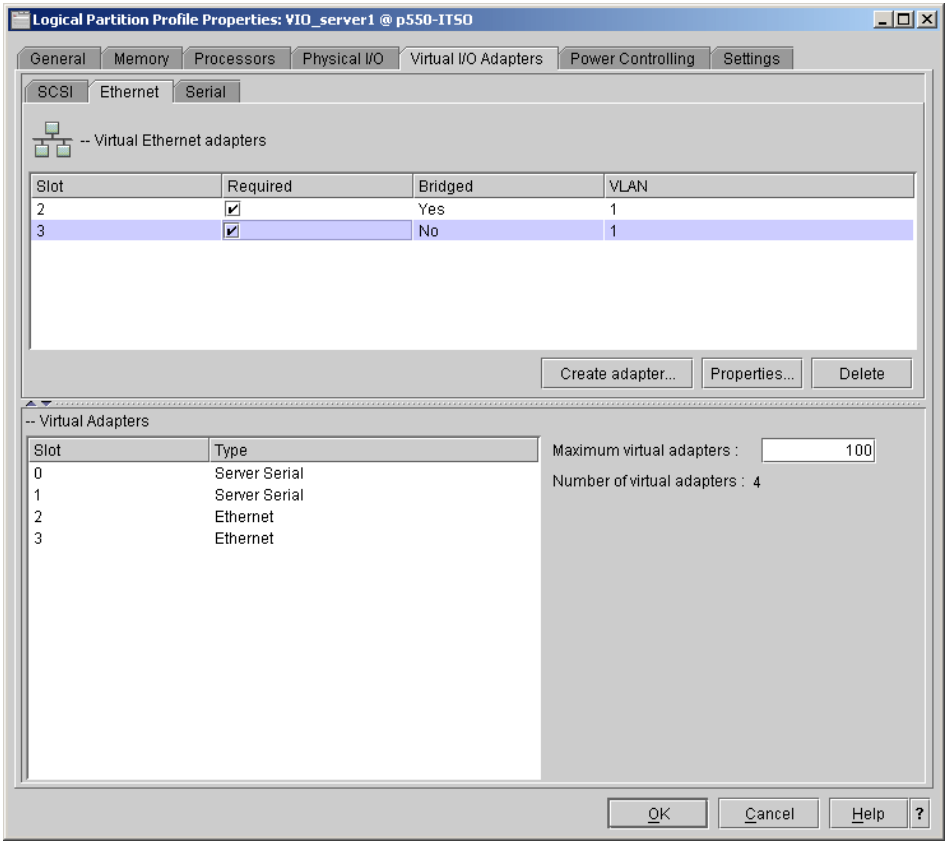


Figure 3-15 New virtual Ethernet adapter tab

We created two Ethernet adapters, one for the SEA and one to carry the IP interface. The second adapter is used to connect to the virtual network without having to define the interface on the SEA.

5. Add a virtual SCSI adapter to virtualize the DVD-ROM drive. Note the setting **Any partition can connect**. See Figure 3-16.

Note: If no virtual SCSI adapters have been defined for the Virtual I/O Server, the option to add client and server adapters is not available on the HMC when creating the profiles for the virtual clients.

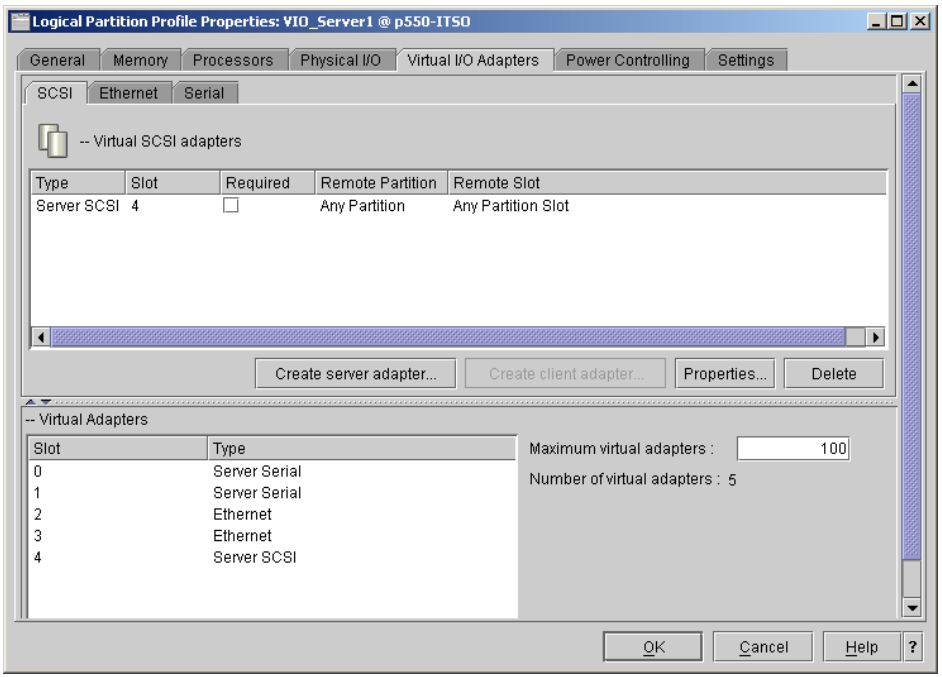


Figure 3-16 Add a virtual SCSI adapter

6. Skip the settings for power controlling partitions, as shown in Figure 3-17, by clicking **Next**.

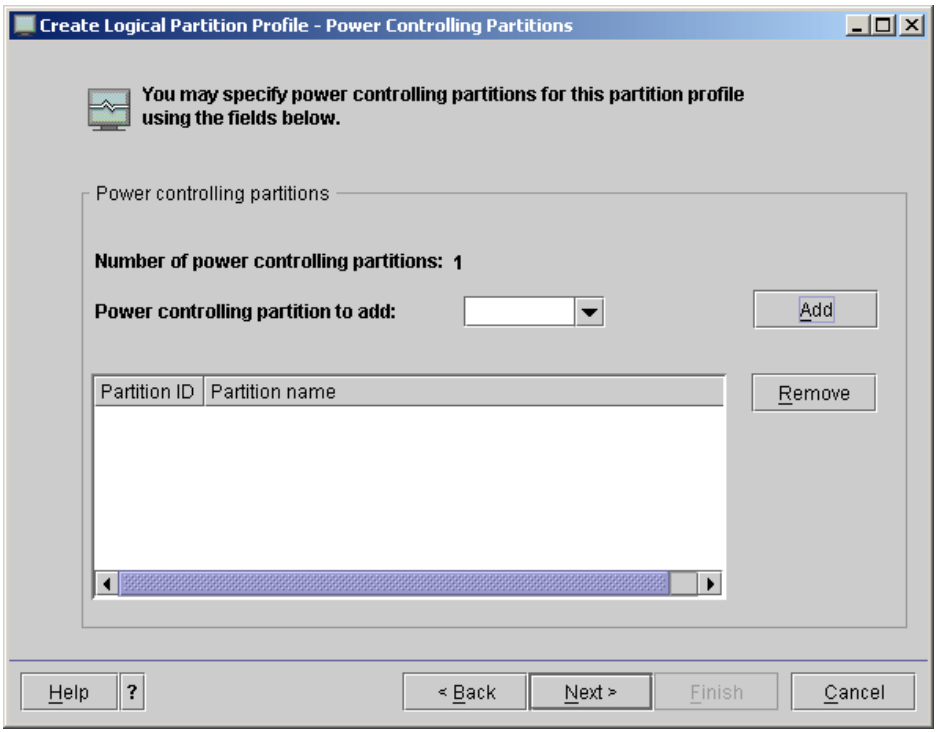


Figure 3-17 Skipping settings for power controlling partitions

7. Select **Normal** for your boot mode setting, as shown in Figure 3-18.

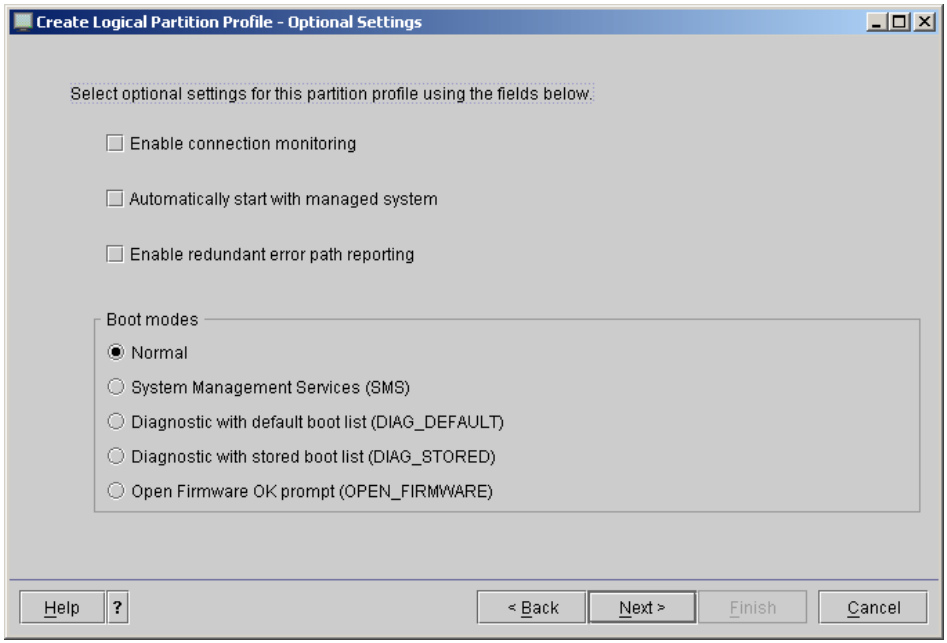


Figure 3-18 Boot mode setting selection

8. Carefully check the settings you had previously chosen for the partition, as shown in Figure 3-19, and then launch the partition creation wizard by clicking **Finish** when done.

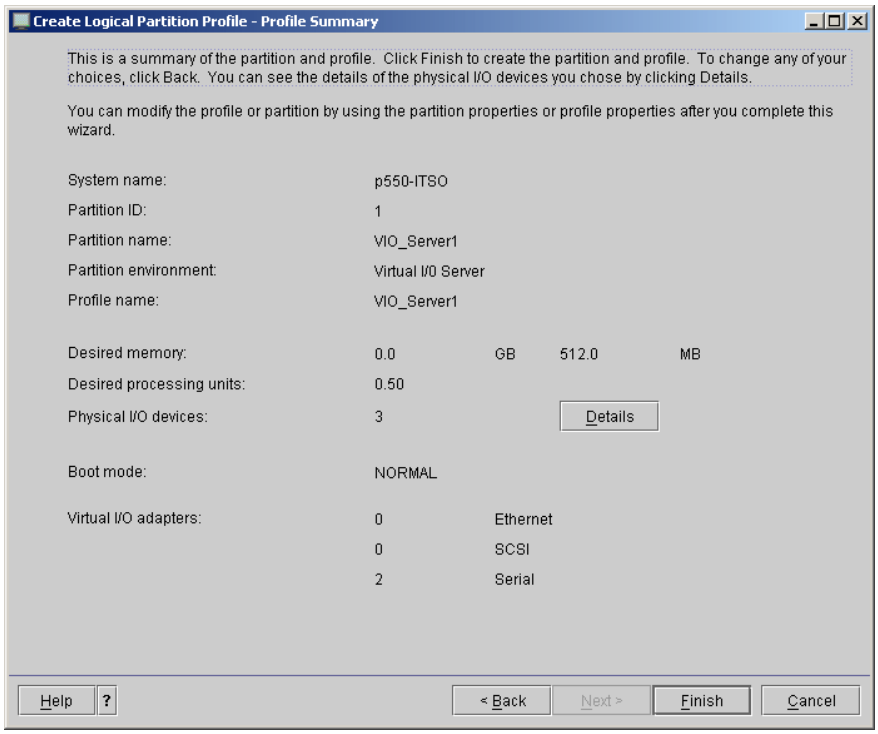


Figure 3-19 Partition settings view

The working window will be shown during the partition creation process (see Figure 3-20).

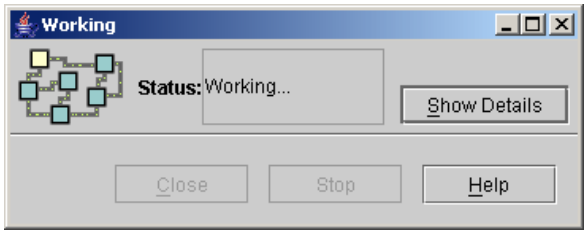


Figure 3-20 Status window

A few seconds after the status window finishes processing, you will be able to see the partition that was defined on the main Server Management window under the Partitions tab (see Figure 3-21).

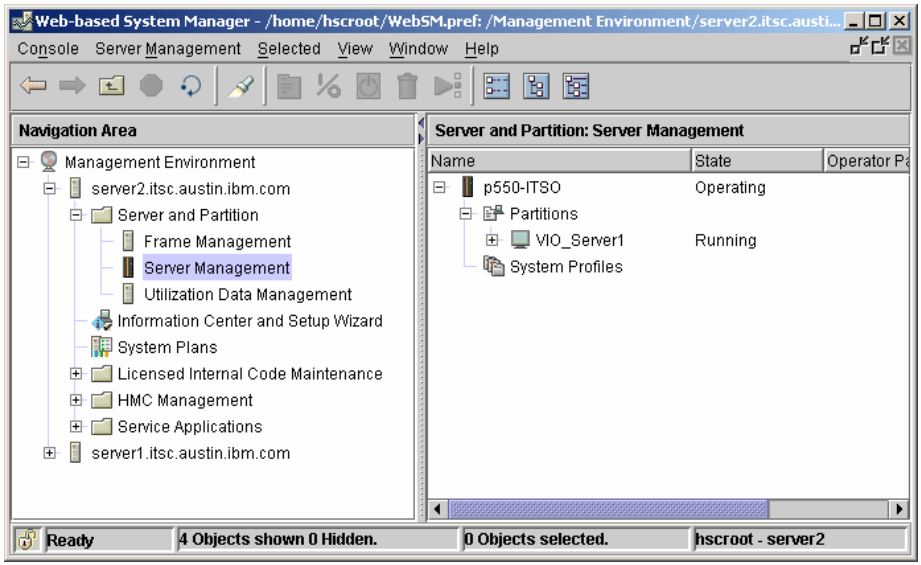


Figure 3-21 The window now shows the newly created partition VIO_Server1

3.3 Virtual I/O Server software installation

This section describes the installation of the Virtual I/O Server Version 1.3 software into the previously created Virtual I/O partition named VIO_Server1. There are three supported methods of installing the Virtual I/O Server software Version 1.3:

- Using the optical drive allocated to the Virtual I/O Server partition and booting from it.
- Installing the VIOS software from the HMC using the **`installios`** command, which is using NIM for a network installation. If you just enter **`installios`** without any flags, a wizard will be invoked and the you will be prompted to interactively enter the information contained in the flags. The default is to use the optical drive on the HMC for the Virtual I/O Server installation media, but you can also specify a remote file system instead. See the HMC Information Center for details on how to use **`installios`** from the HMC.

Note: A network adapter with connection to the HMC network is required for the Virtual I/O Server installation.

- When installing the media using NIM, the **installios** command is also available in AIX 5L both for the NIM server and any NIM client. If you run the **installios** command on a NIM client, you are prompted for the location of the bos.sysmgt.nim.master fileset. The NIM client is then configured as a NIM master. Use the following link and search for **installios** for additional information:

  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp

Tip: If you plan on using two Virtual I/O Servers (described in 4.1, “Providing higher serviceability” on page 182), you could install the first server, apply updates, multipath drivers and so on, then make a NIM backup and use the image for installing the second Virtual I/O Server.

The following steps show the installation using the optical install device:

1. Place the Virtual I/O Server Version 1.3 DVD disk in the drive.
2. Activate the VIO_Server1 partition by right-clicking the partition name and selecting the **Activate** button, as shown in Figure 3-22. Select the default profile you used to create this partition.

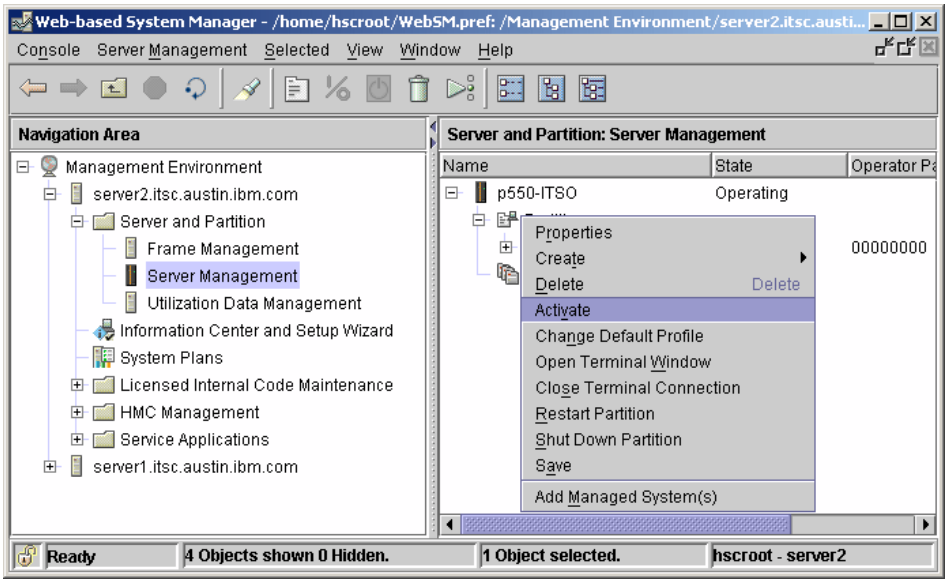


Figure 3-22 Àctivate VIO_Server1 partition

3. Select the VIO_Server1 profile and then check the **Open a terminal window or console session** check box, as shown in Figure 3-23, and then click the **Advanced** tab.

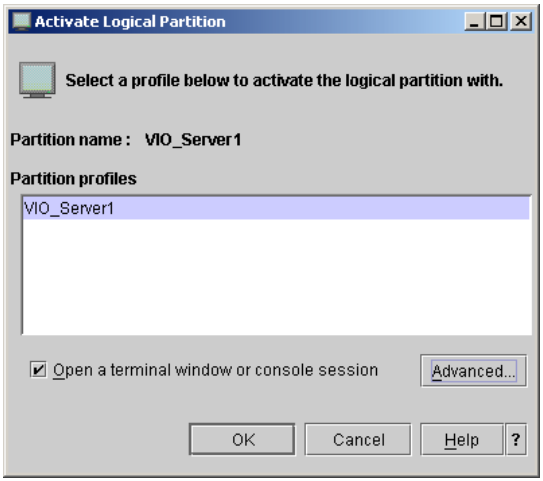


Figure 3-23 Selecting the profile

4. Under the Boot Mode drop-down list, choose **SMS**, as shown in Figure 3-24, and then click **OK**.

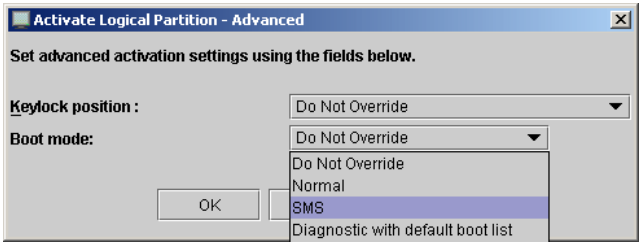


Figure 3-24 Selecting SMS boot mode

5. Figure 3-25 shows the SMS menu after booting the partition on SMS mode.

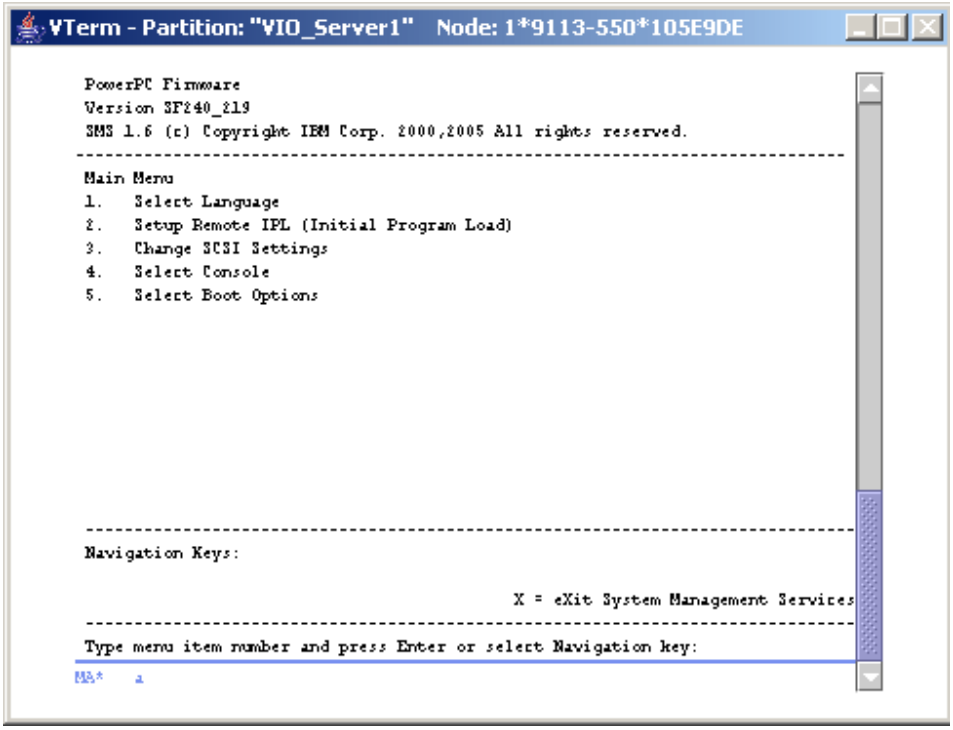


Figure 3-25 SMS menu

6. Follow the steps below to continue and boot the Virtual I/O Server partition:
   a. Choose 5 for Select Boot Options and then press Enter.
   b. Choose 1 for Select Install/Boot Device and then press Enter.
   c. Choose 3 for CD/DVD and then press Enter.
   d. Choose 4 for IDE then press Enter.
   e. Choose 1 for IDE CD-ROM and then press Enter.
   f. Choose 2 for Normal Mode Boot and then press Enter.
   g. Confirm your choice with 1 for Yes and then press Enter.
7. When the installation procedure has finished, use the padmin user name to login. Upon initial login, you will be asked to supply the password.

After logging in successfully, you will be placed under the Virtual I/O Server command line interface (CLI). Type in the command below to accept the license:

```
$ license -accept
```

You are now ready to use the newly installed Virtual I/O Server software.

3.4 Basic Virtual I/O Server scenario

In this section, a simple configuration consisting of a single Virtual I/O Server partition servicing virtual SCSI devices to four logical partitions is shown. Refer to Figure 3-26 for the basic configuration scenario.

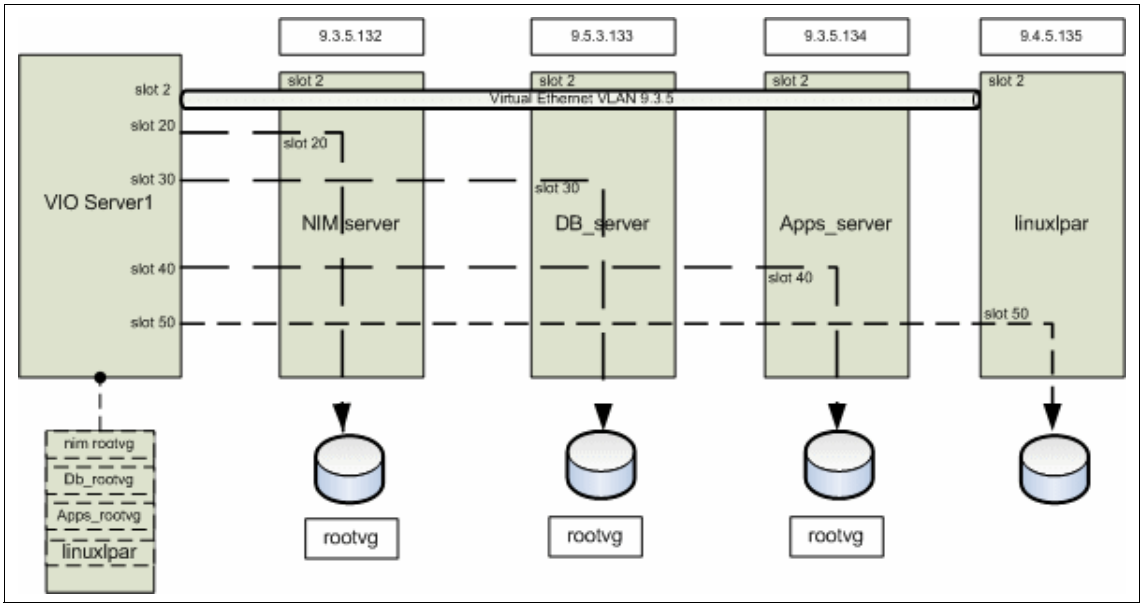


Figure 3-26 Basic Virtual I/O Server scenario

Based on Figure 3-26, the following sections will guide you through configuring virtual Ethernet and virtual SCSI adapters.

3.4.1 Creating virtual SCSI server adapters

At this stage, the clients are not known to the HMC. If you create the SCSI server adapters now, you will have to specify the partition ID of the client (from your planning) or you specify that **Any client can connect**, which means you will have to change this after you have created the clients.

A more efficient procedure would be to leave the creation of SCSI adapters until you create the client LPARs. In the HMC menus, you can create the client and server adapter simultaneously. This procedure requires that the network is configured with connection to the HMC to allow for dynamic LPAR.

Follow these steps to create virtual SCSI server adapters:

1. Right-click the VIO_Server1 partition profile and go to the **Properties** tab.
2. Click the **Virtual I/O Adapters** tab and then click the **SCSI** tab, as shown in Figure 3-27.

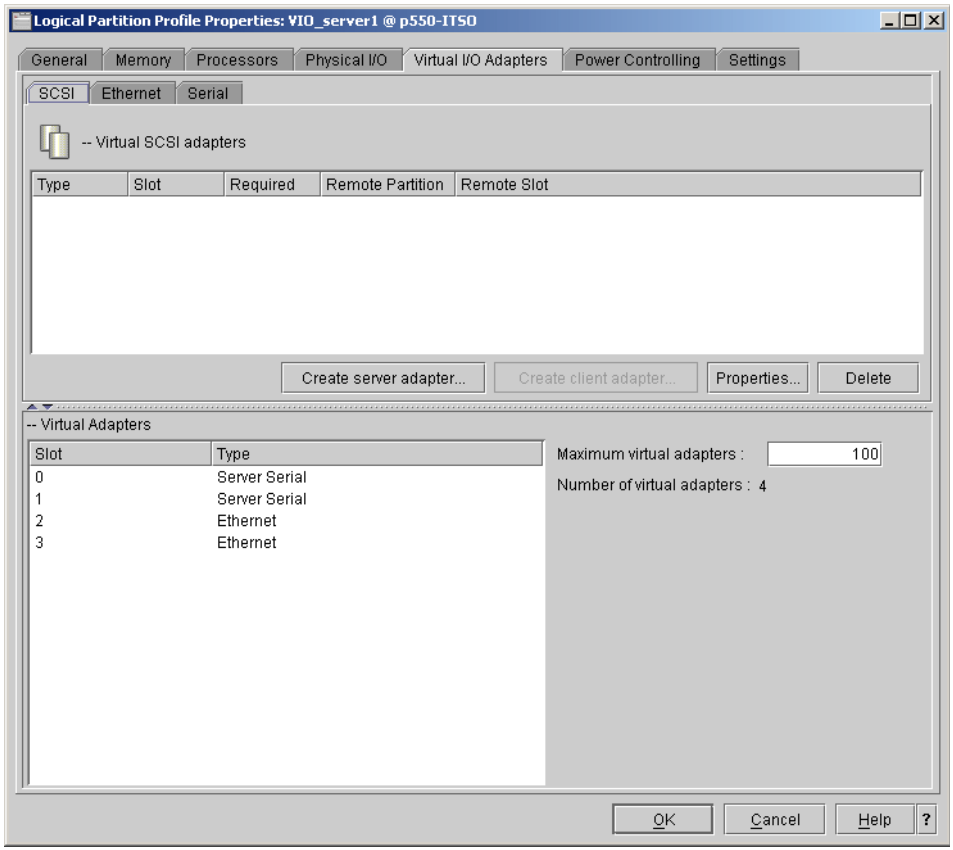


Figure 3-27 SCSI properties tab

3. Specify the **Maximum virtual adapters** that will be supported by the Virtual I/O Server and then click **Create server adapter**. The drop-down list under Client partition allows you to choose which partition can use the slot, as shown in Figure 3-28. Click **OK** when done.

Tip: We recommend that you increase the maximum number of adapters to suit your configuration. Changing this value can only be done in the profile and to activate this setting you must change the profile and reactivate the LPAR.

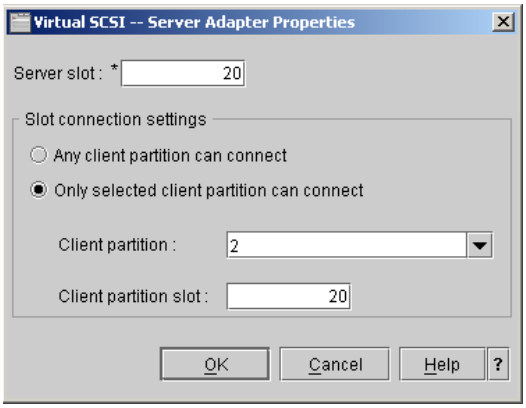


Figure 3-28 Server adapter properties

Note: Our experience tells us that it is good to have a server slot number consistent with the client slot number. You will save yourself a lot of time figuring out the slot mappings later on.

An attempt to specify a non-existing client partition name instead of a partition ID causes the error message shown in Figure 3-29.

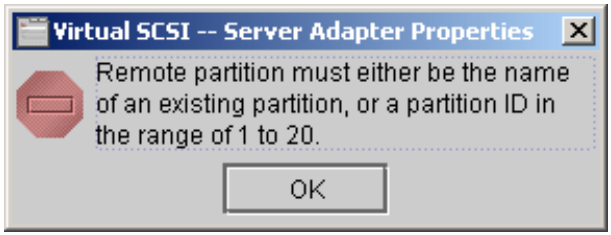


Figure 3-29 HMC message

3.4.2 Creating a Shared Ethernet Adapter

It is important to create the Shared Ethernet Adapter before creating the client profiles since simultaneous creation of client/server SCSI adapters relies on dynamic LPAR being available.

To create a Shared Ethernet adapter, perform the following steps:

1. Use the **lsdev** command on the Virtual I/O Server to verify that the Ethernet trunk adapter is available (see Example 3-3).

Example 3-3 Check for shared Ethernet adapter

```
$ lsdev -virtual
name             status     description
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0           Available  Virtual SCSI Server Adapter
vhost1           Available  Virtual SCSI Server Adapter
vhost2           Available  Virtual SCSI Server Adapter
vhost3           Available  Virtual SCSI Server Adapter
vsa0             Available  LPAR Virtual Serial Adapter
vapps            Available  Virtual Target Device - Logical Volume
vdbsrv           Available  Virtual Target Device - Logical Volume
vlnx             Available  Virtual Target Device - Logical Volume
vnim             Available  Virtual Target Device - Logical Volume
```

2. Select the appropriate physical Ethernet adapter that will be used to create the Shared Ethernet Adapter. The **lsdev** command will show a list of available physical adapter (see Example 3-4).

Example 3-4 Check for physical Ethernet adapter

```
$ lsdev -type adapter
name             status     description
ent0             Available  10/100/1000 Base-TX PCI-X Adapter (14106902)
ent1             Available  Virtual I/O Ethernet Adapter (l-lan)
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
ent3             Available  Shared Ethernet Adapter
fcs0             Available  FC Adapter
ide0             Available  ATA/IDE Controller Device
sisioa0          Available  PCI-X Dual Channel U320 SCSI RAID Adapter
vsa0             Available  LPAR Virtual Serial Adapter
```

You can use the **lsmap -all -net** command to check the slot numbers of the virtual Ethernet adapters. We use ent1 in slot C2 (see Example 3-5).

Example 3-5 Checking slot numbers

```
$ lsmap -all -net
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V1-C2-T1

SEA                 NO SHARED ETHERNET ADAPTER FOUND
------ --------------------------------------------
ent2   U9113.550.105E9DE-V1-C3-T1

SEA                 NO SHARED ETHERNET ADAPTER FOUND
```

Use the **mkvdev** command to create a new ent3 device as the Shared Ethernet Adapter. ent0 will be used as the physical Ethernet adapter and ent1 as the virtual Ethernet adapter (Example 3-6).

Example 3-6 Create Shared Ethernet Adapter

```
$ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1
ent3 Available
en3
et3
```

3. Confirm that the newly created Shared Ethernet Adapter is available using the **lsdev -virtual** command (Example 3-7).

Example 3-7 Confirm Shared Ethernet device

```
$ lsdev -virtual
name            status     description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vsa0            Available  LPAR Virtual Serial Adapter
ent3            Available  Shared Ethernet Adapter
```

The Shared Ethernet Adapter will form a bridge, allowing communication between the inter-partition VLAN and the external network.

Based on our basic scenario, we only have one physical network connection to the public network that is through the physical Ethernet, so we configure the Shared Ethernet Adapter to act as a bridge between the public network and the inter-partition VLAN.

We used the following values for our scenario (Table 3-1).

Table 3-1 Network settings

Settings	Value
hostname	VIO_Server1
IP-address	9.3.5.110
netmask	255.255.255.0
gateway	9.3.5.41

We chose to configure the IP interface on the additional virtual Ethernet adapter instead of the SEA. Use the **mktcpip** command to configure the interface on the virtual Ethernet adapter, ent2. (see Example 3-8).

Note: There is no performance penalty for adding the IP address to the SEA interface instead of keeping it on a separate virtual Ethernet adapter. However, the SEA can be redefined without having to detach the interface when the interface is kept on a separate virtual Ethernet adapter. This provides increased serviceability.

Example 3-8 Define the Shared Ethernet Adapter

```
$ mktcpip -hostname VIO_Server1 -inetaddr 9.3.5.110 -interface en3
-netmask 255.255.255.0 -gateway 9.3.5.41
```

3.4.3 Creating client partitions

This section shows you how to create the four client partitions for our basic Virtual I/O scenario. The definitions are similar to the creation of our Virtual I/O Server partition, but instead of clicking **Virtual I/O**, we choose **AIX or Linux**. For the client partitions we are creating, refer to Figure 3-30 on page 152. Follow the steps below to create the client partitions:

1. Restart the Create Logical Partition Wizard, as you did at the beginning of the previous process 3.2.1, “Defining the Virtual I/O Server partition” on page 124. Refer to Figure 3-2 on page 125 for a view of the window.

2. Select the check box **AIX or Linux** and enter a partition ID and partition name, as shown in Figure 3-30.

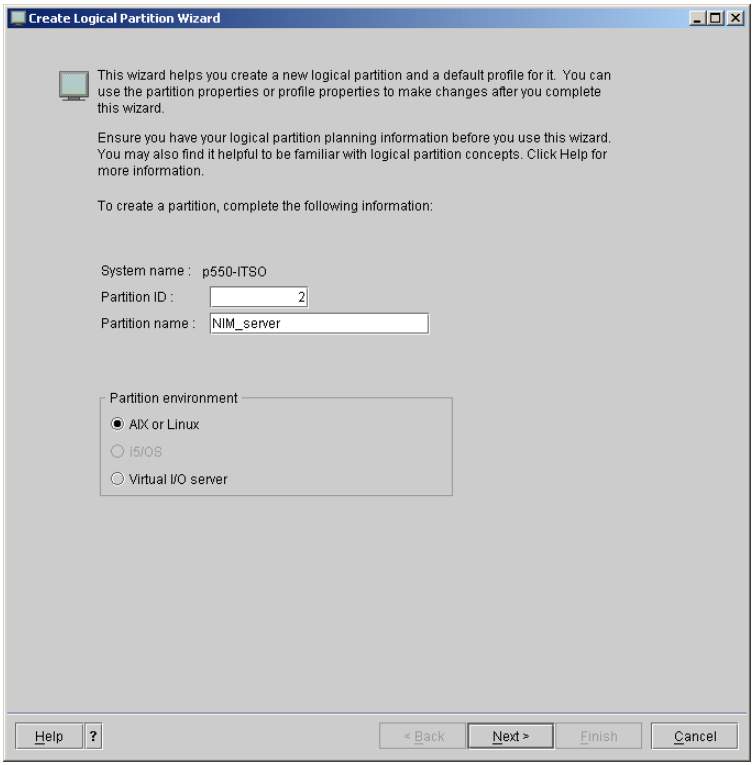


Figure 3-30 Creating NIM_server partition

3. Repeat steps 4 to 15 of 3.2.1, “Defining the Virtual I/O Server partition” on page 124 with the following exceptions:
   a. Make a note of the Partition ID, as you will need this later.
   b. Use 256/512/768 MB for minimum/desired/maximum memory settings on step 6, or appropriate values for your configuration.
   c. Use 0.1/0.2/0.4 processing units for minimum/desired/maximum settings on step 8, or appropriate values for your configuration.
   d. Use 1/2/2 for minimum/desired/maximum virtual processor settings on step 9. We set the maximum to two since we use a system with two processors and the system cannot utilize more than the number of physical processors.
4. Skip step 10 by clicking **Next** instead of choosing to create a physical I/O component selection since we are not using physical adapters in our clients.

5. Create one virtual Ethernet adapter, as shown in 3.4.2, “Creating a Shared Ethernet Adapter” on page 149. Do not check the Access external network box.

Note: We increased the maximum number of adapters to 100.

Create the client and server SCSI adapters. We chose one SCSI adapter for disk and one SCSI adapter for the virtual optical device. Follow these steps to accomplish this:

1. Select the **SCSI** tab. You will see a window similar to Figure 3-31.

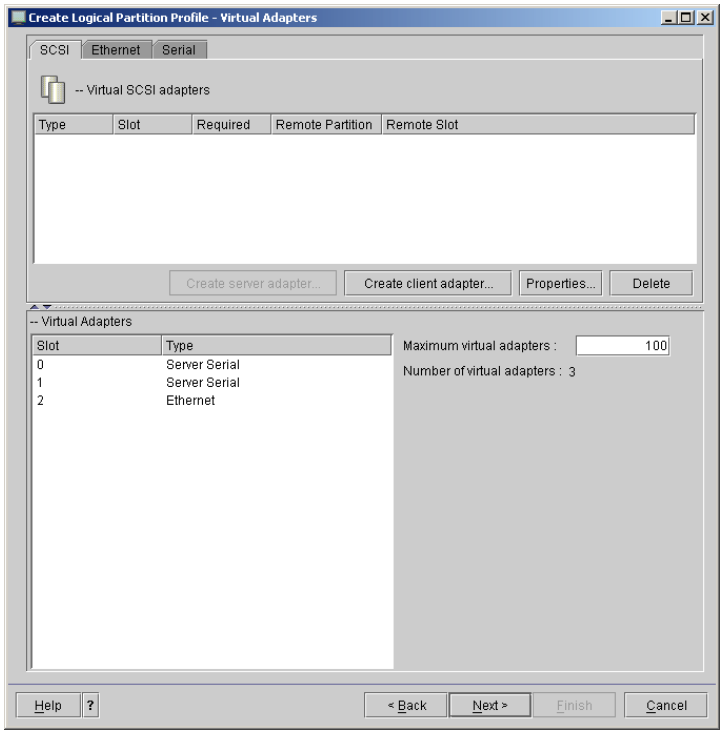


Figure 3-31 Create client virtual SCSI adapter

2. Click **Create client adapter**. The window in Figure 3-32 appears.

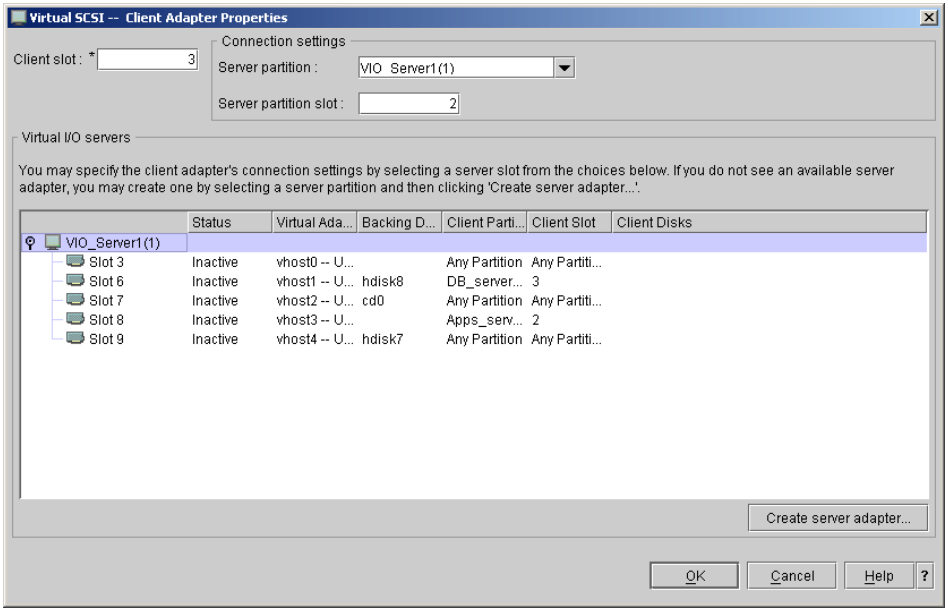


Figure 3-32 Client adapter properties

3. Select the **Server partition** from the drop-down menu. Since you do not have a suitable server adapter from the list, you can create one. This function relies on the network being set up and dynamic LPAR operations working. Point to the Virtual I/O Server and select **Create server adapter**. See Figure 3-33.

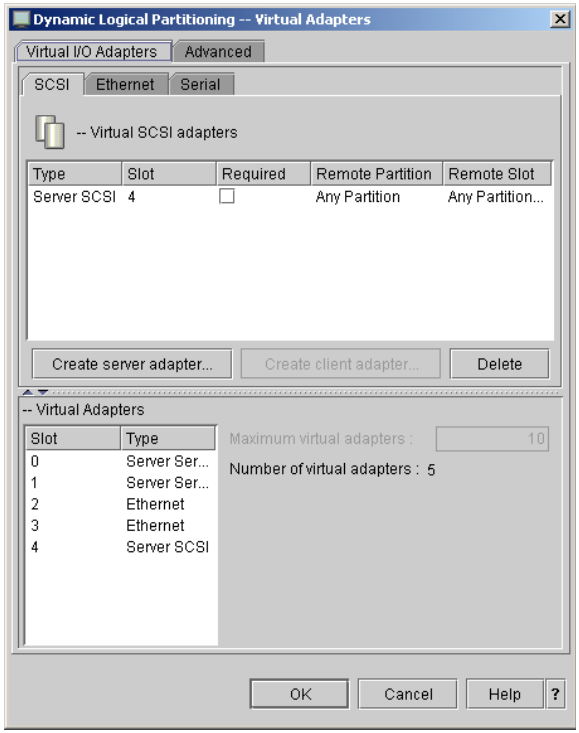


Figure 3-33 Dynamic Logical Partitioning dialog

4. Click **Create server adapter** and a properties dialog appears, as shown in Figure 3-34.

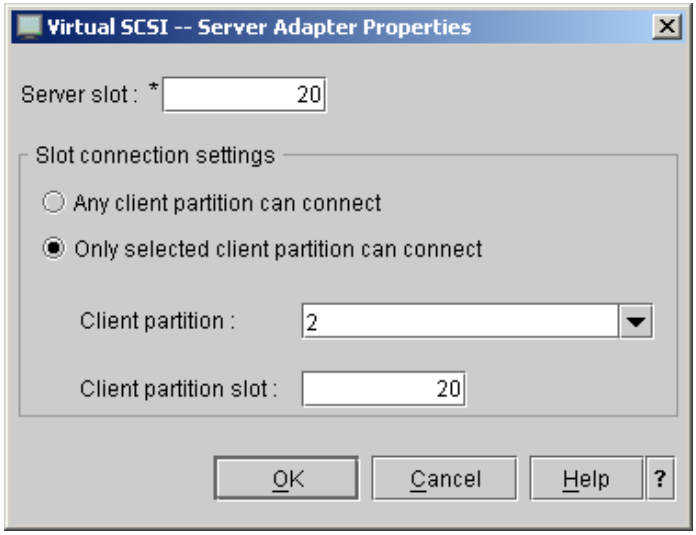


Figure 3-34 Server adapter properties

An available server slot is selected for you by the system. If you want to change it, make sure it is an available slot. Select **Only selected client partition can connect**. Type the **Client partition** number since the partition name is not known to the HMC yet. Type in the **Client partition slot** number from the client adapter dialog. Click **OK**.

5. You will see the server adapter in the list with the correct slot client/server slot numbers. Click **OK** to perform the dynamic LPAR operation (Figure 3-35).

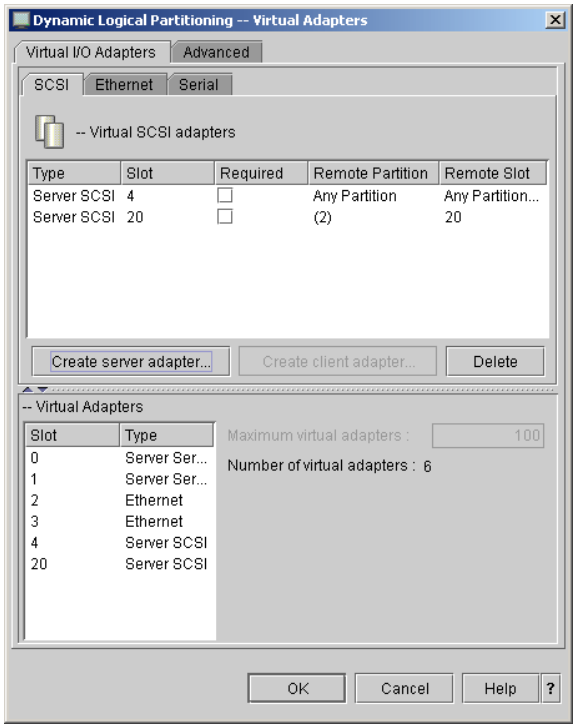


Figure 3-35 Dynamic Logical Partitioning

6. When the operation is complete, you will get a window with the correct client/server slot numbers. Remember to update the correct server and client partition slot numbers for the new adapter according to the list on the window, as shown in Figure 3-36.

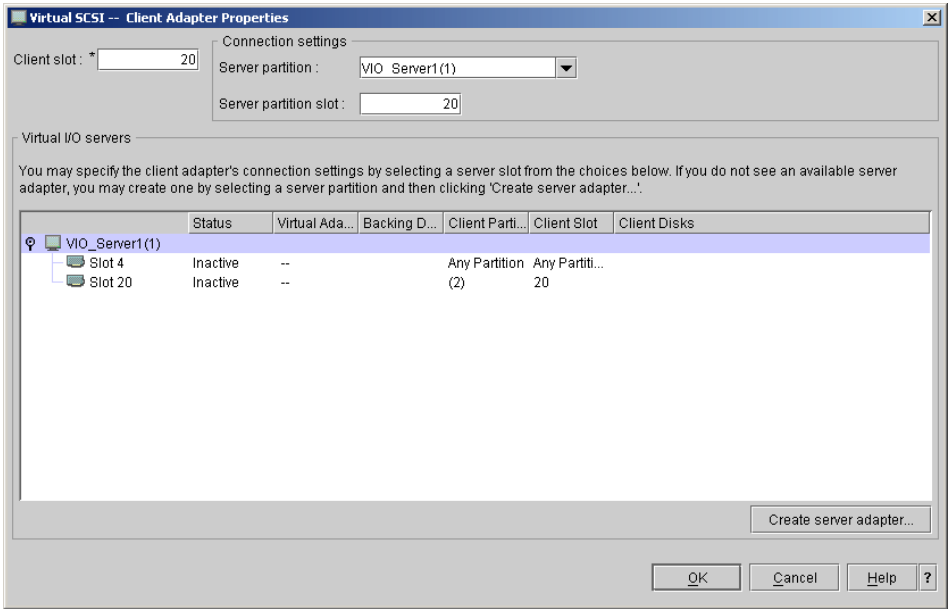


Figure 3-36 Client Adapter properties

Note: After the partition is created, subsequent Virtual I/O Server operations will reflect the name of the partition and the partition number.

7. Click **OK** to complete the create client adapter operation.

8. Create another client adapter by pointing to the server adapter marked **Any partition**, fill in the client slot number you want to use (here we use 99), and click **OK**. See Figure 3-37. We will use this adapter for the virtual DVD. See section “Virtual SCSI optical devices” on page 95 for the steps required to setup the virtual optical device.

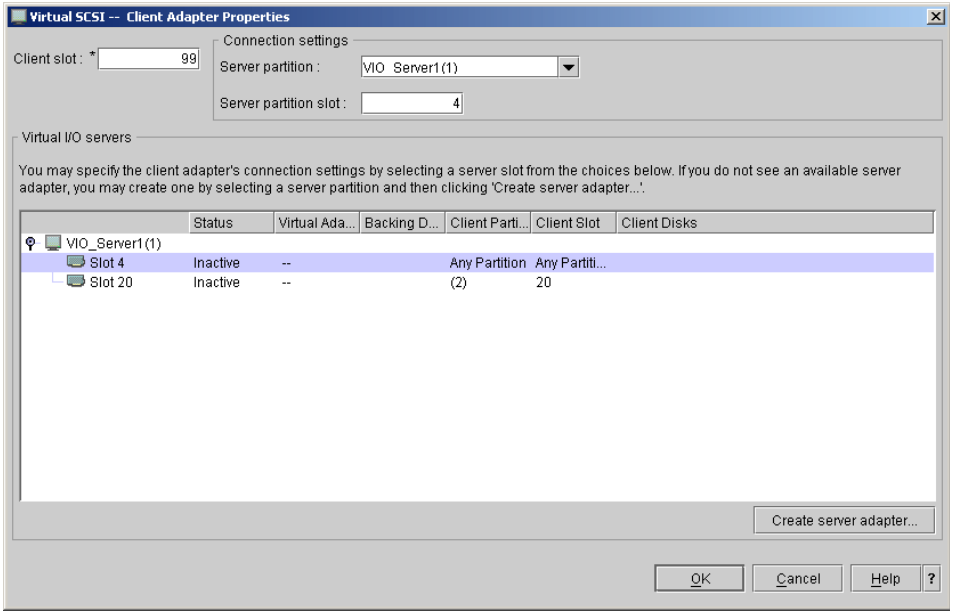


Figure 3-37 Adding the virtual SCSI adapter for the DVD

9. When you have completed adding virtual adapters, you will see a window similar to Figure 3-38.

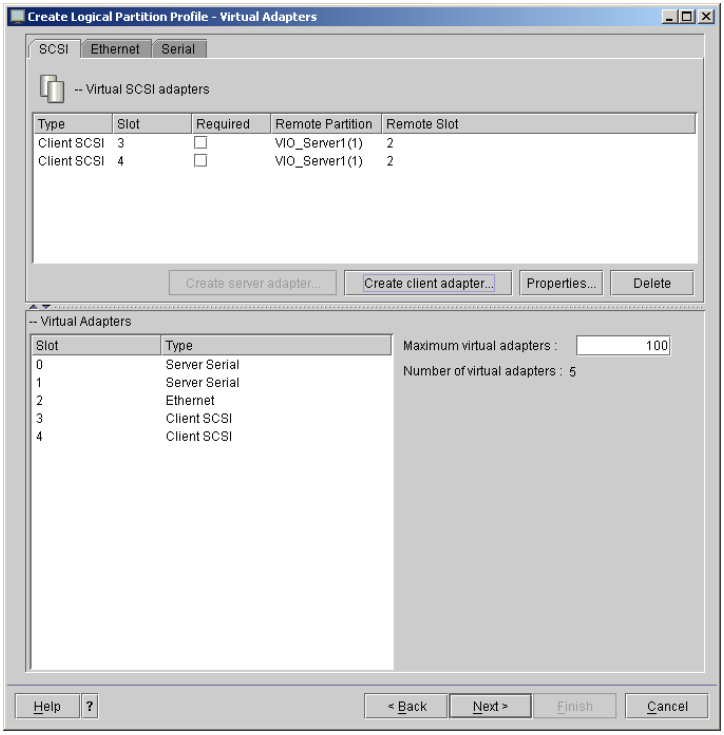


Figure 3-38 Virtual adapters window

10. Clicking **Next** brings you to the Power Controlling Partitions window. Skip this by clicking **Next** again and go to the Optional Settings window. Skip this for now by clicking **Next**. You then get to the Partition Summary window. Review this window before clicking **Finish** to create the partition.

Figure 3-39 shows the HMC partitions when you have finished creating client partitions.

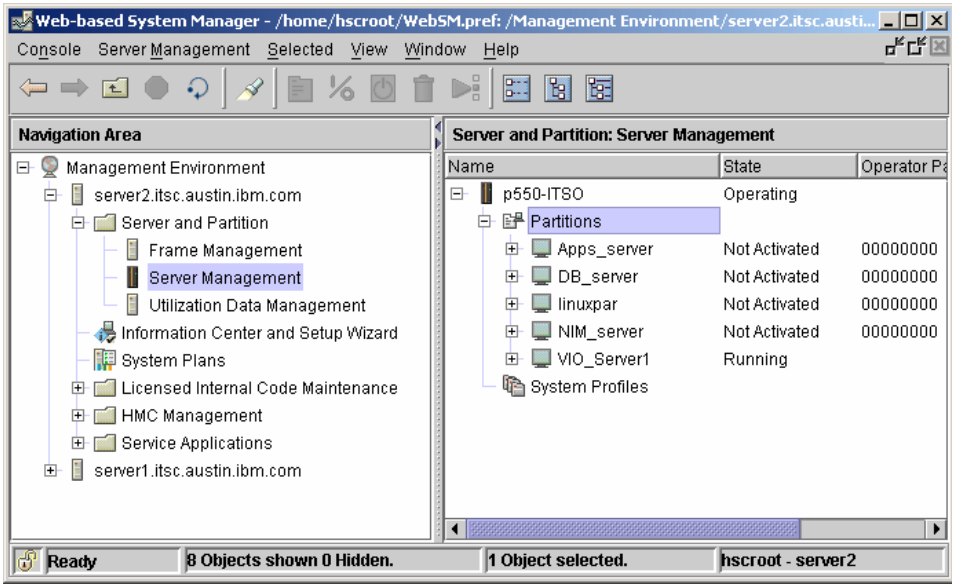


Figure 3-39 HMC view with new partitions created

Server SCSI adapters have been added using dynamic LPAR. For the configuration to be permanent across restarts (not just reboot), the profile needs to be updated. Alternatively, you can save the current configuration to a new profile, as shown in Figure 3-40.

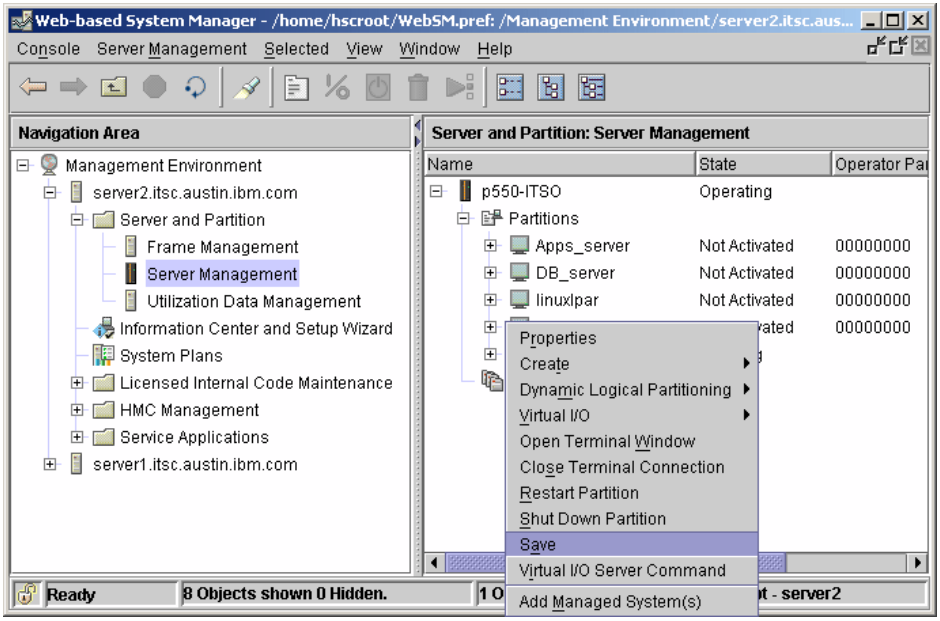


Figure 3-40 Saving the current configuration to a new profile

Figure 3-41 shows the Save Partition window.

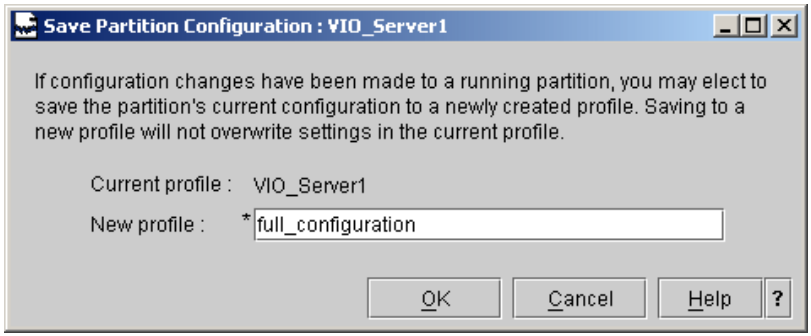


Figure 3-41 Saving a new profile called full_configuration

If you want the new profile to be the default, right-click the partition and select **Change Default Profile**. Click **OK** to set the profile as default. If you have several profiles, use the drop-down menu to select the correct profile.

3.4.4 Defining virtual disks

Virtual disks can either be whole physical disks or logical volumes. The physical disks can either be local disks or SAN attached disks.

SAN disks can be used both for the Virtual I/O Server rootvg and for virtual I/O clients disks.

Tip: A virtual disk, physical volume, can be mapped to more than one partition by using the -f option of the **mkvdev** command for the second mapping of the disk. This could be used for concurrent-capable disks between partitions.

Use the following steps to build the logical volumes required to create the virtual disk for the clients partition's rootvg based on our basic scenario using logical volumes:

1. Log in with the padmin user ID and run the **cfgdev** command to rebuild the list of visible devices used by the Virtual I/O Server.

   The virtual SCSI server adapters are now available to the Virtual I/O Server. The name of these adapters will be vhost*x,* where x is a number assigned by the system.

2. Use the **lsdev -virtual** command to make sure that your new virtual SCSI adapter is available, as shown in Example 3-9.

Example 3-9 Virtual I/O Command Line Interface

```
$ lsdev -virtual
name             status     description
ent1             Available  Virtual I/O Ethernet Adapter (l-lan)
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0           Available  Virtual SCSI Server Adapter
vhost1           Available  Virtual SCSI Server Adapter
vhost2           Available  Virtual SCSI Server Adapter
vhost3           Available  Virtual SCSI Server Adapter
vhost4           Available  Virtual SCSI Server Adapter
vsa0             Available  LPAR Virtual Serial Adapter
ent3             Available  Shared Ethernet Adapter
```

3. Use the **lsmap -all command** to check slot numbers as shown:

   If the devices are not available, then there was a problem defining them. You can use the **rmdev -dev vhost0 -recursive** command for each device and then reboot the Virtual I/O Server if needed. Upon reboot, the configuration manager will detect the hardware and re-create the vhost devices.

Using logical partitions

In our basic scenario, we will create the volume group named rootvg_clients on hdisk2 and partition it to serve as boot disks to our client partitions.

Important: We do not recommend using the Virtual I/O Server rootvg disk for virtual client disks (logical volumes).

1. Create a volume group and assign hdisk2 to rootvg_clients using the **mkvg** command, as shown in Example 3-10.

Example 3-10 Creating the rootvg_clients volumegroup

```
$ mkvg -f -vg rootvg_clients hdisk2
rootvg_clients
```

2. Define all the logical volumes that are going to be presented to the client partitions as hdisks. In our case, these logical volumes will be our rootvg for the client partitions (see Example 3-11).

Example 3-11 Create logical volumes

```
$ mklv -lv rootvg_dbsrv rootvg_clients 10G
rootvg_dbsrv
$ mklv -lv rootvg_apps rootvg_clients 10G
rootvg_apps
$ mklv -lv rootvg_nim rootvg_clients 10G
rootvg_nim
$ mklv -lv rootvg_lnx rootvg_clients 2G
rootvg_lnx
```

3. Define the SCSI mappings to create the virtual target device that associates to the logical volume you have defined in the previous step. Based on Example 3-12, we have four virtual hosts devices on the Virtual I/O Server. These vhost devices are the ones we are going to map to our logical volumes. Adapter C4 is the adapter for the virtual DVD. See 2.2.1, “Virtual DVD-RAM, DVD-ROM, and CD-ROM” on page 25 for details on virtual optical devices.

Example 3-12 Create virtual device mappings

```
$ lsdev -vpd|grep vhost
  vhost4 U9113.550.105E9DE-V1-C50 Virtual SCSI Server Adapter
  vhost3 U9113.550.105E9DE-V1-C40 Virtual SCSI Server Adapter
  vhost2 U9113.550.105E9DE-V1-C30 Virtual SCSI Server Adapter
  vhost1 U9113.550.105E9DE-V1-C20 Virtual SCSI Server Adapter
  vhost0 U9113.550.105E9DE-V1-C4 Virtual SCSI Server Adapter

$ mkvdev -vdev rootvg_nim -vadapter vhost1 -dev vnim
vnim Available
$ mkvdev -vdev rootvg_dbsrv -vadapter vhost2 -dev vdbsrv
vdbsrv Available
$ mkvdev -vdev rootvg_apps -vadapter vhost3 -dev vapps
vapps Available
$ mkvdev -vdev rootvg_lnx -vadapter vhost4 -dev vlnx
vlnx Available
$ mkvdev -vdev cd0 -vadapter vhost0 -dev vcd
$ lsdev -virtual
name            status     description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vhost3          Available  Virtual SCSI Server Adapter
vhost4          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
vnim             Available  Virtual Target Device - Logical Volume
vapps           Available  Virtual Target Device - Logical Volume
vcd             Available  Virtual Target Device - Optical Media
vdbsrv          Available  Virtual Target Device - Logical Volume
vlnx            Available  Virtual Target Device - Logical Volume
ent3            Available  Shared Ethernet Adapter
```

Note: Based on the **lsdev -vpd** command, the mappings exactly correspond to the slot numbering we intended (refer to Figure 3-26 on page 146). For example, the vhost0 device is slot number 20 (U9111.520.10DDEEC-V1-C20) on the Virtual I/O Server, which is then being shared to the NIM_server partition. The NIM_server partition has its virtual SCSI device slot set to 20. This is for easy association between virtual SCSI devices on the server and client side.

4. Use the **lsmap** command to ensure that all logical connections between newly created devices are correct, as shown in Example 3-13.

Example 3-13 Checking mappings

```
$ lsmap -all
SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost0          U9113.550.105E9DE-V1-C4                     0x00000000

VTD                   vcd
LUN                   0x8100000000000000
Backing device        cd0
Physloc               U787B.001.DNW108F-P4-D2

SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost1          U9113.550.105E9DE-V1-C20                    0x00000000

VTD                   nim
LUN                   0x8100000000000000
Backing device        rootvg_nim
Physloc

SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost2          U9113.550.105E9DE-V1-C30                    0x00000000

VTD                   vdbsrv
LUN                   0x8100000000000000
```

```
Backing device        rootvg_dbsrv
Physloc

SVSA            Physloc                                    Client
Partition ID
--------------- -------------------------------------------
------------------
vhost3          U9113.550.105E9DE-V1-C40                   0x00000000

VTD                   vapps
LUN                   0x8100000000000000
Backing device        rootvg_apps
Physloc

SVSA            Physloc                                    Client
Partition ID
--------------- -------------------------------------------
------------------
vhost4          U9113.550.105E9DE-V1-C50                   0x00000000

VTD                   vlnx
LUN                   0x8100000000000000
Backing device        rootvg_lnx
Physloc
```

Tip: The same concept applies when creating virtual disks that are going to be used as data volume groups.

Using whole disks

SAN disks can be assigned to the Virtual I/O Server in sizes appropriate for the clients to be mapped as whole disks.

Note: When whole disks are going to be mapped, they cannot belong to a volume group.

These are the steps to map whole disks in the same way as in the previous section.

It is useful to be able to map the LUNs to hdisk numbers in a SAN environment. The **fget_config -Av** command is provided on the IBM DS4000™ series for a listing of LUN names, as shown in Example 3-14.

This command is part of the storage device driver and you will have to use the **oem_setup_env** command to access it. Use the same SCSI Server adapters as with the logical volumes.

Tip: A similar function is provided on DS8000™ and DS6000™ with the **lssdd** command.

Example 3-14 Listing of LUN to hdisk mapping

```
$ oem_setup_env
# fget_config -Av

---dar0---

User array name = 'FAST200'
dac0 ACTIVE dacNONE ACTIVE

Disk     DAC   LUN Logical Drive
hdisk4   dac0    0 vios1_rootvg
hdisk5   dac0    1 nim_rootvg
hdisk6   dac0    2 db_rootvg
hdisk7   dac0    3 apps_rootvg
hdisk8   dac0    4 linux_lvm
```

1. You can use the **lsdev -vpd** command to list the virtual slot numbers corresponding to vhost numbers as in Example 3-15

Example 3-15 Listing of slot number to vhost mapping

```
$ lsdev -vpd|grep vhost
  vhost4 U9113.550.105E9DE-V1-C50 Virtual SCSI Server Adapter
  vhost3 U9113.550.105E9DE-V1-C40 Virtual SCSI Server Adapter
  vhost2 U9113.550.105E9DE-V1-C30 Virtual SCSI Server Adapter
  vhost1 U9113.550.105E9DE-V1-C20 Virtual SCSI Server Adapter
  vhost0 U9113.550.105E9DE-V1-C4 Virtual SCSI Server Adapter
```

2. Define the SCSI mappings to create the virtual target devices that associates to the logical volume you have defined in the previous step. Based on Example 3-12 on page 165, we have four virtual hosts devices on the Virtual I/O Server. These vhost devices are the ones we are going to map to our disks. We also add the virtual DVD and call it vcd. See Example 3-16 on page 169.

Example 3-16 Mapping SAN disks and the DVD drive, cd0

```
$ mkvdev -vdev hdisk5 -vadapter vhost1 -dev vnim
vnim Available
$ mkvdev -vdev hdisk6 -vadapter vhost2 -dev vdbsrv
vdbsrv Available
$ mkvdev -vdev hdisk7 -vadapter vhost3 -dev vapps
vapps Available
$ mkvdev -vdev hdisk8 -vadapter vhost4 -dev vlnx
vlnx Available
$ mkvdev -vdev cd0 -vadapter vhost0 -dev vcd
vcd Available
```

Tip: The same concept applies when creating disks that are to be used as data disks.

Tip: You can map data disks through the same vhost adapters that are used for rootvg. VSCSI connections operate at memory speed and each adapter can handle a large number of target devices.

You are now ready to install AIX 5L V5.3 or Linux in each of the partitions. The disks should be available for client partitions to use at this point.

3.4.5 Client partition AIX 5L installation

This section describes the method to install AIX 5L Version 5.3 onto a previously defined client partition. You can choose your preferred method but for our basic scenario we opted to install the DB_server partition and Apps_server partition with the Network Installation Manager (NIM) that comes with AIX 5L Version 5.3. We are also going to use the virtual Ethernet adapters for network booting and the virtual SCSI disks that were previously allocated to client partitions for rootvg.

Tip: A virtual optical device can be used for a CD or DVD installation as long as it is not already assigned to a client partition.

Assuming that a NIM master is configured, the following are the basic steps required to perform an AIX 5L installation using NIM:

1. Create the NIM machine client dbserver and definitions on your NIM master. Example 3-17 shows you how to check for resources.

Example 3-17 Check if resources had been allocated

```
# lsnim -l dbserver
dbserver:
   class          = machines
   type           = standalone
   connect        = shell
   platform       = chrp
   netboot_kernel = mp
   if1            = ent-Network1 dbserver 0
   cable_type1    = N/A
   Cstate         = BOS installation has been enabled
   prev_state     = ready for a NIM operation
   Mstate         = not running
   boot           = boot
   lpp_source     = aix53-tl5
   mksysb         = lpar_base_aix53-tl5_ssh
   nim_script     = nim_script
   spot           = aix53_tl5
   control        = master
# tail /etc/bootptab
#      dt   -- old style boot switch
#      T170 -- (xstation only) -- server port number
#      T175 -- (xstation only) -- primary / secondary boot host
indicator
#      T176 -- (xstation only) -- enable tablet
#      T177 -- (xstation only) -- xstation 130 hard file usage
#      T178 -- (xstation only) -- enable XDMCP
#      T179 -- (xstation only) -- XDMCP host
#      T180 -- (xstation only) -- enable virtual screen
dbserver:bf=/tftpboot/dbserver:ip=9.3.5.113:ht=ethernet:sa=9.3.5.111:sm=255.255.255.0:
```

2. Initiate the install process by activating the DB_server client partition in SMS mode (see Figure 3-42).

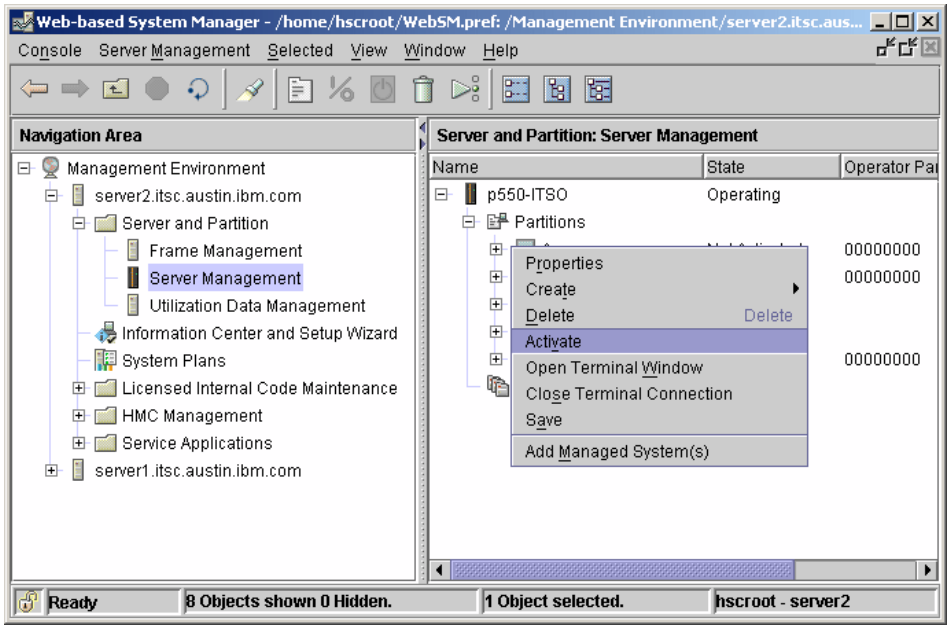


Figure 3-42 Activate DB_server partition

3. Set up the NIM master IP address and client address by choosing option 2 Setup Remote IPL (see Example 3-18).

Example 3-18 SMS Firmware Menu

```
PowerPC Firmware
 Version SF240_219
 SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------
--------
 Main Menu
 1.   Select Language
 2.   Setup Remote IPL (Initial Program Load)
 3.   Change SCSI Settings
 4.   Select Console
 5.   Select Boot Options
```

4. Choose option 1, as shown in Example 3-19.

Example 3-19 Available adapters for network booting

```
PowerPC Firmware
Version SF240_219
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------
-------
NIC Adapters
     Device                            Location Code
Hardware

Address
1.  Interpartition Logical LAN      U9113.550.105E9DE-V3-C2-T1
0a42a0003002
```

Note: Interpartition Logical LAN number 1 is the virtual Ethernet adapter that was defined on the NIM master for the DB_server client:

```
dbserver:
   class          = machines
   type           = standalone
   connect        = shell
   platform       = chrp
   netboot_kernel = mp
   if1            = ent-Network1 dbserver 0
```

5. Choose option 1 for IP Parameters, then go through each of the options and supply the IP address, as shown in Example 3-20.

Example 3-20 Specify IP address

```
PowerPC Firmware
 Version SF240_219
 SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------
--------
 IP Parameters
Interpartition Logical LAN: U9113.550.105E9DE-V3-C2-T1
 1.   Client IP Address                  [9.3.5.113]
 2.   Server IP Address                  [9.3.5.111]
 3.   Gateway IP Address                 [9.3.5.41]
 4.   Subnet Mask                       [255.255.255.000]
```

6. Press **Esc** to go one level up for the **Ping test** as shown in Example 3-21.

Example 3-21 Ping test

```
PowerPC Firmware
 Version SF240_219
 SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------
--------
 Network Parameters
Interpartition Logical LAN: U9113.550.105E9DE-V3-C2-T1
 1.   IP Parameters
 2.   Adapter Configuration
 3.   Ping Test
 4.   Advanced Setup: BOOTP
```

7. Execute a ping test and, provided it is successful, you are ready to do the NIM installation.
8. Go back to the main menu and go through booting the partition over the network.

3.4.6 Mirroring the Virtual I/O Server rootvg

Once the installation of the Virtual I/O Server is complete, the following commands can be used to mirror the VIOS rootvg volume group to a second physical volume. The following steps shows how to mirror the VIOS rootvg:

1. Use the **extendvg** command to include hdisk1 as part of the rootvg volume group. The same LVM concept applies; you cannot use an hdisk that belongs to another volume group and the disk needs to be of equal size or greater.
2. Use the **lspv** command, as shown in Example 3-22, to confirm that rootvg has been extended to include hdisk1.

Example 3-22 lspv command output

```
$ lspv
NAME            PVID                                VG
STATUS
hdisk0          00cddeec2dce312d                    rootvg
active
hdisk1          00cddeec87e69f91                    rootvg
active
hdisk2          00cddeec68220f19                    rootvg_clients
active
hdisk3          00cddeec685b3e88                    None
```

3. Use the **mirrorios** command to mirror the rootvg to hdisk1, as shown in Example 3-23. With the -f flag, the **mirrorios** command will automatically reboot the VIOS partition.

Note: SAN disks are usually RAID protected in the storage subsystem. If you use a SAN disk for the rootvg of the Virtual I/O Server, mirroring may not be required.

Example 3-23 Mirroring the Virtual I/O Server rootvg volumegroup

```
$ extendvg rootvg hdisk1
Changing the PVID in the ODM.

$ mirrorios -f hdisk1
SHUTDOWN PROGRAM
Fri Oct 13 17:32:33 CDT 2006
0513-044 The sshd Subsystem was requested to stop.

Wait for 'Rebooting...' before stopping.
```

4. Check if logical volumes are mirrored and if the normal boot sequence has been updated, as shown in Example 3-24.

Example 3-24 Logical partitions are mapped to two physical partitions

```
$ lsvg -lv rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     4     8     2    open/syncd    N/A
paging00            paging     8     16    2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/syncd    N/A
hd4                 jfs2       2     4     2    open/syncd    /
hd2                 jfs2       11    22    2    open/syncd    /usr
hd9var              jfs2       5     10    2    open/syncd    /var
hd3                 jfs2       9     18    2    open/syncd    /tmp
hd1                 jfs2       80    160   2    open/syncd    /home
hd10opt             jfs2       2     4     2    open/syncd    /opt
lg_dumplv           sysdump    8     8     1    open/syncd    N/A

$ bootlist -mode normal -ls
hdisk0 blv=hd5
hdisk1 blv=hd5
```

3.5 Interaction with UNIX client partitions

The following section describes how the Virtual I/O Server provides resources to the AIX 5L or Linux partitions. These resources can be storage and optical devices through virtual SCSI or network connectivity through virtual Ethernet.

At the time of this writing, i5/OS partitions on the System p5 servers do not interact with Virtual I/O Servers, so there is no mention about i5/OS client partitions.

3.5.1 Virtual SCSI services

For virtual SCSI, the interaction between a Virtual I/O Server and an AIX 5L or Linux client partition is enabled when the configuration of both virtual SCSI server adapter and virtual SCSI client adapter have matching slot numbers in their partition profiles, and both operating systems recognize their virtual adapter (with the **cfgmgr** command for dynamically added virtual SCSI adapters).

Once the interaction between virtual SCSI server and client adapters is enabled, mapping storage resources from the VIOS to the client partition is needed. The client partition configures and uses the storage resources when it starts up or when it is reconfigured at runtime.

The processes runs as follows:

- The HMC enables interaction between virtual SCSI adapters.
- The mapping of storage resources is performed in the VIOS.
- The client partition uses the storage after it has been booted or a **cfgmgr** command is run.

For more information about the technical details of virtual SCSI implementation, refer to 2.9, “Virtual SCSI introduction” on page 89.

Figure 3-43 shows the flow needed to enable virtual SCSI resources to AIX 5L or Linux clients. Note that Virtual I/O Server and client partitions do not need to restart when new virtual SCSI server and client adapters are created using the dynamic LPAR menus in the HMC and dynamic LPAR operations are enabled in the operating systems.

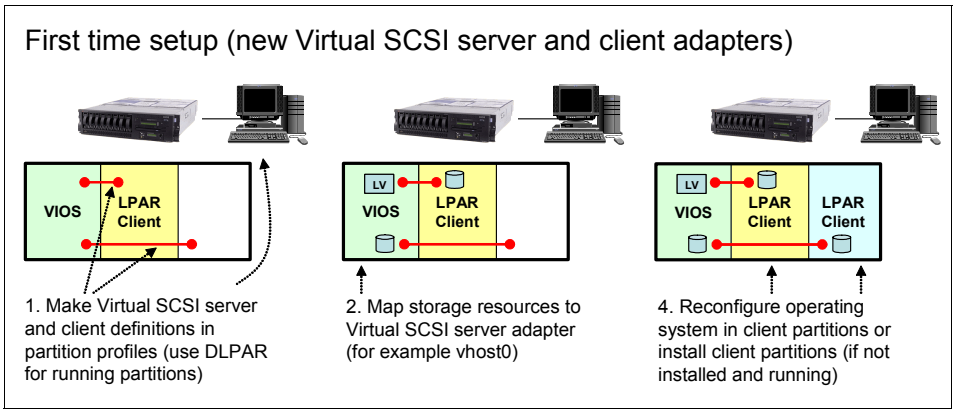


Figure 3-43 Basic configuration flow of virtual SCSI resources

To enable the AIX 5L or Linux client partitions to interact with virtual SCSI resources, the following steps are necessary:

1. Plan which virtual slot will be used in the Virtual I/O Server for the virtual SCSI server adapter and which slot in the AIX 5L or Linux client partition for the virtual SCSI client adapter (each partition has its own pool of virtual slots). In the VIOS, consider ranges of slot numbers for virtual adapters that serve a specific partition (for example, slots 20 through 29 for virtual SCSI server adapters for an AIX 5L client partition).

2. Define the virtual SCSI server adapter on the Virtual I/O Server.
3. Define the SCSI client adapter on the AIX 5L or Linux client partition.
4. Map the desired SCSI resources using the **mkvdev** command, as described in 3.4.4, “Defining virtual disks” on page 163.

Once both the Virtual I/O Server and AIX 5L or Linux client partitions are interacting with each other, virtual devices can be mapped online on the Virtual I/O Server and the AIX 5L or Linux client partitions can be reconfigured online to use them, as shown in the Figure 3-44.

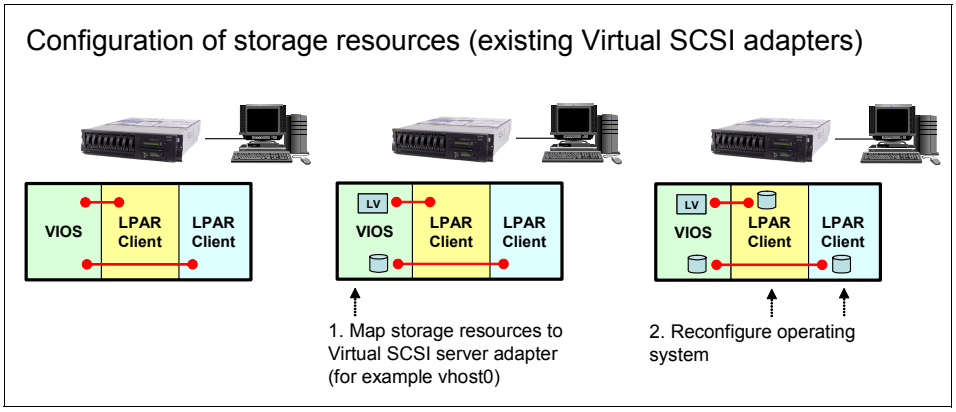


Figure 3-44 Basic configuration flow of virtual SCSI resources

Figure 3-45 shows the steps to create new virtual SCSI server and client adapters and define the initial storage resources for an AIX 5L client partition.

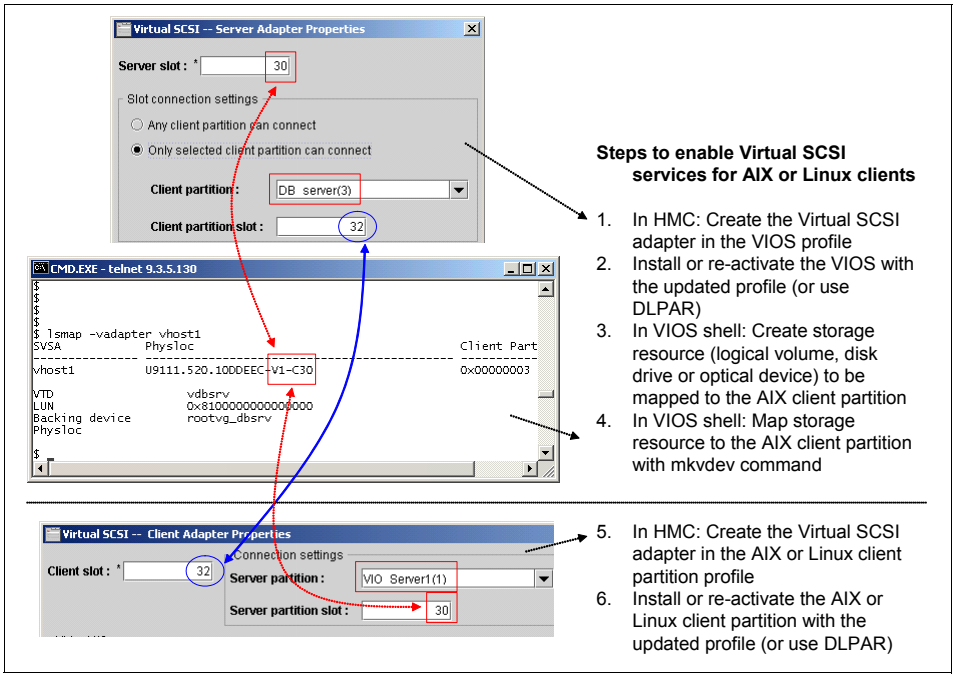


Figure 3-45 Steps to enable virtual SCSI service to an AIX 5L client partition

Virtual SCSI services for AIX 5L partitions

AIX 5L client partitions can use virtual SCSI devices currently mapped from the Virtual I/O Server once the operating system starts up or the system is reconfigured with the **cfgmgr** or **mkdev** commands.

Mapped disk-type resources (physical volumes or logical volumes in the VIOS scope) appear on the AIX 5L client partition as an hdisk type of device (for example, hdisk0, hdisk1). The virtual SCSI client adapter can use these devices like any other physically connected hdisk device for boot, swap, mirror, or any other supported AIX 5L feature. Optical-type resources (DVD-ROM and DVD-RAM) appear as a CD type of device (for example, cd0).

Virtual SCSI services for Linux partitions

The configuration for Linux partitions to allow them to use virtual SCSI adapters and devices is similar to the configuration for AIX 5L partitions. Linux partitions handle virtual SCSI devices as SCSI disc drives. Clients should be aware about the following considerations in the Linux client partitions:

- Virtual SCSI adapters in the Linux client partition are listed in the /sys/class/scsi_host directory; such a directory contains the control files for each of the virtual adapters. The virtual SCSI client adapters can be requested to re-scan for virtual SCSI resources recently mapped from the VIOS using the same interfaces as any other SCSI adapter.
- Virtual SCSI disk drives in the Linux client partition can be seen in the device tree as an sd*x* type of device (for example, sda). Note that Linux shows in the device tree both virtual disk drives (virtual SCSI storage resources mapped from VIOS) and disk partitions created inside those virtual disk drives.

3.5.2 Virtual Ethernet resources

The second type of resource that causes interaction between AIX 5L or Linux client partitions and the Virtual I/O Server is the Shared Ethernet Adapter in the Virtual I/O Server. This feature allows the AIX 5L partitions to connect to external networks without a physical adapter.

The implementation of virtual Ethernet adapters is based on the definition of network interfaces that connect through the POWER Hypervisor to an IEEE VLAN-aware virtual Ethernet switch in the system. All partitions talking on the virtual Ethernet network are peers. Up to 4,096 separate IEEE VLANs can be defined in the system. Each partition can have up to 65,533 virtual Ethernet adapters connected to the virtual Ethernet switch and each adapter can be connected to 21 different IEEE VLANs (20 VID and 1 PVID).

For more information about the technical details of virtual Ethernet implementation, refer to 2.8, “Virtual and Shared Ethernet introduction” on page 70.

The enablement and setup of a virtual Ethernet adapter in an AIX 5L or Linux client partition does not require any special hardware or software. After a specific virtual Ethernet is enabled for a partition, a network device is created within the partition. The user can then set up the TCP/IP configuration appropriately to communicate with other partitions.

In order to allow the AIX 5L or Linux client partition to communicate with external Ethernet networks, the Virtual I/O Server acts like a bridge and forwards the IP packets from the virtual Ethernet client adapter. This is done by creating a Shared Ethernet Adapter (SEA) in the Virtual I/O Server that connects the virtual

Ethernet server adapter and a physical Ethernet adapter in the server. Figure 3-46 shows an example of the steps to configure virtual Ethernet services and details about a Shared Ethernet Adapter in a VIOS.

For more information about the Shared Ethernet Adapter, refer to 2.8, “Virtual and Shared Ethernet introduction” on page 70.

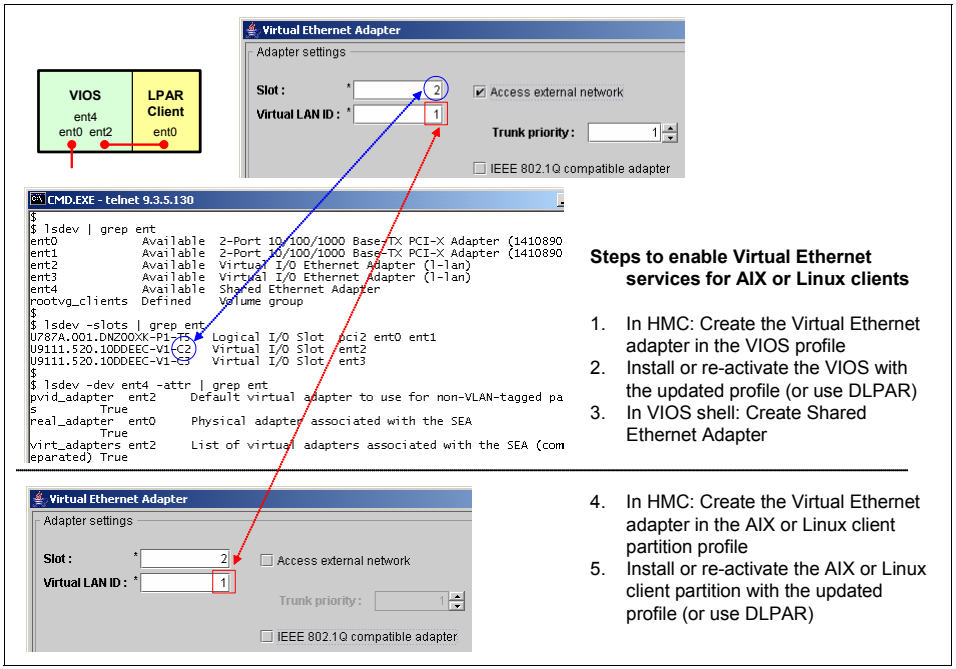


Figure 3-46 Steps required to enable virtual Ethernet connectivity

Note: Remember that virtual SCSI or virtual Ethernet adapters created with dynamic LPAR operations are not reflected in the partition’s profiles and exist at partition runtime only. You can either update your profile or save your current configuration to a new profile to make your changes permanent across restarts.

4

Setting up virtual I/O: advanced

This chapter starts with a discussion of advanced Virtual I/O Server topics, which are important to understand before you start to set up configurations, including:

- Providing higher serviceability for the Virtual I/O Server.

  This includes a discussion of when to use multiple Virtual I/O Servers and how higher availability for communication with external networks can be achieved. We apply these concepts in three scenarios and demonstrate how to set up advanced configurations that provide increased levels of redundancy:

  - Scenario 1: Logical Volume Mirroring
  - Scenario 2: SEA Failover
  - Scenario 3: MPIO in the client with SAN
  - Scenario 4: Network Interface Backup in the client
- We also show how to activate a partition before initiating a Linux installation in a VIO client.
- Finally, we list supported configurations, which concludes a section with some advice on the use of IBM TotalStorage® Solutions, HACMP, and GPFS in a virtualized environment.

4.1 Providing higher serviceability

This section provides a discussion around the requirements and concepts for increasing the availability of Virtual I/O Servers and when to use multiple Virtual I/O Servers for providing virtual SCSI and shared Ethernet services to client partitions.

Several considerations are taken into account when deciding to use multiple Virtual I/O Servers. Client uptime and predicted I/O load averages for both network and storage and system manageability are areas that need consideration when planning to build Virtual I/O Servers.

For a small system with limited resources and limited I/O adapters, a second Virtual I/O Server may not have the required resources. With limited I/O adapters, having a second Virtual I/O Server may adversely affect the overall performance supplied to the clients when the additional adapters are used for increasing redundancy, not throughput.

For a larger system, there is a lower resource constraint and multiple Virtual I/O Server may be deployed without affecting overall client performance. More I/O adapters cater for both throughput and redundancy when used with additional Virtual I/O Servers.

The Virtual I/O Server is very robust since it mainly runs device drivers, does not run any application workloads, and regular users are not logged on. Two Virtual I/O Severs are often implemented to provide higher serviceability. One Virtual I/O Server can be rebooted or even reinstalled without affecting the virtual I/O clients.

Note: IVM supports a single Virtual I/O Server.

4.1.1 Providing higher serviceability with multiple Virtual I/O Servers

While redundancy can be built into the Virtual I/O Server itself with the use of MPIO and LVM mirroring (RAID) for storage devices and Link Aggregation or Shared Ethernet Failover for network devices, the Virtual I/O Server must be available with respect to the client. Planned outages, such as software updates (see 5.4.1, “Concurrent software updates for the VIOS” on page 294), and unplanned outages, such as hardware outages, challenge 24x7 availability.

Having multiple Virtual I/O Servers provide access to the same resources achieves good redundancy on the client with availability similar to having redundant adapters connected directly into the client.

Virtual SCSI redundancy

With the availability of MPIO on the client, each Virtual I/O Server can present a virtual SCSI device that is physically connected to the same physical disk. This achieves redundancy for the Virtual I/O Server itself and for any adapter, switch, or device that is used between the Virtual I/O Server and the disk.

With the use of logical volume mirroring on the client, each Virtual I/O Server can present a virtual SCSI device that is physically connected to a different disk and then used in a normal AIX 5L mirrored volume group on the client. This achieves a potentially higher level of reliability by providing redundancy. Client volume group mirroring is also required when a Virtual I/O Server logical volume is used as a virtual SCSI device on the Client. In this case, the virtual SCSI devices are associated with different SCSI disks, each controlled by one of the two Virtual I/O Server.

Figure 4-1 on page 184 displays an advanced setup using both MPIO and LVM mirroring in the VIO client at the same time: two Virtual I/O Servers host disks for a single client. The client is using MPIO to access a SAN disk and LVM mirroring to access two SCSI disks. From the client perspective, the following situations could be handled without causing downtime for the client:

- Either path to the SAN disk could fail, but the client would still be able to access the data on the SAN disk through the other path. No action has to be taken to reintegrate the failed path to the SAN disk after repair.
- The failure of a SCSI disk would cause stale partitions for the volume group with the assigned virtual disks, but the client still would be able to access the data on the other copy of these mirrored disk. After the repair of the failed SCSI disk or reboot of the Virtual I/O Server, all stale partitions would have to be synchronized using the **varyonvg** command.
- Either Virtual I/O Server could be rebooted, which would result in a temporary simultaneous failure of one path to the SAN disk and stale partitions for the volume group on the SCSI disks, as described before.

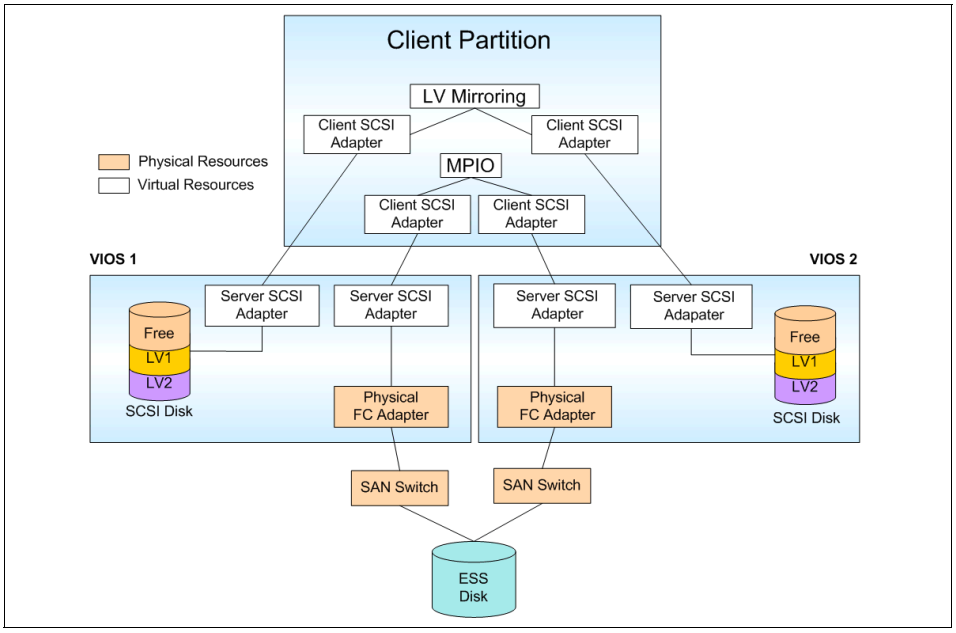


Figure 4-1 MPIO and LV Mirroring with two Virtual I/O Server

Note: If mirroring and MPIO are both configurable in your setup, MPIO should be the preferred method for adding disk redundancy to the client. LV mirroring causes stale partitions that requires synchronization, while MPIO does not.

Note: We recommend that you use two Fibre Channel adapters in each Virtual I/O Server for adapter redundancy.

For further examples of virtual SCSI configurations, see 4.7.1, “Supported VSCSI configurations” on page 238.

Shared Ethernet redundancy

Having link aggregation configured on the Virtual I/O Server protects the Virtual I/O Server from adapter and network switch failures, but this does not remove the dependency between the client and the server. The client partition still requires the layer of abstraction from the Virtual I/O Server that can be achieved by using Network Interface Backup, IP multipathing with Dead Gateway Detection, or Shared Ethernet Adapter (SEA) Failover, which are explained later.

Shared Ethernet Adapter Failover was delivered with Virtual I/O Server V1.2 and offers Ethernet redundancy to the client at the virtual level. The client gets one standard virtual Ethernet adapter hosted by two Virtual I/O Server. The two Virtual I/O Servers use a control channel to determine which of them is supplying the Ethernet service to the client. Through this active monitoring between the two Virtual I/O Servers, failure of either will result in the remaining Virtual I/O Server taking control of the Ethernet service for the client. The client has no special protocol or software configured and uses the virtual Ethernet adapter as though it was hosted by only one Virtual I/O Server.

In Figure 4-2, a typical setup combining Link Aggregation and SEA Failover to increase availability is shown: two Virtual I/O Servers have Link Aggregation configured over two Ethernet cards to provide better bandwidth and redundancy. A Shared Ethernet Adapter has been configured. The control channel for the Shared Ethernet Adapter is completely separate. When the client partition starts, its network traffic will be serviced by Virtual I/O Server 1, as it has the higher priority assigned. If Virtual I/O Server 1 is unavailable, Virtual I/O Server 2 will determine this using the control channel and take control of the network service for the client partition.

Note: Note that a numerically lower priority receives a higher overall priority.

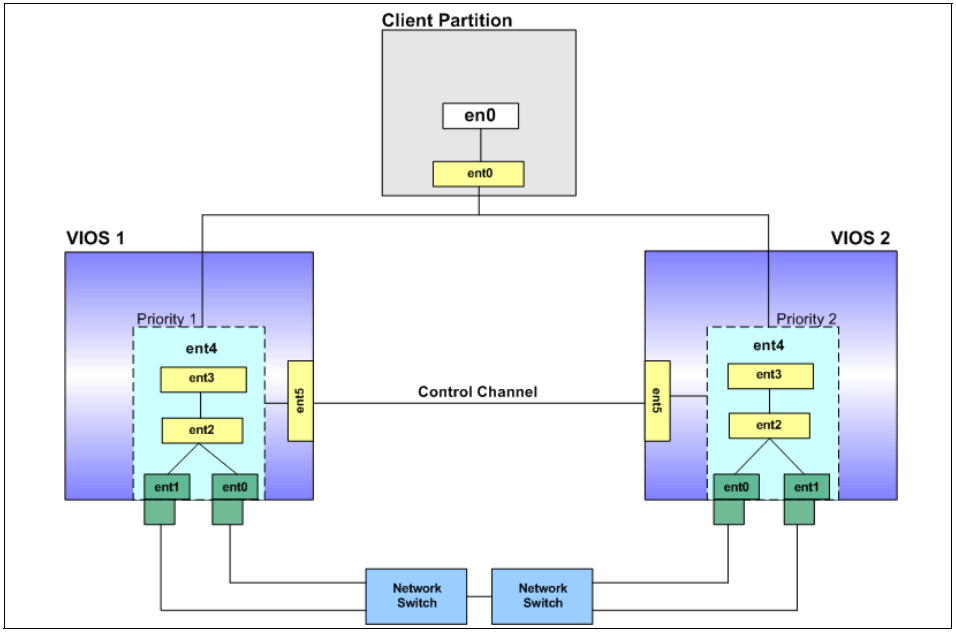


Figure 4-2 Shared Ethernet Adapter failover

Some advanced topics with respect to networking high-availability and performance are now discussed in the next sections in detail:

- Performance and availability of a Shared Ethernet Adapter can be improved by using Link Aggregation or EtherChannel.
- Approaches to provide redundancy for a Shared Ethernet Adapter for access to external networks are discussed.
- POWER Hypervisor implementation backgrounds and performance implications are discussed.

Summaries of the benefits and considerations conclude this discussion of advanced Virtual and Shared Ethernet concepts.

Network Interface Backup for virtual I/O client redundancy

Network Interface Backup (NIB) can be used in the virtual I/O client to achieve network redundancy when using two Virtual I/O Severs. The client uses two virtual Ethernet adapters to create an EtherChannel that consists of one primary adapter and one backup adapter. The interface is defined on the EtherChannel. If the primary adapter becomes unavailable, the Network Interface Backup switches to the backup adapter. See 4.5, “Scenario 4: Network Interface Backup in the client” on page 234for setup details. Figure 4-3 shows the concept.

,

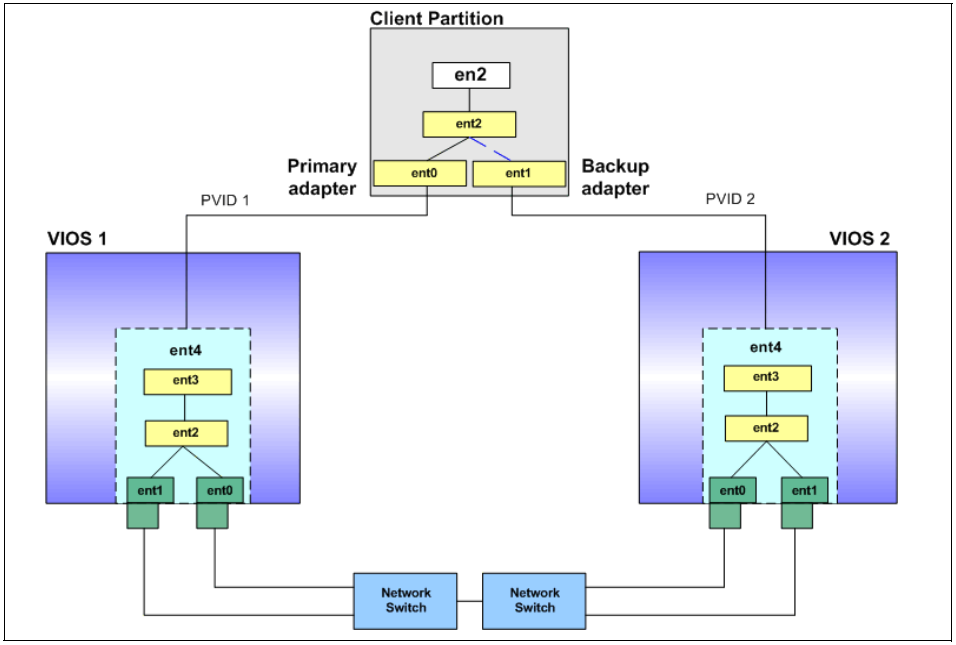


Figure 4-3 Network redundancy using two VIOS and Network Interface Backup

4.1.2 Using Link Aggregation or EtherChannel to external networks

Link Aggregation is a network port aggregation technology that allows several Ethernet adapters to be aggregated together to form a single pseudo Ethernet adapter. This technology is often used to overcome the bandwidth restriction of a single network adapter and avoid bottlenecks when sharing one network adapter among many client partitions.

The main benefit of a Link Aggregation is that it has the network bandwidth of all of its adapters in a single network presence. If an adapter fails, the packets are automatically sent on the next available adapter without disruption to existing user connections. The adapter is automatically returned to service on the Link Aggregation when it recovers. Thus, Link Aggregation also provides some degree of increased availability. A link or adapter failure will lead to a performance degradation, but not a disruption.

But Link Aggregation is not a complete high-availability networking solution because all the aggregated links must connect to the same switch. This restriction can be overcome with the use of a backup adapter: You can add a single additional link to the Link Aggregation, which is connected to a different Ethernet switch with the same VLAN. This single link will only be used as a backup.

As an example for Link Aggregation, ent0 and ent1 can be aggregated to ent2. The system considers these aggregated adapters as one adapter. Interface en2 would then be configured with an IP address. Therefore, IP is configured as on any other Ethernet adapter. In addition, all adapters in the Link Aggregation are given the same hardware (MAC) address, so they are treated by remote systems as though they were one adapter.

There are two variants of Link Aggregation supported in AIX 5L:

- Cisco EtherChannel (EC)
- IEEE 802.3ad Link Aggregation (LA)

While EC is an Cisco-specific implementation of adapter aggregation, LA follows the IEEE 802.3ad standard. Table 4-1 shows the main differences between EC and LA.

Table 4-1 Main differences between EC and LA aggregation

Cisco EtherChannel	IEEE 802.3ad Link Aggregation
Cisco-specific	Open Standard.
Requires switch configuration	Little, if any, configuration of switch required to form aggregation. Some initial setup of the switch may be required.
Supports different packet distribution modes	Supports only standard distribution mode.

The main benefit of using LA is, that if the switch supports the *Link Aggregation Control Protocol* (LACP), no special configuration of the switch ports is required. The benefit of EC is the support of different packet distribution modes. This means it is possible to influence the load balancing of the aggregated adapters. In the remainder of this redbook, we use Link Aggregation where possible since that is considered a more universally understood term.

Figure 4-4 on page 189 shows the aggregation of four plus one adapters to a single pseudo-Ethernet device, including a backup feature. The Ethernet adapters ent0 to ent3 are aggregated for bandwidth and must be connected to the same Ethernet switch, while ent4 connects to a different switch, but is only used for backup, for example, if the main Ethernet switch fails. The adapters ent0 through ent4 are now exclusively accessible through the pseudo Ethernet adapter ent5 and its interface en5. You could not, for example, attach a network interface en0 to ent0, as long as ent0 is a member of an EtherChannel or Link Aggregation.

Note: A Link Aggregation or EtherChannel of virtual Ethernet adapters is not supported. But you may use the Network Interface Backup feature of Link Aggregation with virtual Ethernet adapters.

A Link Aggregation with only one primary Ethernet adapter and one backup adapter is said to be operating in Network Interface Backup (NIB).

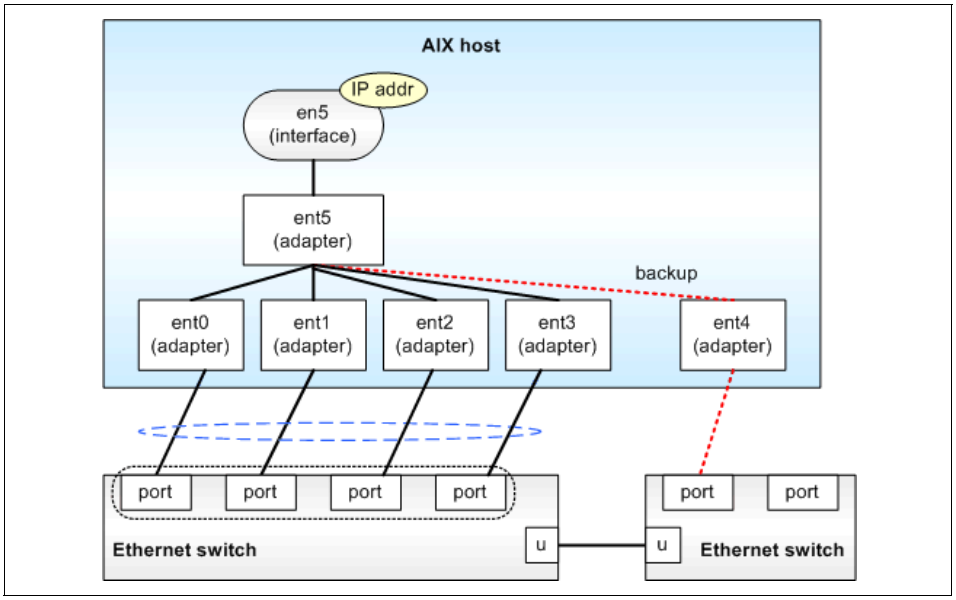


Figure 4-4 Link Aggregation (EtherChannel) on AIX 5L

4.1.3 High availability for communication with external networks

In a single Virtual I/O Server configuration, communication to external networks ceases if the Virtual I/O Server is unavailable. VIO clients will experience this disruption if it uses the SEA as a means to access the external networks. Communication through the SEA is suspended as soon as the Virtual I/O Server goes down, and communication resumes when the Virtual I/O Server comes up again. Internal communication between partitions through virtual Ethernet connections can continue unaffected while the Virtual I/O Server is unavailable. VIO clients do not have to be rebooted or otherwise reconfigured to resume communication through the SEA. Fundamentally, with respect to virtual Ethernet, the reboot of an Virtual I/O Server with a SEA affects the clients similar to unplugging and replugging of an uplink of a physical Ethernet switch.

If the temporary failure of communication with external networks is unacceptable, more than a single forwarding instance and some function for failover has to be implemented.

Consideration: The Integrated Virtualization Manager (IVM) supports a single Virtual I/O Server, and all physical devices are owned by this Virtual I/O Server. This section only applies to systems managed by the Hardware Management Console (HMC).

Alternatives for network high availability in AIX 5L

With physical Ethernet adapters, the following concepts could be used in AIX 5L for high-availability of network communications:

- On layer-2 (Ethernet), as shown in Figure 4-5 on page 191, with the backup adapter feature of EtherChannel or Link Aggregation or Network Interface Backup (NIB).
- On layer-3 (TCP/IP), as shown in Figure 4-6 on page 192, with the IP Multipathing (IPMP) feature and one of the following:
  - With Dead Gateway Detection (DGD)
  - With Virtual IP Addresses (VIPA) and dynamic routing protocols, such as Open Shortest Path First (OSPF)
- Local IP Address Takeover (IPAT), with High Availability Cluster Management or Automation Software, such as HACMP for AIX 5L or Tivoli System Automation (TSA).

The following are some important considerations for the use of Link Aggregation with virtual, instead of physical, Ethernet adapters.

Consideration: A Link Aggregation of more than one active virtual Ethernet adapter is not supported. Only one primary virtual Ethernet adapter plus one backup virtual Ethernet adapter are supported. Thus, a total of exactly two virtual adapters, one active and one standby, as shown in Figure 4-5 on page 191, is allowed.

Important: When using NIB with virtual Ethernet adapters, it is mandatory to use the ping-to-address feature to be able to detect network failures, because there is no hardware link failure for virtual Ethernet adapters to trigger a failover to the other adapter.

You can have multiple active physical Ethernet adapters in a Link Aggregation plus one backup virtual Ethernet adapter, however.

Consideration: When configuring NIB with two virtual Ethernet adapters, the two internal networks used must stay separated in the POWER Hypervisor, so you have to use two different PVIDs for the two adapters in the client and cannot use additional VIDs on them. The two different internal VLANs are then bridged to the same external VLAN.

IP Multipathing with Dead Gateway Detection can only make outbound connections highly available, while the use of VIPA with dynamic routing

protocols can make outbound and inbound connections highly available. The implementation of dynamic routing protocols may be quite complex, and network equipment must also be capable of participating.

Note: VIPA and OSPF are used with IBM System z Geographically Dispersed Parallel Sysplex™ (GDPS®). Thus, if you are implementing highly available shared Ethernets on System p5 in a System z9 environment, the external network devices may possibly already be configured to operate with OSPF, so you might consider this option.

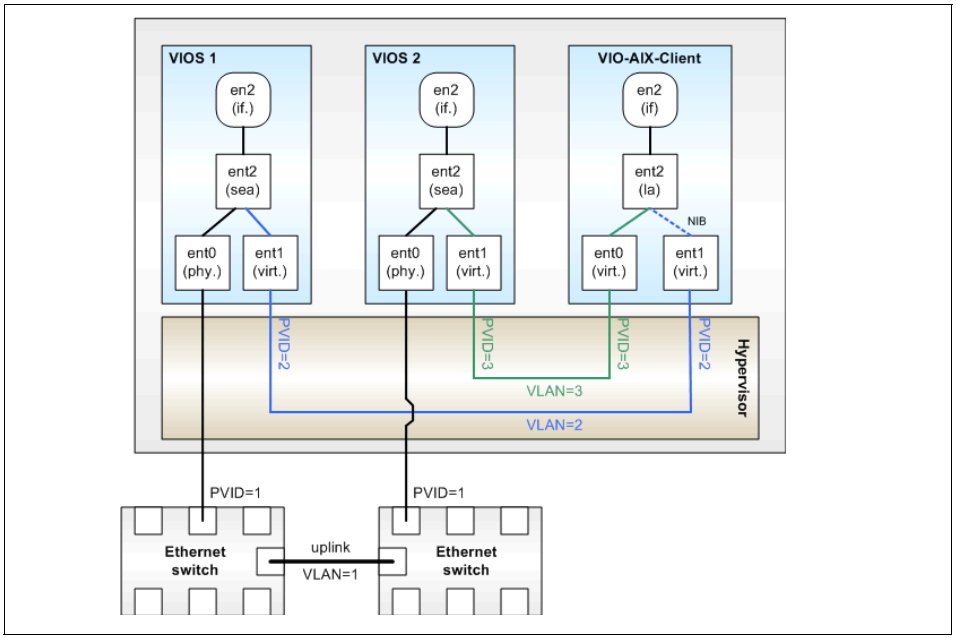


Figure 4-5 Network Interface Backup with two Virtual I/O Server

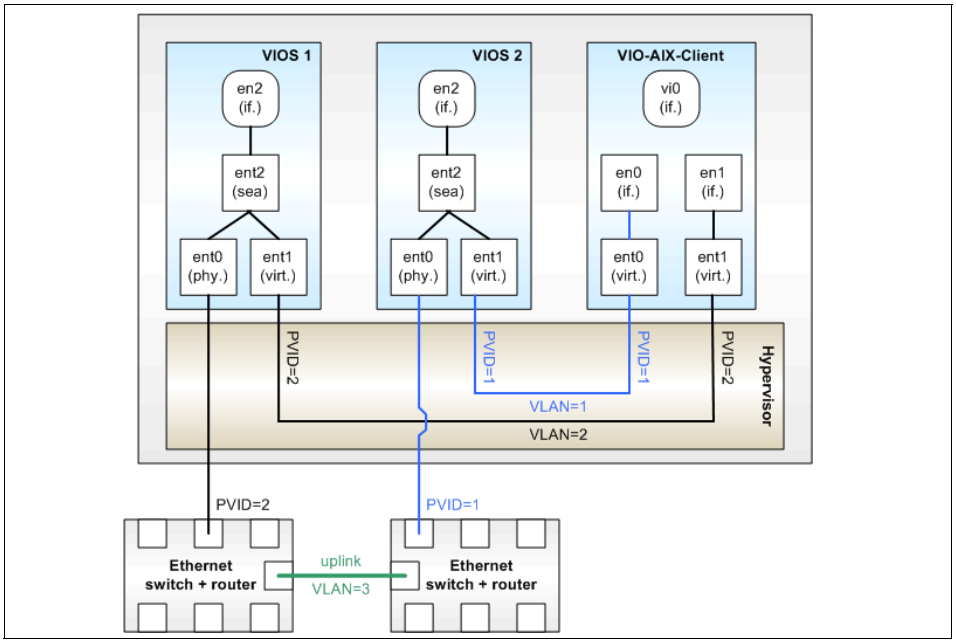


Figure 4-6 IP multipathing in the client using two SEA of different VIOS

Implement network HA in the client or server

The approaches just described, such as NIB, IPMP, and IPAT, are provided by AIX 5L and could be used by a shared Ethernet client partition to achieve high availability for communication with external networks. These approaches have in common that multiple virtual Ethernet adapters are required in the client partitions, and the failover logic is implemented in the client partitions, which makes configuration of the client partitions more complex.

The next two sections will describe two approaches to high availability for access to external networks (router failover and Shared Ethernet Adapter Failover) that have in common that no failover logic has to be implemented in the client partitions, thus simplifying client configuration.

Router failover

When routing (layer-3 forwarding) instead of bridging (layer-2 forwarding) is used to connect inter-partition networks to external networks, two router partitions could be used and IP Address Takeover (IPAT) configured between these routing partitions to provide high availability for access to external networks. The client partitions would then use a highly-available IP address as their default route, which simplifies configuration for these partitions and concentrates the

complexity of failover logic in the routing partitions. This approach is shown in Figure 4-7.

To implement IPAT in the router partitions, additional software would be required, such as HACMP for AIX 5L, Tivoli System Automation for AIX 5L or Linux, Heartbeat for Linux, or an implementation of the Virtual Router Redundancy Protocol (VRRP) on Linux, for example, similar to the approach described in *Linux on IBM eServer zSeries and S/390: Virtual Router Redundancy Protocol on VM Guest LANs*, REDP-3657.

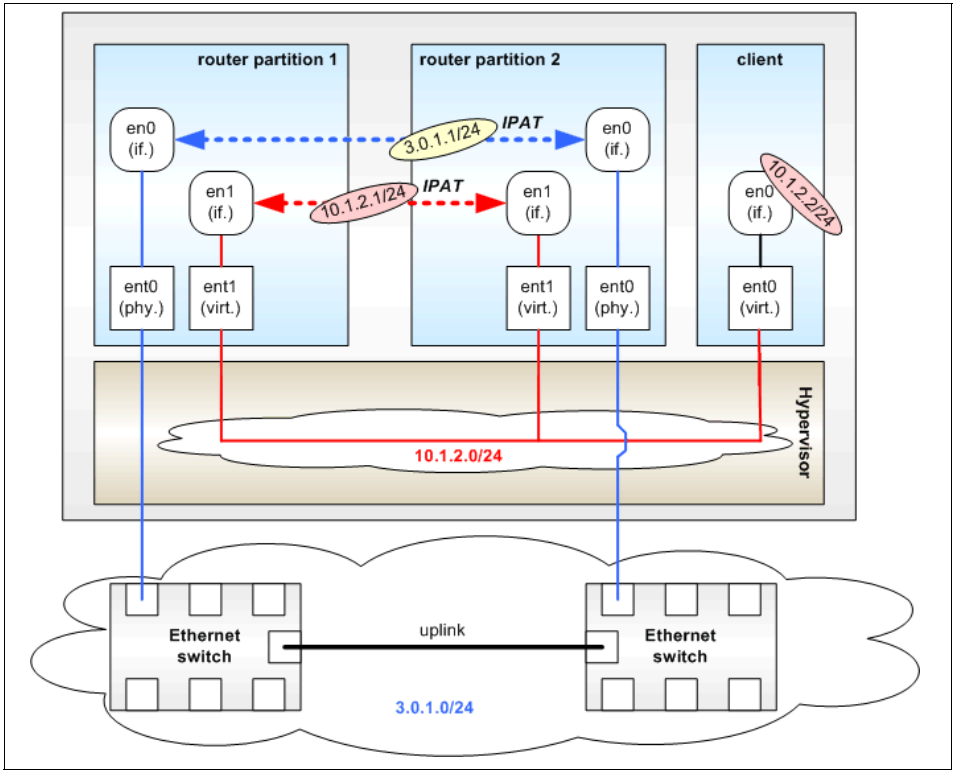


Figure 4-7 Router failover

Shared Ethernet Adapter Failover

Starting with Virtual I/O Server V1.2, there is a very straightforward solution for Shared Ethernet high availability, *Shared Ethernet Adapter Failover* (SEA Failover). SEA Failover is implemented on the Virtual I/O Server and not on the client, and it supports the simpler bridging (layer-2) approach to access external networks. SEA Failover supports IEEE 802.1Q VLAN-tagging, just like the basic SEA feature.

SEA Failover works as follows: Two Virtual I/O Servers have the bridging functionality of the Shared Ethernet Adapter to automatically fail over, if one Virtual I/O Server fails, shuts down, or the SEA looses access to the external network through its physical Ethernet adapter. You can also trigger a manual failover.

As shown in Figure 4-8, both Virtual I/O Servers attach to the same virtual and physical Ethernet networks and VLANs, and both virtual Ethernet adapters of both SEAs will have the *access the external network flag* enabled, which was called the *trunk flag* in earlier releases. An additional virtual Ethernet connection has to be set up as a separate VLAN between the two Virtual I/O Servers and must be attached to the Shared Ethernet Adapter (SEA) as a *control channel*, not as regular member of the SEA. This VLAN serves as a channel for the exchange of keep-alive or heartbeat messages between the two Virtual I/O Servers and therefore controls the failover of the bridging functionality. No network interfaces have to be attached to the control channel Ethernet adapters; the control channel adapter should be dedicated and on a dedicated VLAN that is not used for anything else.

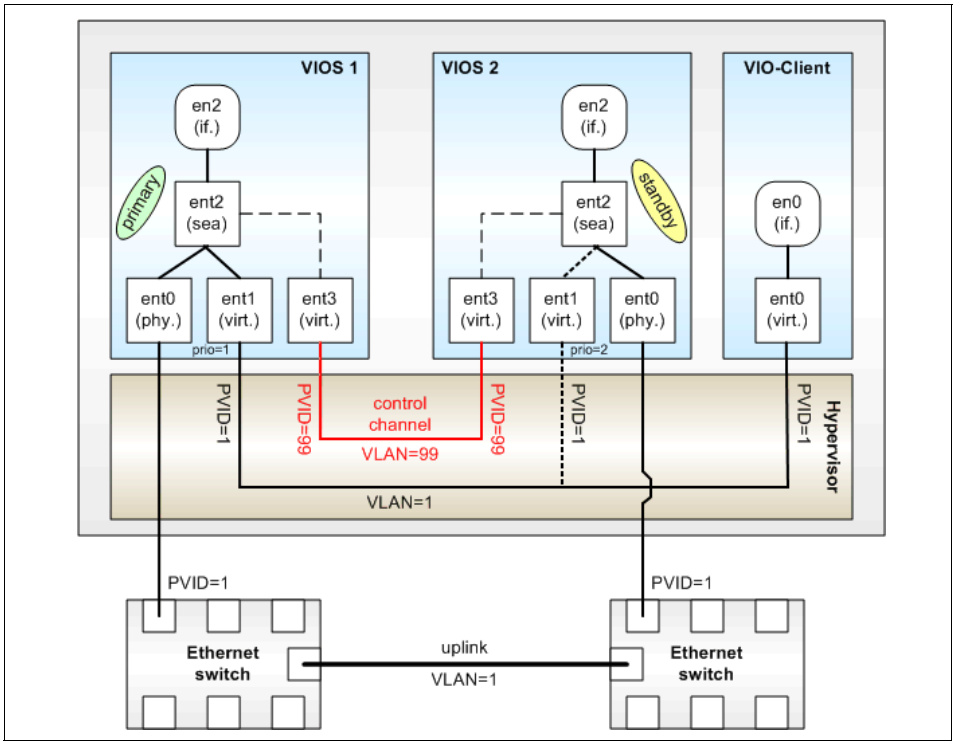


Figure 4-8 Basic SEA Failover configuration

You must select different priorities for the two SEAs by setting all virtual Ethernet adapters of each SEA to that priority value. The priority value defines which of the two SEAs will be the primary (active) and which will be the backup (standby). The lower the *priority value*, the higher the priority, thus priority=1 means highest priority.

The SEA can also be configured with an IP address that it will periodically try to ping to confirm network connectivity is available. This is similar to the IP address to ping that may be configured with Network Interface Backup.

There are basically four different cases that will initiate a SEA Failover:

1. The standby SEA detects that keep-alive messages from the active SEA are no longer received over the control channel.
2. The active SEA detects that a loss of the physical link is reported by the physical Ethernet adapter's device driver.
3. On the Virtual I/O Server with the active SEA, a manual failover can be initiated by setting the active SEA to standby mode.
4. The active SEA detects that it cannot ping a given IP address anymore.

An end of the keep-alive messages would occur when the Virtual I/O Server with the primary SEA is shut down or halted, has stopped responding, or has been deactivated from the HMC.

There may also be some types of network failures that would not trigger a failover of the SEA, because keep-alive messages are only sent over the control channel. No keep-alive messages are sent over other SEA networks, especially not over the external network. But the SEA Failover feature can be configured to periodically check the reachability of a given IP address. The SEA will periodically ping this IP address, so it can detect some other network failures. You may already know this feature from AIX 5L Network Interface Backup (NIB).

Important: The Shared Ethernet Adapters must have network interfaces with IP addresses associated to be able to use this periodic reachability test.

These IP addresses have to be unique and you have to use different IP addresses on both SEAs.

The SEAs must have IP addresses to provide as return-to-address for the ICMP-Echo-Requests sent and ICMP-Echo-Replies received when pinging the given IP address. These IP addresses must be different.

Although the alternative configuration for SEA Failover in Figure 4-9 would be nearly equivalent to the one in Figure 4-8 on page 194, you would not be able to make the SEAs periodically ping a given IP address. We recommend associating the interface and IP address to the SEA, as shown in Figure 4-8 on page 194.

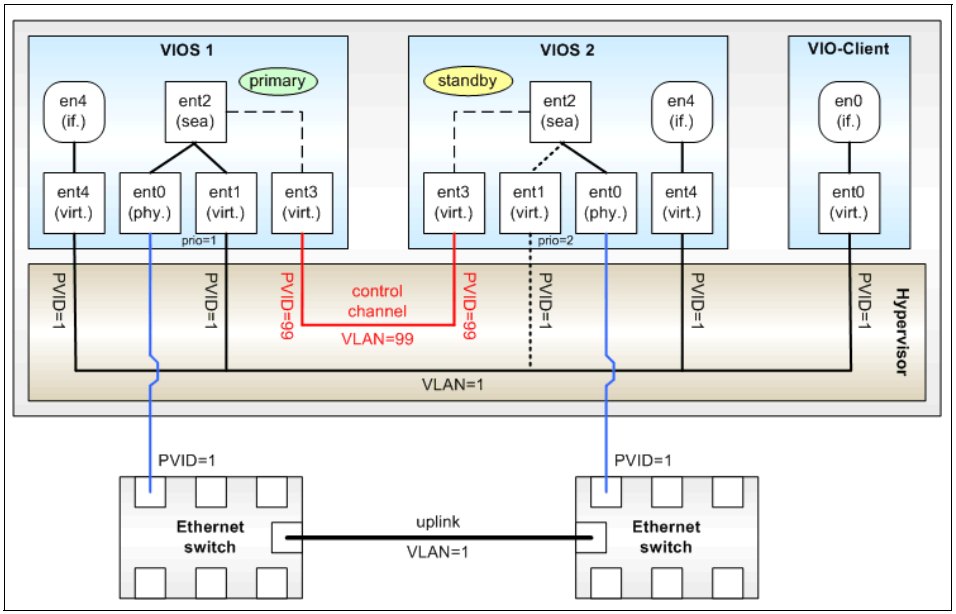


Figure 4-9 Alternative configuration for SEA Failover

The advantage of SEA Failover when compared to Network Interface Backup, besides allowing VLAN tagging, is the simplification of the Virtual I/O client's configurations. This is because there is only a single Ethernet adapter, a single default route, and no logic for failover in the Virtual I/O client. All redundancy and HA-related functions are implemented in the Virtual I/O Server. This demonstrates that virtualization has the potential for more flexible deployment and better resource utilization, but also for simplification and separation of concerns.

The next section discusses this architectural consideration in more depth.

When to use SEA Failover or Network Interface Backup

In the previous sections, we described two different approaches on how to provide high-availability for bridged (layer-2) access to external networks with redundant Shared Ethernet Adapters for two Virtual I/O Servers:

- Network Interface Backup (NIB) (see Figure 4-5 on page 191 and Figure 4-10)
- Shared Ethernet Adapter Failover (see Figure 4-8 on page 194)

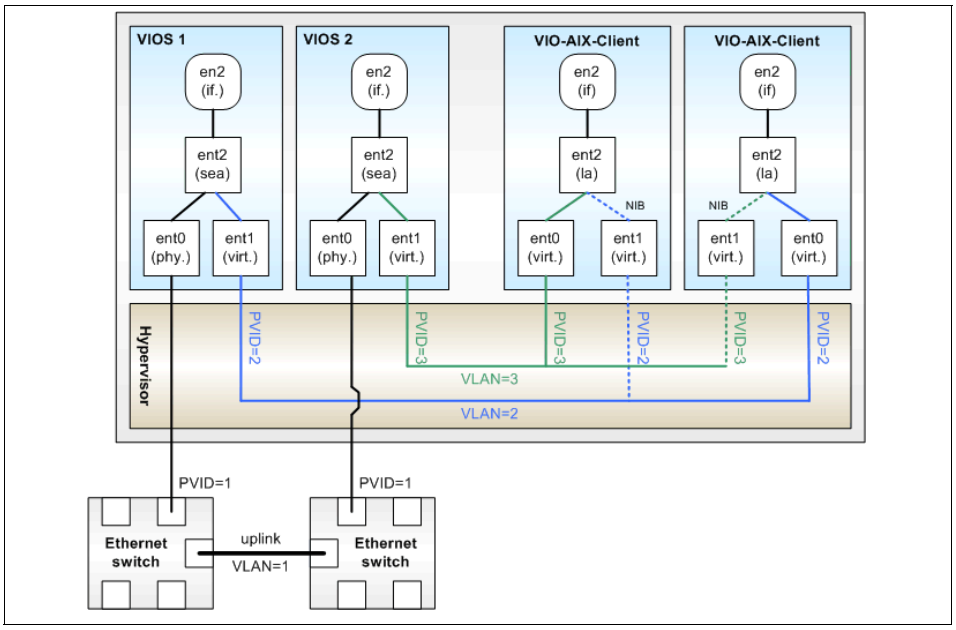


Figure 4-10 Network Interface Backup with multiple clients

In the following, we compare these two approaches:

- SEA Failover is implemented on the Virtual I/O Server, while NIB is implemented on the client.
- SEA Failover simplifies the configuration of multiple clients.

  SEA Failover makes the configuration of clients easier, because they just need to have a single adapter and VLAN and no failover logic implemented.

  When using the NIB approach, all clients have to configure a second virtual Ethernet adapter on a different VLAN and a Link Aggregation adapter with the NIB feature. This makes client configuration more complex.
- SEA Failover can be used with IEEE802.1Q VLAN-tagging, while NIB cannot.

With NIB, the two internal networks used must stay separated in the POWER Hypervisor, so you have to use two different PVIDs for the two adapters in the client and cannot use additional VIDs on them. The two different internal VLANs are then bridged to the same external VLAN.

- NIB can provide better resource utilization:
  - With SEA Failover, only one of the two SEAs is actively used at any time, while the other SEA is only a standby. Thus, the bandwidth of the physical Ethernet adapter of the standby SEA adapter is not used.
  - With NIB, you may distribute the clients over both SEAs in such a way that half of them will use the first SEA and the other half will use the second SEA as primary adapter, as shown in Figure 4-10 on page 197. Thus, the bandwidth of the physical Ethernet adapters of both SEAs will be used.

In most cases, the advantages of SEA Failover will outweigh those of NIB, so SEA Failover should be the default approach to provide high-availability for bridged access to external networks.

Note: As SEA Failover is a feature of the Virtual I/O Server V1.2; with earlier releases of the Virtual I/O Server, only the NIB approach to highly available bridged external network access was available.

Important: You may experience up to 30 seconds delay in failover when using SEA Failover.

Note: There is a common behavior with both SEA Failover and NIB: They do not check the reachability of the specified IP address through the backup-path as long as the primary path is active. That is because the virtual Ethernet adapter is always connected and there is no *link up* event as is the case with physical adapters. You do not know if you really have an operational backup until your primary path fails.

Summary of HA alternatives for access to external networks

Table 4-2 summarizes the alternative approaches to achieve high-availability for shared access to external networks that have been discussed in the previous sections.

Table 4-2 Summary of HA alternatives for access to external networks

	Server-implementation	**Client-implementation**
Layer-2 / bridged	SEA Failover	NIB
Layer-3 / routed	Router failover	IPMP, VIPA, and IPAT

Of these approaches, the new SEA Failover feature is the most appropriate in typical IBM System p5 virtualization scenarios, so we will focus on SEA Failover in the rest of this redbook. You should refer to additional publications if you consider implementing one of the other alternatives.

4.1.4 System management with Virtual I/O Server

Redundancy allows for easier system management, such as software upgrades. With two Virtual I/O Servers, the layer of physical resource dependency has been removed.

System maintenance can now be performed on a Virtual I/O Server and any external device it connects to, such as a network or SAN switch. With virtual SCSI and shared Ethernet being hosted by a second Virtual I/O Server, rebooting or disconnecting the Virtual I/O Server from its external devices is now possible without causing client outages. With the client partition running MPIO and using SEA Failover, no actions will need to be performed on the client partition while the system maintenance is being performed or after it has completed. This results in improved uptime and reduced system administration efforts for the client partitions.

Upgrading and rebooting a Virtual I/O Server, network switch, or SAN switch is simpler and more compartmentalized, since the client is no longer dependant on the availability of all of the environment.

In Figure 4-11 on page 201, a client partition has virtual SCSI devices and a virtual Ethernet adapter hosted from two Virtual I/O Servers. The client has MPIO implemented across the virtual SCSI devices and Shared Ethernet Adapter Failover for the virtual Ethernet. When Virtual I/O Server 2 is shut down for maintenance, the client partition continues to access the network and SAN storage through Virtual I/O Server 1.

Note: Combination of MPIO for disk redundancy and Shared Ethernet Adapter Failover for network redundancy is recommended.

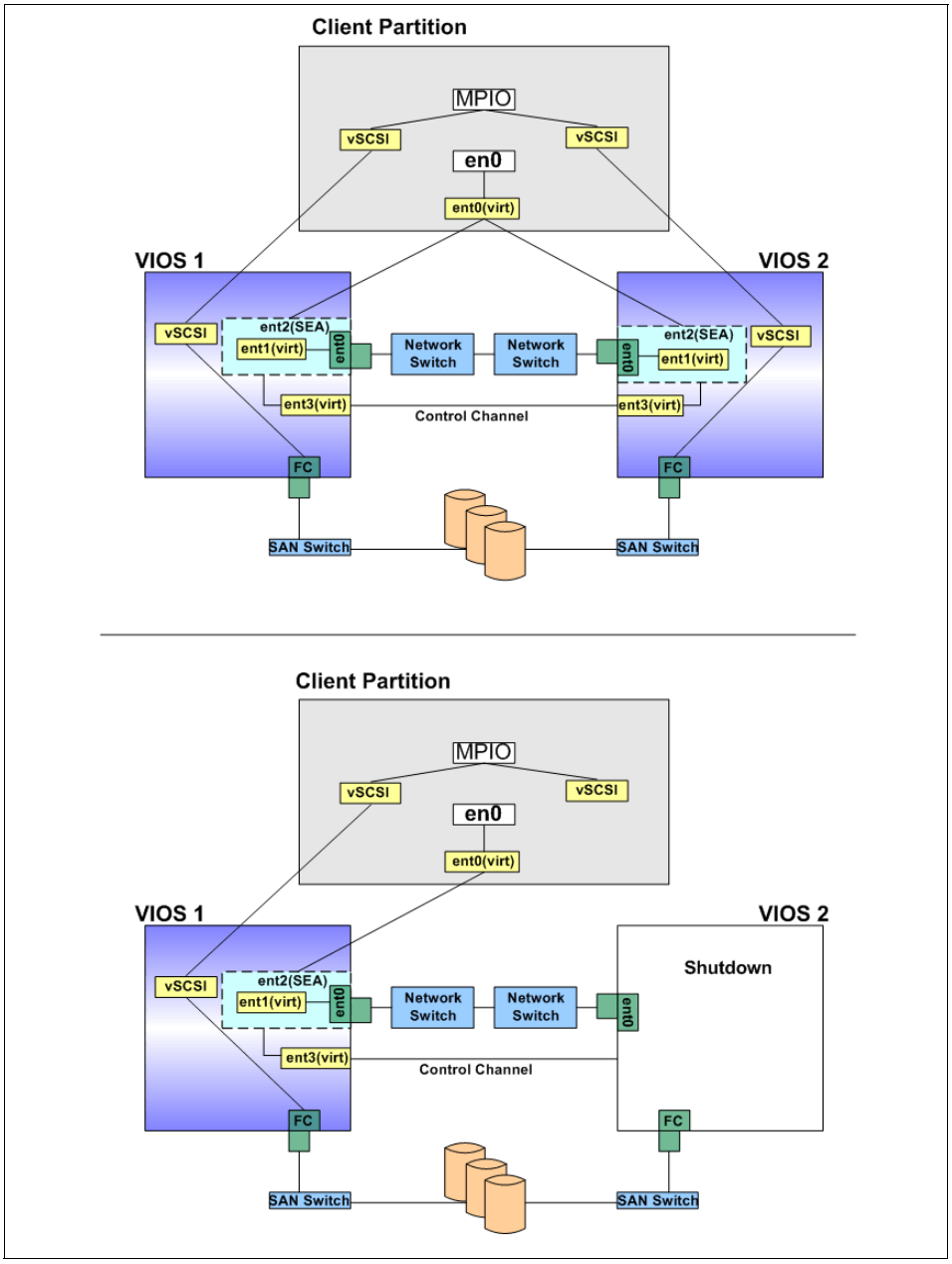


Figure 4-11 Redundant Virtual I/O Servers during maintenance

When Virtual I/O Server 2 returns to a full running state, the client will continue using the MPIO path through Virtual I/O Server 1, while the virtual Ethernet will be reset back to the primary Virtual I/O Server Shared Ethernet Adapter.

4.1.5 Virtual Ethernet implementation in the POWER Hypervisor

Virtual Ethernet connections use VLAN technology to ensure that the partitions can only access data directed to them. The POWER Hypervisor provides a virtual Ethernet switch function based on the IEEE 802.1Q VLAN standard that allows partition communication within the same server. The connections are based on an implementation internal to the POWER Hypervisor that moves data between partitions. This section describes the various elements of a virtual Ethernet and implications relevant to different types of workloads. Figure 4-12 is a simplified illustration of an inter-partition network emphasizing the interaction of the device drivers and the Hypervisor.

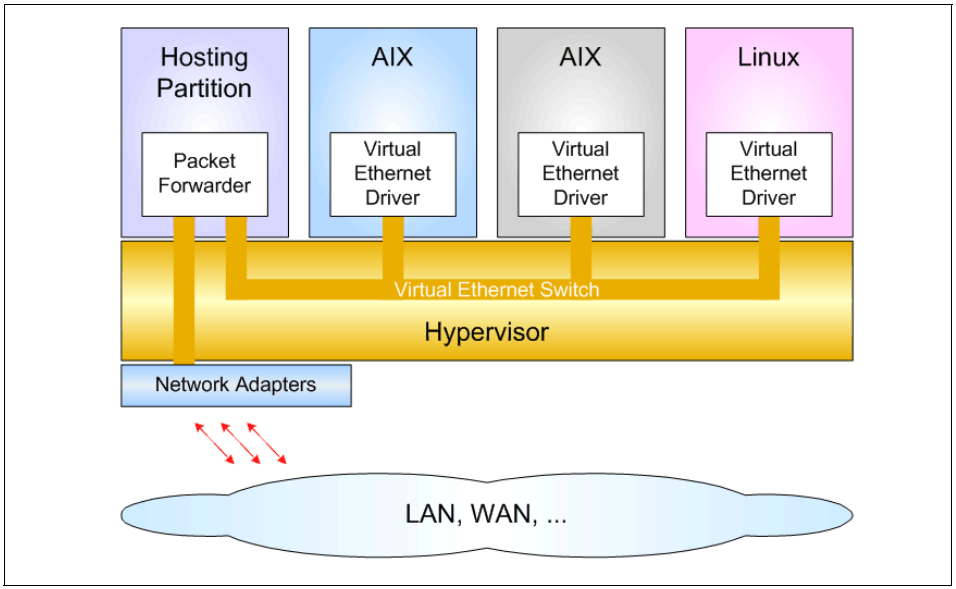


Figure 4-12 Logical view of an inter-partition VLAN

Virtual Ethernet adapter creation

Partitions that communicate through virtual Ethernet have additional in-memory channels implemented in the Hypervisor:

- The creation of in-memory channels between partitions occurs automatically when configuring virtual Ethernet adapters for partitions on the HMC or IVM.
- The AIX 5L or Linux kernel automatically creates a virtual network device for each memory channel indicated by the POWER5 firmware.
- The AIX 5L configuration manager creates the necessary ODM objects for the:
  - Ethernet network adapter device (ent*) with available state
  - Ethernet network interface device (en* and et*) with defined state

A unique 6 byte Media Access Control (MAC) address (also called Ethernet, hardware or layer-2 address) is generated when the virtual Ethernet device is created on the HMC or IVM. A *prefix* value can be assigned for the system so that the auto-generated MAC addresses in a system consist of a common system prefix, plus an algorithmically-generated unique part per adapter. Thus, the generated MAC addresses will not conflict with those of other network devices.

The virtual Ethernet can also be used as a bootable device to allow such tasks as operating system installations to be performed using NIM.

Dynamic partitioning for virtual Ethernet devices

Virtual Ethernet resources can be assigned and removed dynamically through dynamic LPAR-operations. On the HMC or IVM, virtual Ethernet target and server adapters can be assigned and removed from a partition using dynamic logical partitioning. The creation of physical and virtual Ethernet adapters on the Virtual I/O Server can also be done dynamically. After the addition of an adapter on the HMC, be it virtual or physical, the **cfgmgr** command has to be run in an AIX 5L partition, and the **cfgdev** command on the Virtual I/O Server.

4.1.6 Performance considerations for Virtual I/O Servers

The transmission speed of virtual Ethernet adapters is in the range of multiple gigabits per second, depending on the transmission size (MTU) and overall system performance. Virtual Ethernet connections generally take up more CPU cycles than connections through physical Ethernet adapters. The reason is that modern physical Ethernet adapters contain many functions to off-load some work from the system's CPUs, for example, checksum computation and verification, interrupt modulation, and packet reassembly. These adapters use Direct Memory Access (DMA) for transfers between the adapter and the RAM, which uses only a

few cycles to set up, but none for the actual transfers. Many disks can be mapped to the same server SCSI adapter without performance degradation.

- For shared processor partitions, performance will be limited by the partition definitions (for example, entitled capacity, and number of processors). Small partitions, with respect to small CPU entitlements, communicating with each other will experience more packet latency due to partition context switching.
- For dedicated processor partitions, throughput should be comparable to a Gigabit Ethernet for small packets, and much better for large packets. For large packets, the virtual Ethernet communication is memory-copy-bandwidth limited.

Tip: In general, high bandwidth applications should not be deployed in small shared processor partitions. Consider using physical adapters for high bandwidth applications.

Multiple Virtual I/O Servers also present different performance considerations, especially for larger installations. With many client partitions using both shared Ethernet and virtual SCSI, the load on the Virtual I/O Server can become excessive.

Tip: Consider using different Virtual I/O Server to separate competing workloads, such as network-latency sensitive applications and I/O bandwidth intensive applications, and to provide these workloads with guaranteed resources. In such environments, you might consider separating the virtual Ethernet from the virtual SCSI and place them in different Virtual I/O Server.

With a Virtual I/O Server dedicated solely to hosting virtual SCSI and another dedicated solely to Shared Ethernet, the idea of a device driver implementation becomes evident. Keeping redundancy in mind, having a pair of each adds fault tolerance to the environment and keeps the administration of each Virtual I/O Server simpler with respect to maintenance.

Separating disk and network traffic

When using MPIO for disk redundancy and SEA Failover for network redundancy, traffic can be separated to each Virtual I/O Server. In a SEA Failover configuration, the network traffic goes through the Virtual I/O Server with the trunk adapter set to the highest priority (lowest value). AIX 5L MPIO is also for failover and not load balancing. By default, both paths are set to priority 1 and the system will select the path to use.

By setting the path priority of MPIO so that disk traffic is directed to the Virtual I/O Server that is backup for SEA Failover, disk and network traffic are separated.

See 4.3, “Scenario 2: SEA Failover” on page 211 for details on SEA Failover configuration and 4.4, “Scenario 3: MPIO in the client with SAN” on page 218 for details on MPIO configuration.

Network traffic is generally heavier on the Virtual I/O Server than disk traffic so there could be unbalanced loads between the two servers. If the two Virtual I/O Servers are uncapped and shared, they would be automatically balanced.

Note: Do not switch off threading on the network server as it will do both disk and network serving in case the disk server is unavailable.

Note: Use the storage manager application to check that the preferred paths are in accordance with the path priorities set in the virtual I/O clients.

Note: The SEA Failover priority is set by the priority of the trunk adapter in the HMC and the MPIO path priority is set on each disk in the virtual I/O clients.

See Figure 4-13 for an example of separating traffic.

- VIOS1has priority 1 for network and priority 2 for disk.
- VIOS2 has priority 2 for network and priority 1 for disk.

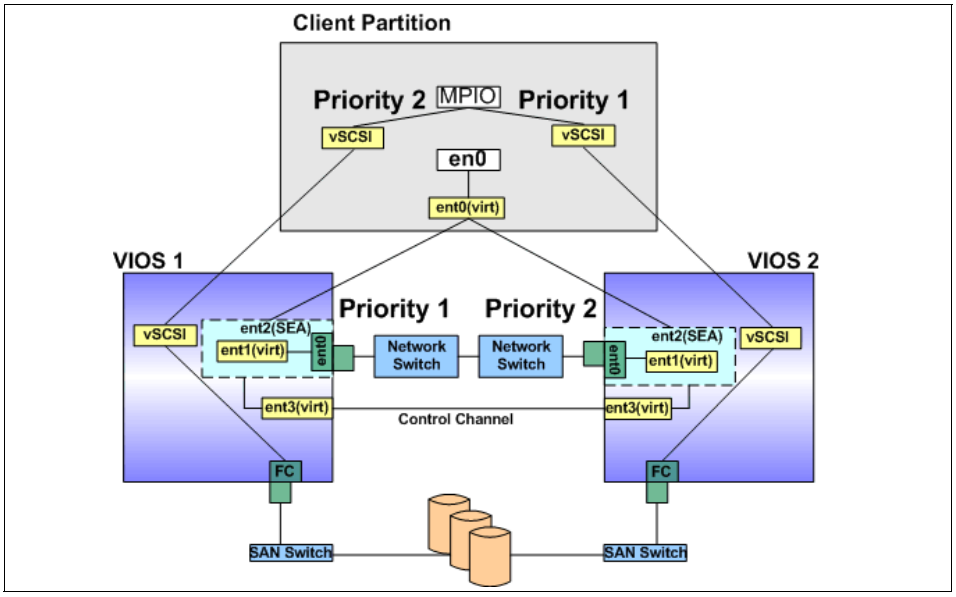


Figure 4-13 Separating disk and network traffic

4.1.7 Considerations

The following are for your consideration when implementing virtual Ethernet and Shared Ethernet Adapters in the Virtual I/O Server:

- Virtual Ethernet requires a POWER5-based system and an HMC or IVM to define the virtual Ethernet adapters.
- Virtual Ethernet is available on all POWER5-based systems, while Shared Ethernet Adapter and Virtual I/O Server may require the configuration of additional features on some models.
- Virtual Ethernet can be used in both shared and dedicated processor partitions.
- Virtual Ethernet does not require a Virtual I/O Server for communication between partitions in the same system.
- A maximum of up to 256 virtual Ethernet adapters are permitted per partition.
- Each virtual Ethernet adapter is capable of being associated with up to 21 VLANs (20 VIDs and 1 PVID).
- A system can support up to 4096 different VLANs, as defined in the IEEE802.1Q standard.
- The partition must be running AIX 5L Version 5.3 or Linux with the 2.6 kernel or a kernel that supports virtual Ethernet.
- A mixture of virtual Ethernet connections, real network adapters, or both are permitted within a partition.
- Virtual Ethernet can only connect partitions within a single system.
- Virtual Ethernet connections between AIX 5L and Linux partitions are supported.
- Virtual Ethernet connections from AIX 5L or Linux partitions to an i5/OS partition may work; however, at the time of writing, these capabilities were unsupported.
- Virtual Ethernet uses the system processors for all communication functions instead of off-loading that load to processors on network adapter cards. As a result, there is an increase in system processor load generated by the use of virtual Ethernet.
- Up to 16 virtual Ethernet adapters with 21 VLANs (20 VID and 1 PVID) on each can be associated to a Shared Ethernet Adapter, sharing a single physical network adapter.
- There is no explicit limit on the number of partitions that can attach to a VLAN. In practice, the amount of network traffic will limit the number of clients that can be served through a single adapter.

- To provide highly available virtual Ethernet connections to external networks, two Virtual I/O Servers with Shared Ethernet Adapter Failover or another network HA mechanism has to be implemented.
- You cannot use SEA Failover with Integrated Virtualization Manager (IVM), because IVM only supports a single Virtual I/O Server.

4.2 Scenario 1: Logical Volume Mirroring

In this scenario, we had modified our basic configuration and put in a second Virtual I/O Server named VIO_Server2 that will serve additional disks to our client partitions. Having this setup would satisfy the availability of rootvg through logical volume mirroring on the client side. See Figure 4-14 for an illustration of our logical volume mirroring setup.

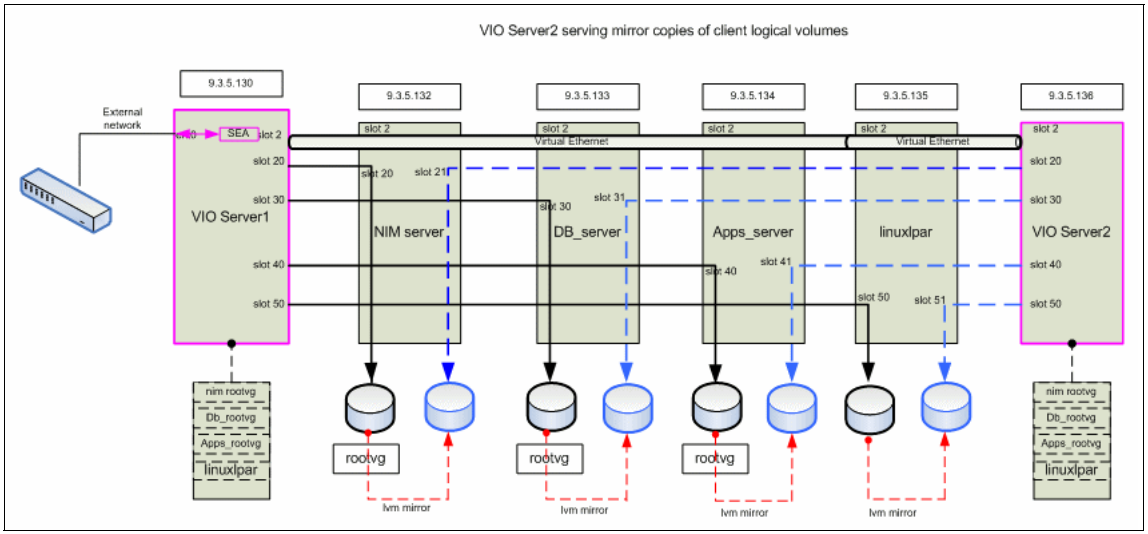


Figure 4-14 LVM mirroring scenario

Essentially, the steps in creating a secondary Virtual I/O Server to serve additional hdisks to the client partitions is somewhat the same as creating the first one. Follow the steps below to define the LVM solution:

1. Follow the instructions in 3.2.1, “Defining the Virtual I/O Server partition” on page 124 to create your second Virtual I/O Server with the following exceptions:

   a. Use the name VIO_Server2 on step 3 and step 5.

b. On step 10, allocate an unused Ethernet card slot and another storage controller slot to provide physical devices to our Virtual I/O Server. In our case, we had allocated Bus 2 Slot C4 and Bus 3 Slot T10 for the Ethernet and Storage controller, as shown in Figure 4-15.

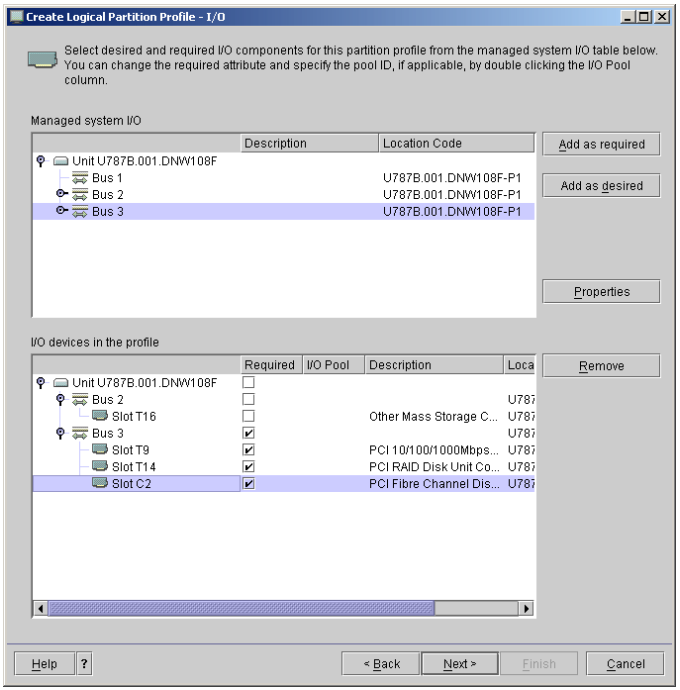


Figure 4-15 VIO_Server2 physical component selection

2. Follow the instructions in 3.3, “Virtual I/O Server software installation” on page 142 to install the Virtual I/O Server software on the Virtual I/O Server VIO_Server2 partition.
3. Create the virtual SCSI adapters on the VIO_Server2 partition that will hold the logical volume devices that will be shared to client partitions. Refer to 3.4.1, “Creating virtual SCSI server adapters” on page 146 on how to create virtual SCSI adapters.
4. Based on Figure 4-14 on page 207, create the client virtual SCSI adapters (on NIM_server, DB_server, Apps_server, and linuxlpar partitions) that will be mapped to the virtual SCSI slots on the VIO_Server2. Refer to 3.4.1, “Creating virtual SCSI server adapters” on page 146 for instructions on how to create virtual SCSI adapters.

5. This is how the mappings would look like on VIO_Server2 partition from the HMC, as shown in Figure 4-16.

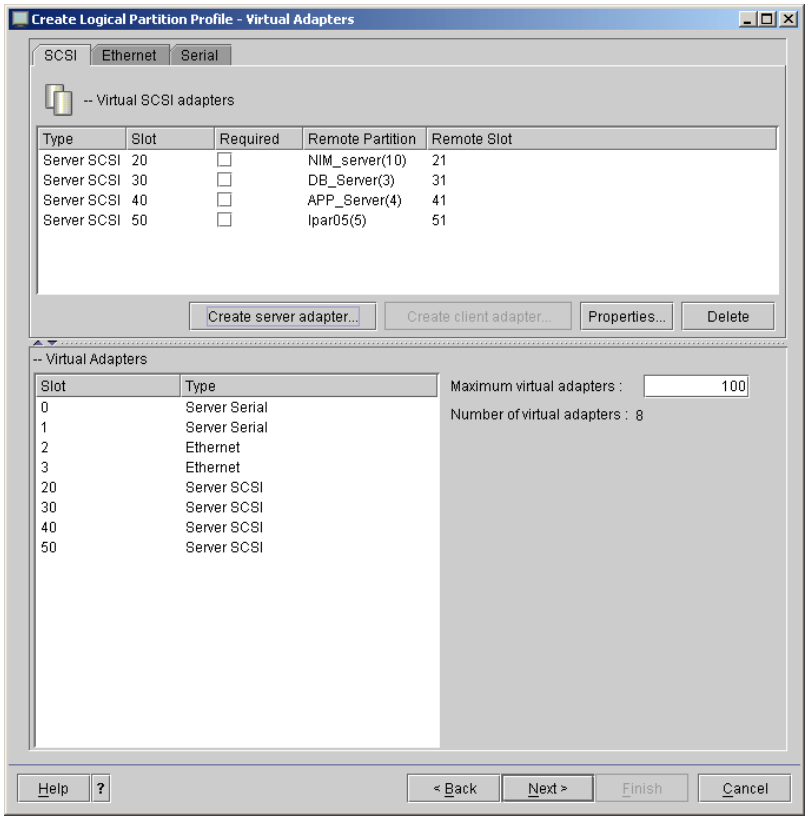


Figure 4-16 VIO_Server2 partition virtual SCSI properties view

6. You are ready to create the volume group and logical volumes on the second Virtual I/O Server. Refer to 3.4.4, “Defining virtual disks” on page 163 for more information.

7. Check the devices list on your VIO_Server2 partition with the **lsdev** command, as shown in Example 4-1.

Example 4-1 VIO_Server2 partition

```
$ lsdev -virtual
name             status     description
ent1             Available  Virtual I/O Ethernet Adapter (l-lan)
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0           Available  Virtual SCSI Server Adapter
vhost1           Available  Virtual SCSI Server Adapter
vhost2           Available  Virtual SCSI Server Adapter
vhost3           Available  Virtual SCSI Server Adapter
vhost4           Available  Virtual SCSI Server Adapter
vsa0             Available  LPAR Virtual Serial Adapter
vnim              Available  Virtual Target Device - Logical Volume
vapps            Available  Virtual Target Device - Logical Volume
vdbsrv           Available  Virtual Target Device - Logical Volume
vlnx             Available  Virtual Target Device - Logical Volume
```

8. When you bring up one of the partitions, you should have hdisk1 present, as shown in Example 4-2. Mirror your rootvg as you would normally do on AIX 5L.

Example 4-2 NIM_server partition with hdisk1 served off of VIO_Server2

```
# hostname
NIM_server
# lspv
hdisk0          00c5e9de205cf5c6                    rootvg          active
hdisk1          none                                None            active
# lsdev -Cc disk
hdisk0 Available  Virtual SCSI Disk Drive
hdisk1 Available  Virtual SCSI Disk Drive
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
# lspv
hdisk0          00c5e9de205cf5c6                    rootvg          active
hdisk1          00c5e9dea5571a32                    rootvg          active

# mirrorvg rootvg hdisk1
0516-1124 mirrorvg: Quorum requirement turned off, reboot system for this
        to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records.  Then, user must modify
        bootlist to include:  hdisk1 hdisk0.
# bosboot -a -d /dev/hdisk1
bosboot: Boot image is 30420 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -r
```

```
hdisk0 blv=hd5
hdisk1 blv=hd5
```

4.3 Scenario 2: SEA Failover

This scenario will show you how to modify an existing SEA to use SEA Failover and how to create a second SEA in failover mode on VIO_Server2, which is going to be a backup path in case the primary SEA on VIO_Server1 is not available. This is a feature that came first with Virtual I/O Server V1.2.

The highly-available Shared Ethernet Adapter setup is achieved by creating an additional virtual Ethernet adapter on each Virtual I/O Server, defining it as the control channel for each SEA, and modifying two attributes on each Shared Ethernet Adapter. The control channel is used to carry heartbeat packets between the two VIO servers. A feature that was introduced in Virtual I/O Server V1.2 when creating virtual Ethernet adapter is the Access External Networks check box selection. This is where you would specify the primary and backup adapter by giving them different priority numbers.

Figure 4-17 shows the intended solution for a highly available Shared Ethernet Adapter.

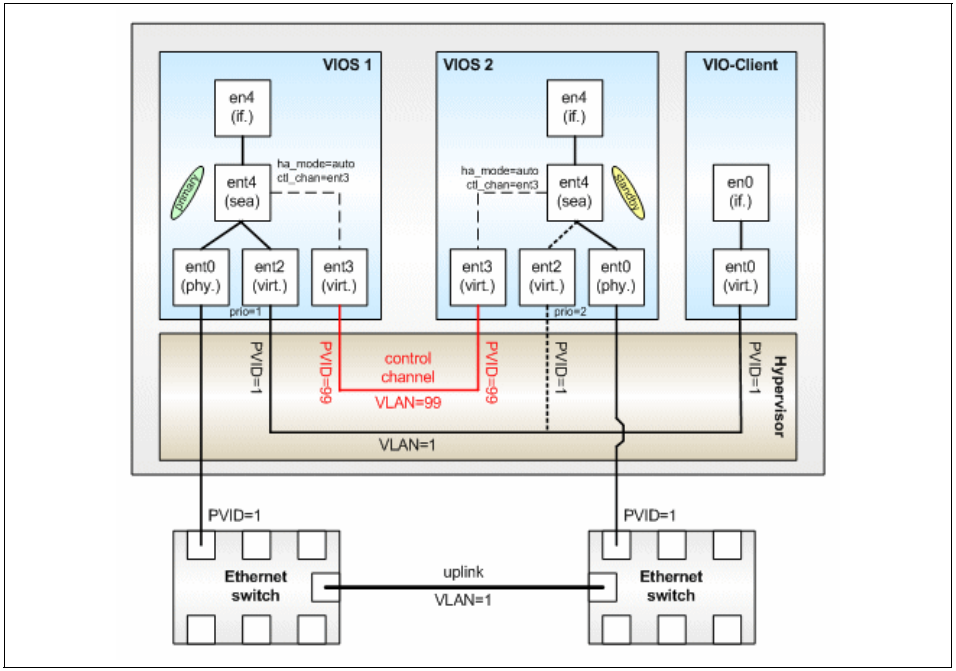


Figure 4-17 Highly available SEA adapter setup

The following steps will guide you through the setup process of a secondary backup SEA adapter:

1. Dynamically create the virtual Ethernet adapter on each Virtual I/O Server that will act as a control channel:

   a. On the VIO_Server1 partition, right-click it and set **Dynamic Logical Partitioning** → **Virtual Adapter Resources** → **Add/Remove**, as shown in Figure 4-18.

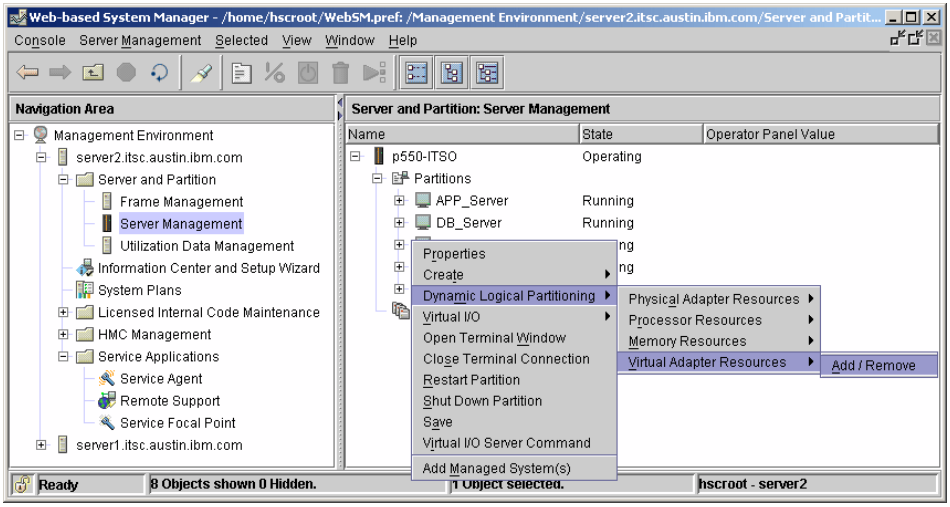


Figure 4-18 VIO_Server1 dynamic LPAR operation to add virtual Ethernet

b. Click the **Create Adapter** button, and then input the value for Virtual LAN ID, as shown in Figure 4-19 (the value for Slot will be filled in automatically. The use of the default value is sufficient). Click the **OK** button when done.

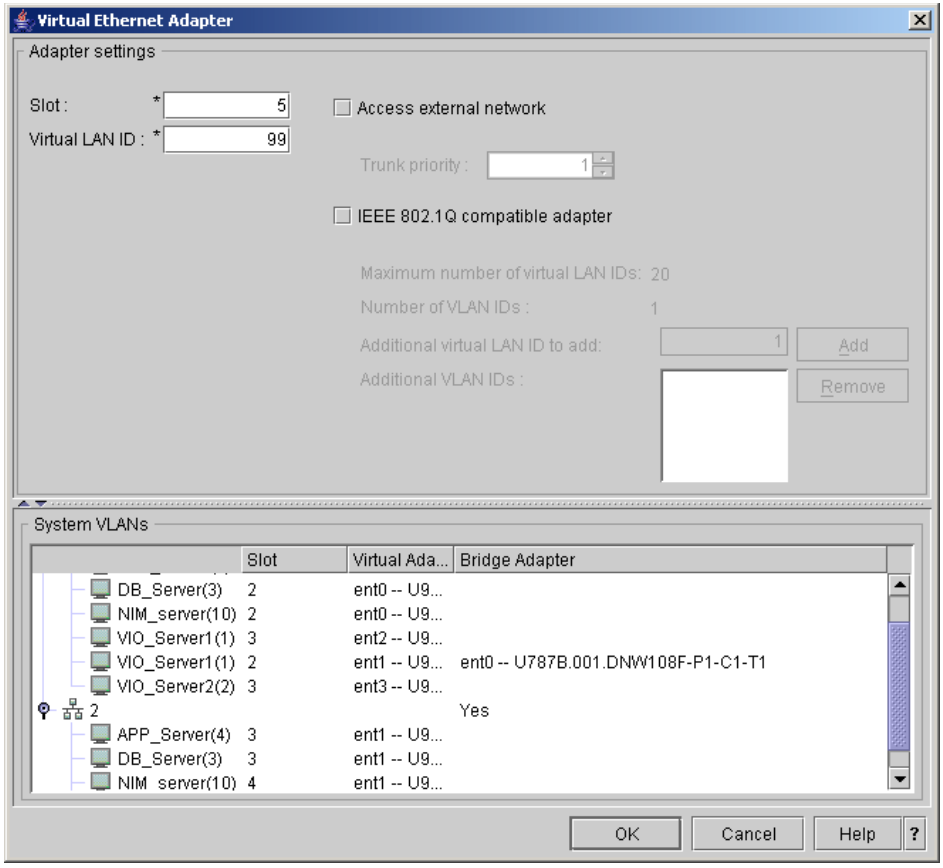


Figure 4-19 Virtual Ethernet Slot and Virtual LAN ID (PVID) value

c. Perform the same steps on the VIO_Server2 profile.

Note: The two adapters you are creating need to be on a PVID that is unique for both (for example, both adapters must have the same PVID, and the PVID must not be used by any other adapter on the machine). As Figure 4-17 on page 212 shows, we put them on VLAN 99.

2. Right-click the VIO_Server2 partition and create a virtual Ethernet adapter. The value for Virtual LAN ID must be the same as the primary SEA (in this example, 1) and the Access External Network check box must be checked.

The Trunk priority flag for this SEA adapter is set to 2. This is for making the SEA adapter on VIO_Server2 the backup adapter. Click **OK** when done.

3. For the adapters to be available on both Virtual I/O Servers, you need to log in using the padmin user ID and run the **cfgdev** command on each Virtual I/O Server, since they were added dynamically.

Tip: The settings for the Virtual Devices on each Virtual I/O Server are:

On VIO_Server1:

- SEA Virtual Ethernet properties:
  - Virtual LAN ID 1
  - Trunk priority 1
  - Access external network button checked
- Virtual Ethernet for control channel use: Virtual LAN ID 99

On VIO_Server2:

- SEA Virtual Ethernet properties:
  - PVID 1
  - Trunk Priority 2
  - Access external network button checked
- Virtual Ethernet for control channel use: Virtual LAN ID 99

4. Change the SEA adapter device on VIO_Server1 using the **chdev** command, as shown in Example 4-3.

Example 4-3 Shared Ethernet Adapter on VIO_Server1

```
$chdev -dev ent4 -attr ha_mode=auto ctl_chan=ent3
ent4 changed
```

Define the SEA adapter device on VIO_Server2 using the **mkdev** command, as shown in Example 4-4.

Example 4-4 Shared Ethernet Adapter on VIO_Server2

```
$mkvdev -sea ent0 -vadapter ent2 -default ent2
 -defaultid 1 -attr ha_mode=auto ctl_chan=ent3
ent4 Available
en4
et4
```

Tip: Mismatching SEA and SEA Failover could cause broadcast storms to occur and affect the stability of your network. When upgrading from an SEA to an SEA Failover environment, it is imperative that the Virtual I/O Server with regular SEA be modified to SEA Failover *prior* to creating the second SEA with SEA Failover enablement.

5. Verify the SEA adapter attributes on both Virtual I/O Servers' SEA adapter, as shown in Example 4-5.

Example 4-5 Verify and change attributes for SEA adapter

```
$ lsdev -dev ent4 -attr
attribute     value    description                                user_settable

ctl_chan      ent3     Control Channel adapter for SEA failover       True
ha_mode       auto     High Availability Mode                         True
netaddr        Address to ping                                        True
pvid          1        PVID to use for the SEA device                 True
pvid_adapter  ent2     Default virtual adapter to use for non-VLAN-tagged
packets                                                               True
real_adapter  ent0     Physical adapter associated with the SEA       True
thread        0        Thread mode enabled (1) or disabled (0)        True
virt_adapters ent2     List of virtual adapters associated with the SEA (comma
separated)                                                            True
```

6. Create the Shared Ethernet Adapter IP address on VIO_Server2 using the **mktcpip** command, as shown in Example 4-6.

Example 4-6 Create an IP address on the Shared Ethernet Adapter

```
$ mktcpip -hostname VIO_Server2 -interface en4 -inetaddr 9.3.5.136
-netmask 255.255.255.0 -gateway 9.3.5.41 -nsrvaddr 9.3.4.2 -nsrvdomain
itsc.austin.ibm.com
```

Testing the SEA Failover

You should test the SEA Failover to validate your setup and configuration.

Test setup

To test if the SEA Failover really works as expected, you should open a remote shell session from any system on the external network, such as your workstation, through the SEA adapter to any VIO client partition. From the shell session, try running any command that continuously produces output. Now you are ready to begin the tests.

Test cases

This section describes the test cases that you should go through to confirm that your setup of SEA Failover works as expected.

Attention: If your Virtual I/O Servers also provide virtual SCSI disks in addition to SEA, then stale partitions will occur on the VIO clients when a Virtual I/O Server shuts down or fails. Thus, after each test, you must make sure that you re-synchronize all stale partitions before proceeding with the next test-case.

1. Manual failover:
   a. Set ha_mode to standby on primary: the SEA is expected to fail over;

      ```
      chdev -dev ent2 -attr ha_mode=standby
      ```

   b. Reset ha_mode to auto on primary: The SEA is expected to fail back:

      ```
      chdev -dev ent2 -attr ha_mode=auto
      ```

2. Virtual I/O Server shutdown:
   a. Reboot Virtual I/O Server on primary: The SEA is expected to fail over.
   b. When primary Virtual I/O Server comes up again: The SEA is expected to fail back.
3. Virtual I/O Server error:
   a. Deactivate from HMC on primary: The SEA is expected to fail over.
   b. Activate and boot Virtual I/O Server: The SEA is expected to fail back.
4. Physical link failure:
   a. Unplug the link of the physical adapter on the primary: The SEA is expected to fail over.
   b. Re-plug the link of the physical adapter on the primary: The SEA is expected to fail back.
5. Reverse the boot sequence:
   a. Shut down both Virtual I/O Server.
   b. Boot the standby Virtual I/O Server: The SEA is expected to become active on standby.
   c. Boot the primary Virtual I/O Server: The SEA is expected to fail back.

Checking which SEA is active

Checking which Virtual I/O Server is active is done in each server with the **entstat** command, as shown in Example 4-7.

Example 4-7 Using the entstat command to check which SEA is active

```
$ entstat -all ent3 | grep Active
  Priority: 1  Active: True
```

4.4 Scenario 3: MPIO in the client with SAN

This section describes the setup of an advanced scenario with two Virtual I/O Servers attached to a DS4200. Both Virtual I/O Servers serve the same DS4200 LUN to the client partition. On the client partition, the use of MPIO with the default PCM provides redundant access to the served LUN. Figure 4-20 shows this scenario.

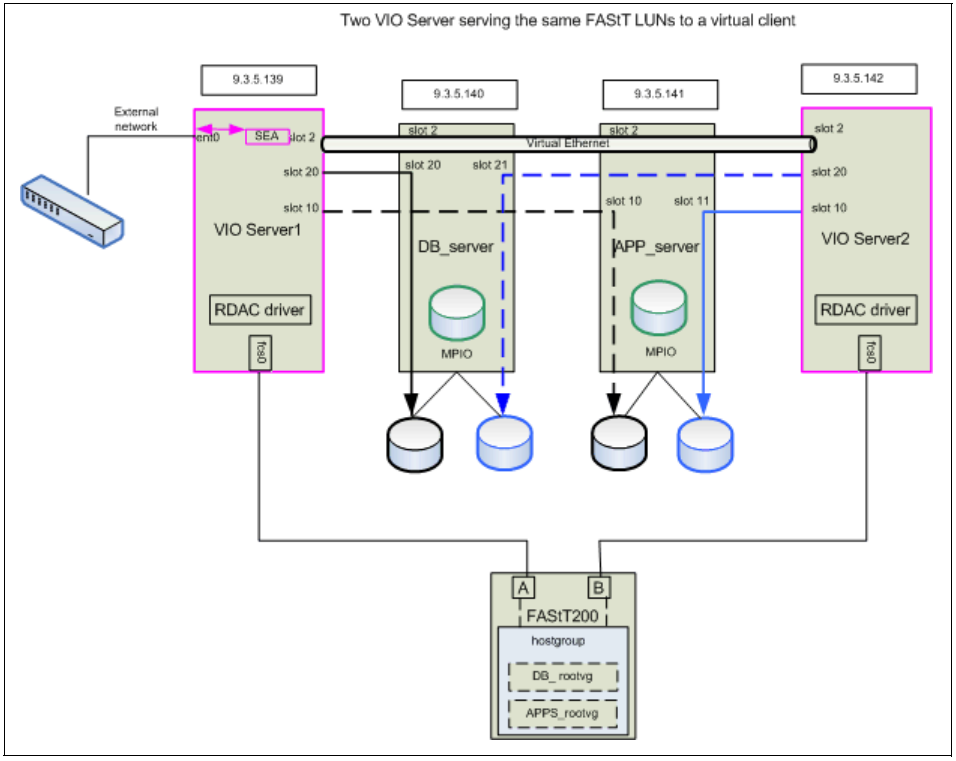


Figure 4-20 SAN attachment with multiple Virtual I/O Server

Table 4-3 and Table 4-4 provide the configuration information used.

Table 4-3 Defining the client SCSI adapter for the DB_server partition

Client slot	Server partition	Server partition slot
20	VIO_Server1	20
21	VIO_Server2	20

Table 4-4 Defining the client SCSI adapter for the APP_server partition

Client slot	Server partition	Server partition slot
10	VIO_Server1	10
11	VIO_Server2	10

In this scenario, we attach each Virtual I/O Server with one Fibre Channel adapter directly to a DS4200. On the Virtual I/O Servers, the RDAC driver is already provided in the base, regardless of the existence of multiple Fibre Channel adapters.

On the DS4200, we configured four LUNs that belong to one hostgroup. The hostgroup consists of Virtual I/O Server 1 and Virtual I/O Server 2, so both Virtual I/O Servers are attached to the same disk. In addition, we configured two disks that belong to each of the Virtual I/O Servers to be used for rootvg on the servers.

Figure 4-21 shows an overview of the hostgroup mapping on the DS4200.

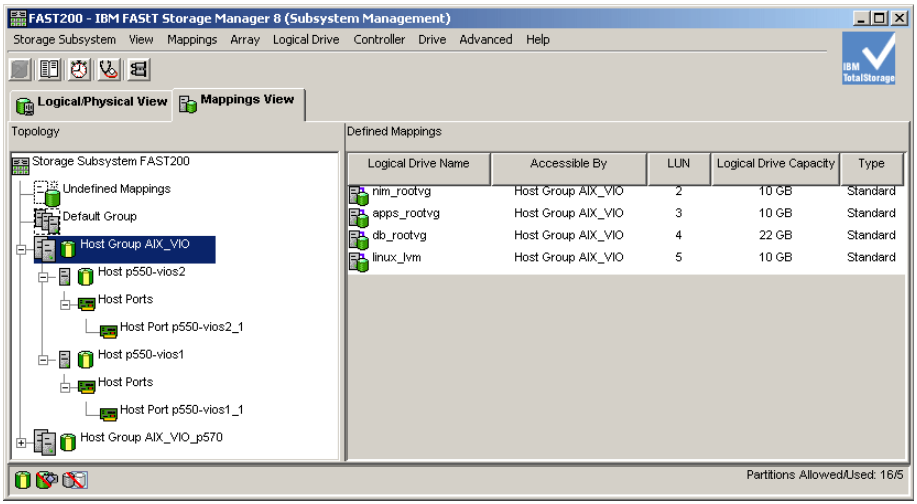


Figure 4-21 Overview of the DS4200 configuration

In this scenario, we attach each Virtual I/O Server with one Fibre Channel adapter directly to a DS4200. In our case, we used this configuration only to show the basic configuration.

When using only one Fibre Channel per Virtual I/O Server, you need an additional switch to have a supported configuration. In this case, it is important that the SAN zoning be configured such that the single HBA in each Virtual I/O Server LPAR is zoned to see both storage controllers in the FAStT. If a second HBA is used for additional redundancy, the storage administrator must ensure that each HBA is zoned to only one of the DS4200 controllers. Rules for attachment of FAStT storage units to AIX can be found in the Storage Manager documentation of the Storage Manager products.

On the Virtual I/O Server, the RDAC driver is always provided, regardless of the existence of multiple Fibre Channel adapters.

4.4.1 Setup on the HMC

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 3.2, “Creating a Virtual I/O Server partition” on page 124. In step 10, select one Fibre Channel adapter in addition to the shown physical adapter.
2. Install both Virtual I/O Servers by following the instructions in 3.3, “Virtual I/O Server software installation” on page 142.
3. After the successful installation, you can map the world wide names of the Fibre Channel adapters to the created LUN. Use the **cfgdev** command to make the disks available on the Virtual I/O Server.
4. Create two client partitions named DB_server and APP_server following the instructions in 3.4.3, “Creating client partitions” on page 151. Create the server adapters when you create the client adapters using dynamic LPAR following the plan in Table 4-3 on page 219 and Table 4-4 on page 219. Also add one or two virtual Ethernet adapters to each client, one adapter if you plan on using SEA Failover for network redundancy, as described in 4.3, “Scenario 2: SEA Failover” on page 211, or two adapters if you plan on using Network Interface Backup for network redundancy, as described in 4.5, “Scenario 4: Network Interface Backup in the client” on page 234.

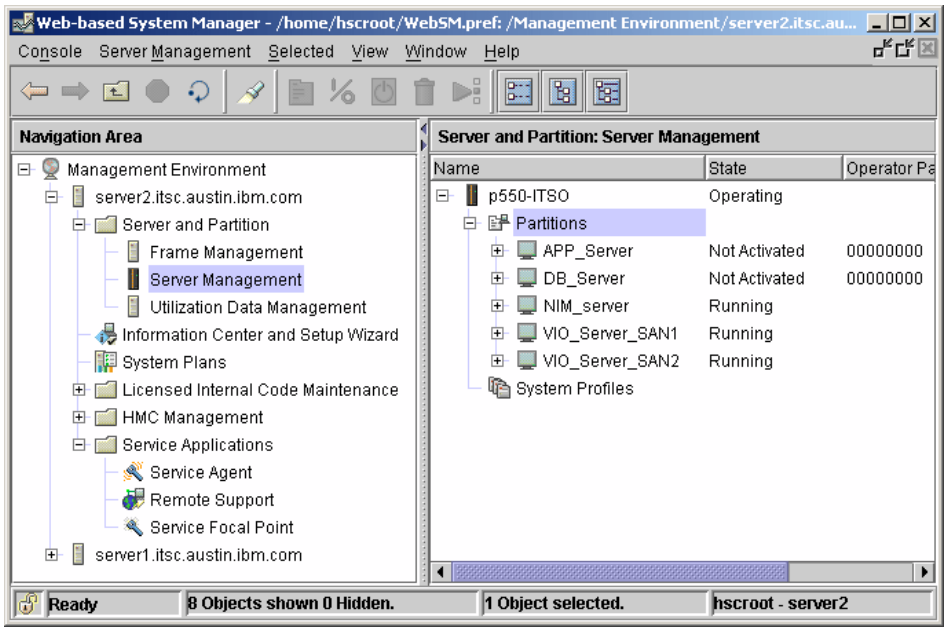


Figure 4-22 Starting configuration for the scenario

5. Server adapters have been added using dynamic LPAR. To save the configuration, you either have to update the Virtual I/O Server profiles or save the new configuration as a new profile, as described in 3.4.3, “Creating client partitions” on page 151.

4.4.2 Configuration on the Virtual I/O Servers

In order to configure the disks on VIO_Server1, follow these steps:

1. Open a terminal window to the VIO_Server1 partition by selecting the partition, right-click, and choose **Open Terminal Window**.

2. Configure the SEA, as described in step 1 to 4 in 3.4.2, “Creating a Shared Ethernet Adapter” on page 149. In our example, we use the network settings shown in Table 4-5 and Table 4-6.

Table 4-5 Network settings for VIO_Server1

Setting	Value
host name	VIO_Server1
IP-address	9.3.5.110
netmask	255.255.255.0
gateway	9.3.5.41

Table 4-6 Network settings for VIO_Server2

Setting	Value
host name	VIO_Server2
IP-address	9.3.5.112
netmask	255.255.255.0
gateway	9.3.5.41

3. Check for the attached DS4200 disks using the **lsdev** command with the -type flag, as shown in Example 4-8.

Example 4-8 Check for the DS4200 disks

```
$ lsdev -type disk
name            status     description
hdisk0          Available  16 Bit LVD SCSI Disk Drive
hdisk1          Available  16 Bit LVD SCSI Disk Drive
hdisk2          Available  16 Bit LVD SCSI Disk Drive
hdisk3          Available  16 Bit LVD SCSI Disk Drive
hdisk4          Available  3542     (200) Disk Array Device
hdisk5          Available  3542     (200) Disk Array Device
hdisk6          Available  3542     (200) Disk Array Device
hdisk7          Available  3542     (200) Disk Array Device
hdisk8          Available  3542     (200) Disk Array Device
```

The output of the command lists four internal SCSI disks and five DS4200 disks.

4. You can use the **fget_config** command from the DS4200 to get the LUN to hdisk mappings, as shown in Example 4-9 on page 223. We will use hdisk6 and hdisk7 on VIO_Server1 for our clients.

Example 4-9 Listing the LUN to hdisk mappings

```
# fget_config -Av
---dar0---
User array name = 'FAST200'
dac0 ACTIVE dacNONE ACTIVE
Disk     DAC   LUN Logical Drive
hdisk4   dac0      1
hdisk5   dac0      2 nim_rootvg
hdisk6   dac0      3 apps_rootvg
hdisk7   dac0      4 db_rootvg
hdisk8   dac0      5 linux_lvm
```

You could also use the **lsdev -dev hdisk*n* -vpd** command, where *n* is the hdisk number, to retrieve this information, as shown in Example 4-10.

Example 4-10 Listing the LUN to hdisk mapping using the **lsdev** *command*

```
$ lsdev -dev hdisk6 -vpd
  hdisk6
U787B.001.DNW108F-P1-C3-T1-W200500A0B8110D0F-L3000000000000  3542
(200) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:  block
```

5. Check for the vhost adapter using the **lsmap** command with the -all flag, as shown in Example 4-11. The slot number that we configured on the HMC is shown in the Physloc column. The Server SCSI adapter we configured in slot 10 shows up in location C10.

Example 4-11 Listing vhost adapters

```
$ lsmap -all
SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost0          U9113.550.105E9DE-V1-C4                     0x0000000a

VTD                   vcd
LUN                   0x8100000000000000
Backing device        cd0
Physloc               U787B.001.DNW108F-P4-D2

SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost2          U9113.550.105E9DE-V1-C10                    0x00000003

VTD                   NO VIRTUAL TARGET DEVICE FOUND

SVSA            Physloc                                     Client
Partition ID
--------------- ------------------------------------------
------------------
vhost3          U9113.550.105E9DE-V1-C20                    0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND
```

6. The disks are to be accessed through both Virtual I/O Servers. The reserve_policy must be set to no_reserve. Check the attributes of the hdisk2 for the settings of the reserve_policy attribute using the **lsdev** command, as shown in Example 4-12 on page 225.

Example 4-12 Showing the attribute of hdisks on the Virtual I/O Server

```
$ lsdev -dev hdisk6 -attr
attribute      value                              description
user_settable

PR_key_value   none                               Persistant Reserve Key Value           True
cache_method   fast_write                         Write Caching method                   False
ieee_volname   600A0B80000BDC160000051D451C54AC   IEEE Unique volume name                False
lun_id         0x0003000000000000                 Logical Unit Number                    False
max_transfer   0x100000                           Maximum TRANSFER Size                  True
prefetch_mult  1                                  Multiple of blocks to prefetch on read False
pvid           00c5e9de0fbcc9210000000000000000   Physical volume identifier             False
q_type         simple                             Queuing Type                           False
queue_depth    10                                 Queue Depth                            True
raid_level     5                                  RAID Level                             False
reassign_to    120                                Reassign Timeout value                 True
reserve_policy single_path                        Reserve Policy                         True
rw_timeout     30                                 Read/Write Timeout value               True
scsi_id        0x10000                            SCSI ID                                False
size           10240                              Size in Mbytes                         False
write_cache    yes                                Write Caching enabled                  False
rw_timeout     30                                 Read/Write Timeout value               True
scsi_id        0xef                               SCSI ID                                False
size           51200                              Size in Mbytes                         False
write_cache    yes                                Write Caching enabled                  False
```

7. Change the reserve_policy attribute from single_path to no_reserve using the **chdev** command on hdisk6, as shown in Example 4-13.

Example 4-13 Set the attribute to no_reserve

```
$ chdev -dev hdisk6 -attr reserve_policy=no_reserve
hdisk6 changed
```

8. Check again, using the **lsdev** command, to make sure reserve_policy attribute is now set to no_reserve, as shown in Example 4-14.

Example 4-14 Output of the hdisk attribute after changing the reserve_policy attribute

```
$ lsdev -dev hdisk6 -attr
attribute      value                            description
user_settable

PR_key_value   none                             Persistant Reserve Key Value           True
cache_method   fast_write                       Write Caching method                   False
ieee_volname   600A0B80000BDC160000051D451C54AC IEEE Unique volume name               False
lun_id         0x0003000000000000               Logical Unit Number                    False
max_transfer   0x100000                         Maximum TRANSFER Size                  True
prefetch_mult  1                                Multiple of blocks to prefetch on read False
pvid           00c5e9de0fbcc9210000000000000000 Physical volume identifier            False
q_type         simple                           Queuing Type                           False
queue_depth    10                               Queue Depth                            True
raid_level     5                                RAID Level                             False
reassign_to    120                              Reassign Timeout value                 True
reserve_policy no_reserve                       Reserve Policy                         True
rw_timeout     30                               Read/Write Timeout value               True
scsi_id        0x10000                          SCSI ID                                False
size           10240                            Size in Mbytes                         False
write_cache    yes                              Write Caching enabled                  False
```

9. Repeat steps 5 to 7 for hdisk7.

10. For the Fibre Channel adapter, change the attribute fc_err_recov to fast_fail and dyntrk to yes. You can use the **lsdev -type adapter** command to find the number of the Fibre Channel adapter. Use the **chdev** command, as shown in Example 4-15.

Example 4-15 Changing the attributes of the Fibre Channel adapter

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
$ lsdev -dev fscsi0 -attr
attribute    value     description
user_settable

attach       switch    How this adapter is CONNECTED         False
dyntrk       yes       Dynamic Tracking of FC Devices        True
fc_err_recov fast_fail FC Fabric Event Error RECOVERY Policy True
scsi_id      0x10300   Adapter SCSI ID                       False
sw_fc_class  3         FC Class for Fabric                   True
```

Note: The reason for changing the fc_err_recov attribute to fast_fail is that if the Fibre Channel adapter driver detects a link event, such as a lost link between a storage device and a switch, then any new I/O or future retries of the failed I/O operations will be failed immediately by the adapter until the adapter driver detects that the device has rejoined the fabric. The default setting for this attribute is delayed_fail.

Setting the dyntrk attribute to yes makes AIX 5L tolerate cabling changes in the SAN. Be aware that this function is not supported on all storage systems. Check with your storage vendor for support.

11. Reboot the Virtual I/O Server for the changes to take effect. You only need to reboot when you change the Fibre Channel attributes.
12. Log on to VIO_Server2. Repeat steps 1 to 10 configure the SEA and either configure the IP interface on the SEA or on a separate virtual Ethernet adapter with the same PVID. In our example, VIO_Server2 has the IP address 9.3.5.112.

    Double check both Virtual I/O Servers that the vhost adapters have the correct slot numbers by running the **lsmap -all** command.
13. Ensure that the disks you want to map to the clients have the correct hdisks and LUN numbers on both Virtual I/O Server using the **lsdev** command, as shown in Example 4-16.
14. Map the hdisk to the vhost adapter using the **mkvdev** command, as shown in Example 4-16.

Example 4-16 Mapping the disks to the vhost adapters

```
$ mkvdev -vdev hdisk6 -vadapter vhost2 -dev app_server
app_server Available
$ mkvdev -vdev hdisk7 -vadapter vhost3 -dev db_server
db_server Available
```

Important: When multiple Virtual I/O Servers attach to the same disk, only hdisk is supported as a backing device. You cannot create a volume group on these disks and use a logical volume as a backing device.

Check the mappings, as shown in Example 4-17.

Example 4-17 lsmap -all output after mapping the disks on VIO_Server1

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9113.550.105E9DE-V1-C4                     0x0000000a

VTD                   vcd
LUN                   0x8100000000000000
Backing device        cd0
Physloc               U787B.001.DNW108F-P4-D2

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost2          U9113.550.105E9DE-V1-C10                    0x00000000

VTD                   app_server
LUN                   0x8100000000000000
Backing device        hdisk6
Physloc               U787B.001.DNW108F-P1-C3-T1-W200500A0B8110D0F-L3000000000000

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost3          U9113.550.105E9DE-V1-C20                    0x00000000

VTD                   db_server
LUN                   0x8100000000000000
Backing device        hdisk7
Physloc               U787B.001.DNW108F-P1-C3-T1-W200500A0B8110D0F-L4000000000000
```

Notice the LUN numbers from the listing.

15. Log on to the VIO_Server2 and repeat steps 13 to 14. Example 4-18 on page 229 shows the output of the **lsmap** command after mapping both disks to the vhosts adapters.

Example 4-18 lsmap -all output after mapping the disks on VIO_Server2

```
$ lsmap -all
SVSA            Physloc                                      Client PartitionID
--------------- -------------------------------------------- ------------------
vhost0          U9117.570.107CD9E-V2-C10                     0x00000004

VTD                   app_server
LUN                   0x8100000000000000
Backing device        hdisk2
Physloc               U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L0

SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.570.107CD9E-V2-C20                     0x00000003

VTD                   db_server
LUN                   0x8100000000000000
Backing device        hdisk3
Physloc               U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L1000000000000
```

16. Install the two client partitions using NIM or assign optical devices as a desired resource to install AIX 5L. For instructions on installing the AIX 5L client partitions, refer to 3.4.5, “Client partition AIX 5L installation” on page 169.

4.4.3 Working with MPIO on the client partitions

AIX 5L will automatically recognize the disk as an MPIO disk.

Note: MPIO in AIX 5L is currently for failover only.

Configure the client partition DB_server with an active path over the VIO_Server1 and the client partition APP_server with an active path over the VIO_Server2.

Alternatively, if you use SEA Failover for network redundancy, you could direct the active path to the backup Virtual I/O Server for the network. See “Separating disk and network traffic” on page 204 for details on separating traffic.

The following are the steps to configure on the client partitions:

1. Log on to your client partition APP_server.

2. Check the MPIO configuration by running the commands shown in Example 4-19. Only one configured hdisk shows up in this scenario.

Example 4-19 Verifying the disk configuration on the client partitions

```
# lspv
hdisk0          00c7cd9eabe9f4bf                    rootvg
active
# lsdev -Cc disk
hdisk0 Available  Virtual SCSI Disk Drive
```

3. Run the **lspath** command to verify that the disk is attached using two different paths. Example 4-20 shows that hdisk0 is attached using the VSCSI0 and VSCSI1 adapter that point to the different Virtual I/O Server. Both Virtual I/O Servers are up and running. Both paths are enabled.

Example 4-20 Verifying the paths of hdisk0

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

4. Configure the client partition to update the mode of a path using the **chdev** command for the hcheck_interval attribute of hdisk0. To check for the attribute setting, use the **lsattr** command, as shown in Example 4-21.

Example 4-21 Showing the attributes of hdisk0

```
# lsattr -El hdisk0
PCM             PCM/friend/vscsi                 Path Control Module        False
algorithm       fail_over                        Algorithm                  True
hcheck_cmd      test_unit_rdy                    Health Check Command       True
hcheck_interval 0                                Health Check Interval      True
hcheck_mode     nonactive                        Health Check Mode          True
max_transfer    0x40000                          Maximum TRANSFER Size      True
pvid            00c7cd9eabdeaf320000000000000000 Physical volume identifier False
queue_depth     3                                Queue DEPTH                False
reserve_policy  no_reserve                       Reserve Policy             True
```

Enable the health check mode for the disk to be able to receive a update of the status of the disks when one Virtual I/O Server is rebooted. Because the hcheck_interval attribute is set to 0, the client partition does not update the path mode when using the **lspath** command in case of a failure of the active path. To activate the health check function, use the **chdev** command, as shown in Example 4-22 on page 231. In this example, we use the health check interval of 60 seconds.

Note: The path switching also works even if the hcheck_interval attribute is set to 0, but it is useful to have the status updated automatically.

Example 4-22 Changing the health check interval

```
# chdev -l hdisk0 -a hcheck_interval=60 -P
hdisk0 changed
# lsattr -El hdisk0
PCM             PCM/friend/vscsi                 Path Control Module        False
algorithm       fail_over                        Algorithm                  True
hcheck_cmd      test_unit_rdy                    Health Check Command       True
hcheck_interval 60                               Health Check Interval      True
hcheck_mode     nonactive                        Health Check Mode          True
max_transfer    0x40000                          Maximum TRANSFER Size      True
pvid            00c7cd9eabdeaf320000000000000000 Physical volume identifier False
queue_depth     3                                Queue DEPTH                False
reserve_policy  no_reserve                       Reserve Policy             True
```

Note: MPIO on the client partition runs a fail_over algorithm. That means only one path is active at a time. If you shut down a Virtual I/O Server that serves the inactive path, then the path mode does not change to failed because no I/O is using this path.

Failover may require a small period of time in which the client re-establishes a path to the SAN. Testing of production workloads should be made in order to establish if this delay is suitable.

5. Set the path priority for this partition to have the active path going over VIO_Server2. The default setting is priority 1 on both paths, as shown in Example 4-23. In this case, you do not need a special path and the system will pick path0 as the active path. Priority 1 is the highest priority, and you can define a priority from 1 to 255.

Example 4-23 Show path priority using lspath

```
# lspath -AHE -l hdisk0 -p vscsi0
attribute value description user_settable

priority  1     Priority    True

# lspath -AHE -l hdisk0 -p vscsi1
attribute value description user_settable

priority  1     Priority    True
```

To determine which path will use the VIO_Server2 partition, issue the **lscfg** command and verify the slot number for the VSCSI devices, as shown in Example 4-24.

Example 4-24 Find out which parent belongs to which path

```
# lscfg -vl vscsi0
  vscsi0           U9117.570.107CD9E-V4-C10-T1  Virtual SCSI Client
Adapter

        Device Specific.(YL)........U9117.570.107CD9E-V4-C10-T1

# lscfg -vl vscsi1
  vscsi1           U9117.570.107CD9E-V4-C11-T1  Virtual SCSI Client
Adapter

        Device Specific.(YL)........U9117.570.107CD9E-V4-C11-T1
```

In our example, the odd slot numbers are configured to point to the VIO_Server2 partition. To set the active path to use the VSCSI1 device, the priority will remain 1, which is the highest priority. Change the path using VSCSI0 to priority 2, as shown in Example 4-25.

Example 4-25 Changing the priority of a path

```
# chpath -l hdisk0 -p vscsi0 -a priority=2
path Changed
```

Note: You may have to update the Preferred Path in the SAN to reflect the changed priority settings.

6. Reboot the client partition for the changes to take effect. Both changes require a reboot to be activated since this disk belongs to the rootvg and is in use.
7. Repeat these steps on the DB_server partition but with priority 2 for the path VSCSI1.

4.4.4 Concurrent disks in client partitions

Normally, a disk is assigned to one machine and this machine is responsible for handling the access to files or raw devices on that disk. Concurrent disks are disks that are accessed from more than one machine at the same time. This happens regularly with clustering software and application software that is designed to run on more than one machine in parallel. It is possible to implement concurrent virtual disks on one physical machine, using Virtual I/O Servers on

that machine. For redundancy reasons with clustering software, at least two machines are required.

A storage subsystem is required that can assign a disk to more than one machine. This can be some kind of Fibre Channel disk, for example, a DS4000 disk subsystem. Additionally, for smooth operation, we recommend that the Virtual I/O Servers are connected redundantly to the storage subsystem.

On the operating system, the following has to be installed:

- bos.clvm as the basic layer, to enable concurrent access.
- The cluster software that is responsible for managing concurrent disks.
- The application software that is responsible for handling the locking mechanism on files or raw devices on the disk.

To set up a concurrent disk, follow these steps:

1. Create the disk on the storage device.

   Follow the directions provided by the storage subsystem provider.

2. Assign the disk to two Virtual I/O Servers.

   How to do that depends on the type of storage subsystem. On a DS4000 type disk subsystem, for example, it would be assigned to a host group.

3. On the first Virtual I/O Server, scan for the newly assigned disk:

   ```
   $ cfgdev
   ```

4. On the first Virtual I/O Server, change the SCSI reservation of that disk to no_reserve so that the SCSI reservation bit on that disk is not set if the disk is accessed:

   ```
   $ chdev -dev hdiskN -attr reserve_policy=no_reserve
   ```

   (where *N* is the number of the disk in question; reservation commands are specific to the multipathing disk driver in use. This parameter is used with DS4000 disks, it can be different with other disk subsystems.)

5. On the first Virtual I/O Server, assign the disk to the first LPAR:

   ```
   $ mkvdev -vdev hdiskN -vadapter vhostN [ -dev XXX ]
   ```

   where *N* is the number of the disk respectively the vhost in question and the device name can be chosen to your liking but also left out entirely; the system will then create a name automatically.

6. On the first LPAR, scan for that disk:

   ```
   $ cfgdev
   ```

7. On the first LPAR, create an enhanced concurrent capable volume group:

   ```
   # mkvg -C -y myvg hdiskN
   ```

   The system will inform you that the volume group is not varied on automatically and that it has to be done manually. Although it is possible to create an enhanced concurrent capable volume group without cluster manager software installed, it is not recommend. All accesses to the enhanced concurrent enabled volume group are not coordinated between both servers, which can lead to data corruption on this volume group.

8. On the second Virtual I/O Server, scan for the disk:

   ```
   $ cfgdev
   ```

9. On the second Virtual I/O Server, change the SCSI reservation of that disk:

   ```
   $ chdev -dev hdiskN -attr reserve_policy=no_reserve
   ```

10. On the second Virtual I/O Server, assign the disk to the second LPAR

    ```
    $ mkvdev -vdev hdiskN -vadapter vhostN [ -dev XXX ]
    ```

11. On the second LPAR, import the volume group from the disk in learning mode:

    ```
    importvg -L -y myvg hdiskN
    ```

    where *N* is the number of the disk in question.

4.5 Scenario 4: Network Interface Backup in the client

Network Interface Backup, NIB can be used in a redundant network configuration when two Virtual I/O Servers are used. See Figure 4-10 on page 197.

NIB is configured in the client by using two virtual Ethernet adapters. These adapters must have different PVIDs corresponding to each of the Virtual I/O Servers.

In our configuration, we use PVID=1 for VIO_Server1 and PVID=2 for VIO_Server2.

We use the DB_server as an example where the Ethernet adapter ent0 has PVID=1 and Ethernet adapter ent1 has PVID=2. These are the steps to configure Network Interface Backup for the client:

1. Enter **smit** → **Devices** → **Communication** → **EtherChannel / IEEE 802.3ad Link Aggregation** → **Add An EtherChannel / Link Aggregation**. Select the primary adapter, ent0, and press Enter. Example 4-26 on page 235 shows the SMIT menu for adding an EtherChannel / Link Aggregation.

Example 4-26 Add An EtherChannel / Link Aggregation smit menu

```
Add An EtherChannel / Link Aggregation

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  EtherChannel / Link Aggregation Adapters            ent0                 +
  Enable Alternate Address                            no                   +
  Alternate Address                                  []                    +
  Enable Gigabit Ethernet Jumbo Frames                no                   +
  Mode                                                standard             +
  Hash Mode                                           default              +
  Backup Adapter                                      ent1                 +
       Automatically Recover to Main Channel          yes                  +
       Perform Lossless Failover After Ping Failure   yes                  +
  Internet Address to Ping                           [9.3.5.41]
  Number of Retries                                  []                    +#
  Retry Timeout (sec)                                []                    +#



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit           Enter=Do
```

We use ent1 as the backup adapter and the gateway 9.3.5.41 for the Internet address to ping.

2. Press Enter to complete. This gives ent2 as the aggregated adapter.
3. Define the IP interface on en2 and the setup is complete.

If the primary adapter cannot reach the gateway address, going through VIO_Server1 in our case, communication is switched to the backup adapter, which goes through VIO_Server2. To check which channel is active, you can use the **entstat** command. See Example 4-27.

Note: You will also receive an error message in the errorlog when a switch has taken place.

Example 4-27 CheckSwitching channels in a NIB configurationing which channel is active

```
# entstat -d ent2 | grep Active
Active channel: primary channel
```

A switch can be forced with the **/usr/lib/methods/ethchan_config -f** command (available since AIX 5L Version 5.3-ML03), as shown in Example 4-28.

Example 4-28 Switching channels in a NIB configuration

```
# /usr/lib/methods/ethchan_config -f ent2
# entstat -d ent2 | grep Active
Active channel: backup adapter
# /usr/lib/methods/ethchan_config -f ent2
# entstat -d ent2 | grep Active
Active channel: primary channel
```

Important: It is important to test communication through both channels.

In the SMIT menu, there is an option to **"**`Automatically Recover to Main Channel`**"**. It is set to Yes by default and this is the behavior when using physical adapters. However, virtual adapters do not adhere to this. Instead, the backup channel is used until it fails and then switches to the primary channel.

4.6 Initiating a Linux installation in a VIO client

The following steps describe how to boot the linuxlpar partition on SMS and install SUSE Linux Enterprise Server 9 on the assigned virtual SCSI disk:

1. Boot the linuxlpar partition using the CD or DVD drive, as you would normally do on an AIX 5L partition.
2. After booting, the first screen will be similar to the one in Example 4-29.
3. Type in `install` and then press Enter to proceed with the installation.

Example 4-29 Install SUSE Linux Enterprise Server 9 on the linuxlpar partition

```
Config file read, 148 bytes


  Welcome to SuSE Linux (SLES9)!

  Use  "install"     to boot the pSeries 64bit kernel
  Use  "install32"   to boot the 32bit RS/6000 kernel

  You can pass the option "noinitrd"  to skip the installer.
  Example: install noinitrd root=/dev/sda4

Welcome to yaboot version 1.3.11.SuSE
Enter "help" to get some basic usage information
```

Note: After typing in `install` and pressing Enter, you will be placed on Yaboot installer for SUSE Linux, where you can choose your installation options before proceeding with the actual install.

After successful installation, the Linux partition displays `SuSE Linux ppc64` in the operator window column of the HMC, as shown in Figure 4-23.

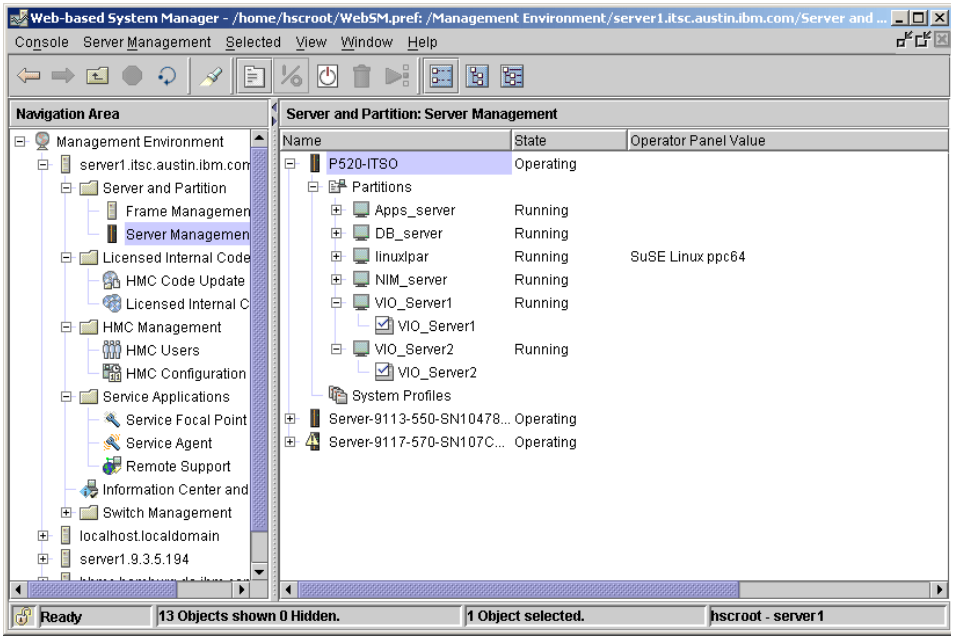


Figure 4-23 Installed SUSE partition

4.7 Supported configurations

This section discusses various supported configurations in a Virtual I/O Server environment. The following configurations are described:

- Supported VSCSI configurations Virtual I/O Servers with:
  - Mirrored VSCSI devices on the client and server
  - Multi-path configurations in a SAN environment
- Virtual Ethernet configurations
- Virtual devices and HACMP support
- Virtual devices and GPFS support

The configurations described in this section are not a complete list of all available supported configurations. It shows a collection of the most frequently adopted configurations that meet the requirements of most production environments.

For latest information about Virtual I/O Server supported environments, visit the following Web site:

http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html

4.7.1 Supported VSCSI configurations

This discussion on VSCSI configurations is a detailed one about supported and recommended configurations when using the Virtual I/O Server. It is a extension to the discussion in 4.1.1, “Providing higher serviceability with multiple Virtual I/O Servers” on page 182. An understanding of the principles of redundancy and availability is assumed.

Refer to 2.9, “Virtual SCSI introduction” on page 89 for considerations when using logical volumes on the Virtual I/O Server as virtual disks for client partitions.

Supported configurations with mirrored VSCSI devices

Figure 4-24 on page 239 shows a supported way for mirroring disks with only one Virtual I/O Server.

On the Virtual I/O Server, you either configure two logical volumes and map them to the vhost adapter assigned to the client partition or you directly map the hdisks to the vhost adapter. On the client partition, the mapped devices will appear as two disks. The client mirrors the two virtual disks using standard AIX 5L LVM mirroring.

In Figure 4-24 on page 239, disks on the Virtual I/O Server are attached to two separate physical adapters for higher availability. This scenario also works when the disks are attached to only one physical adapter, but in this case it does not protect against adapter failure.

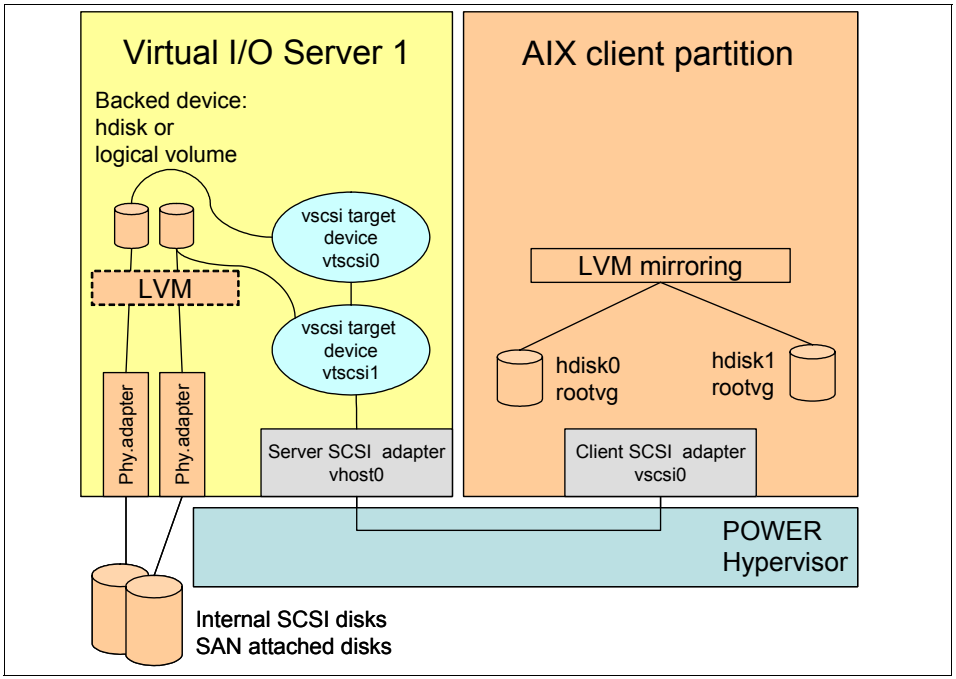


Figure 4-24 Supported and recommended ways to mirror virtual disks

Important: A logical volume of the Virtual I/O Server used as a virtual disk should not span multiple disks.

You can verify that a logical volume is confined to a single disk with the **`lslv -pv lvname`** command. The output of this command should only display a single disk.

We recommend using mirroring, striping, or concatenation of physical disks using the LVM in the VIO client, or to use such features of special RAID-capable host bus adapters or storage subsystems with the Virtual I/O Server. Thus, to provide redundancy for the backed disk, a hardware RAID 5 array on the Virtual I/O Server can be used. Figure 4-25 shows the Virtual I/O Server configured with a SCSI RAID adapter. At the time of writing, the following adapters are supported:

- PCI-X Dual Channel Ultra320 SCSI RAID Adapter (FC 5703)
- Dual Channel SCSI RAID Enablement Card (FC 5709)

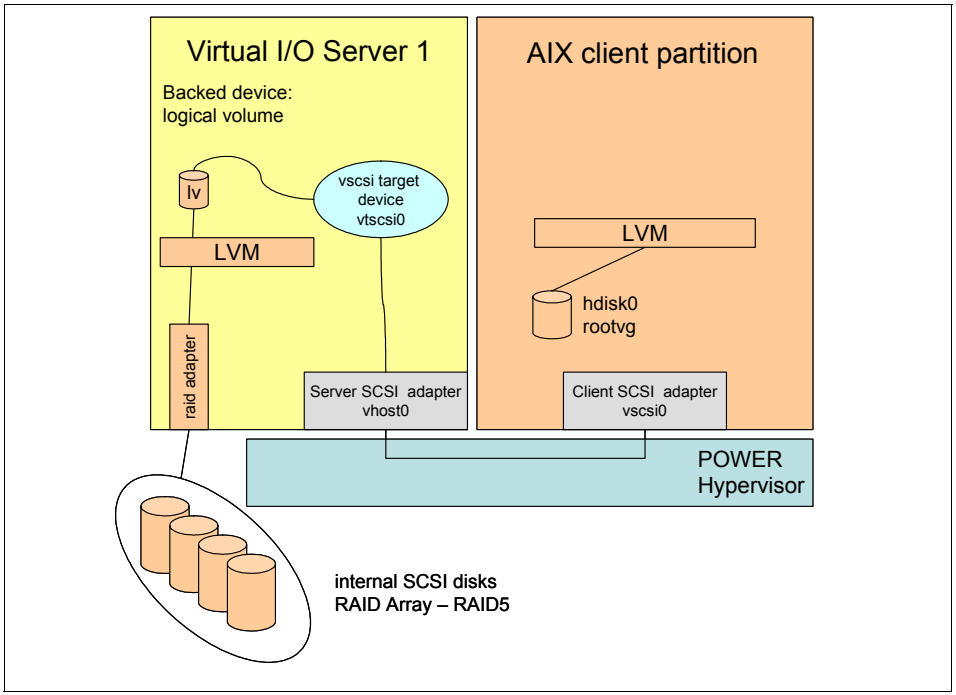


Figure 4-25 RAID5 configuration using a RAID adapter on the Virtual I/O Server

Attention: Only RAID hardware is supported for this configuration.

After creating a RAID 5 array, it will appear as one hdisk on the Virtual I/O Server. You can then divide the large disk into logical volumes and map them to your client partitions.

When using this configuration we recommend that you plan for two additional disks for the installation of the Virtual I/O Server that should be mirrored over two disks. Otherwise, the logical volumes you map to the client partition will be created in the rootvg of the Virtual I/O Server.

Important: Do not use the Virtual I/O Server rootvg for logical volumes that will be used as virtual disks for the clients.

When planning for multiple Virtual I/O Servers, Figure 4-26 shows the supported and recommended way for mirroring the virtual disks on the client partitions.

Note: One Virtual I/O Server is supported on the Integrated Virtualization Manager (IVM). Use the HMC to manage more than one Virtual I/O Server.

IBM supports up to ten Virtual I/O Servers within a single CEC managed by an HMC. Though architecturally up to 254 LPARS are supported, more than ten Virtual I/O Server LPARs within a single CEC has not been tested and therefore is not recommended.

Either configure a logical volume on each Virtual I/O Server and map it to the vhost adapter that is assigned to the same client partition or directly map a hdisk to the appropriate vhost adapter. The mapped logical volume or hdisk will be configured as a hdisk on the client side, each belonging to a different Virtual I/O Server. Then use LVM mirroring on the client site.

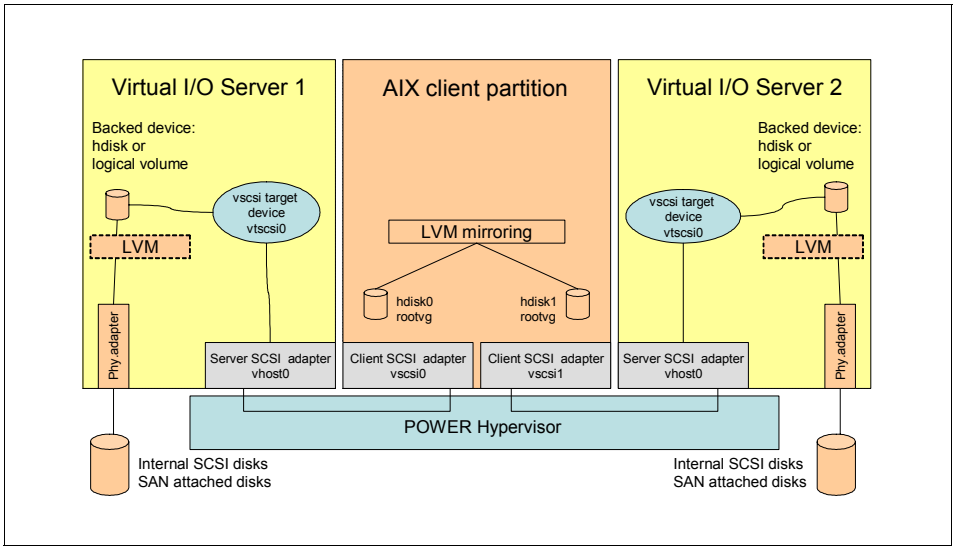


Figure 4-26 Recommended way to mirror virtual disks with two Virtual I/O Server

Supported multi-path configurations in a SAN environment

When discussing a supported configuration with MPIO, you need to distinguish two different scenarios:

- One Virtual I/O Server attaching a LUN in the SAN over more than one path. In this case, you only need to implement multi-path software on the Virtual I/O Server.
- Having multiple Virtual I/O Servers connect to the same LUN and backed up to the same client. In this case, the client partition uses MPIO to access the virtual disk as a single device. You may also consider using multi-path software on the Virtual I/O Server to access the LUN over more than one path for path failover and load balancing.

Note: This redbook only covers IBM storage solutions. For support statements of other storage vendors, contact your IBM representative or your storage vendor directly and ask for specifications on supported configurations.

Supported IBM TotalStorage solutions

At the time of writing, the TotalStorage Enterprise Server includes the following models:

- 2105 Enterprise Storage Server® (models 800, 750, and Fxx)
- 2107 Model 921 IBM TotalStorage DS8100
- 2107 Model 922 IBM TotalStorage DS8300
- 2107 Model 9A2 IBM TotalStorage DS8300
- 2107 Model 92E IBM TotalStorage DS8000 Expansion Unit
- 2107 Model 9AE IBM TotalStorage DS8000 Expansion Unit
- 1750 Model 511 IBM TotalStorage DS6800
- 1750 Model EX1 IBM TotalStorage DS6000 Expansion Unit

When attaching the Virtual I/O Server to a IBM TotalStorage DS Family, the RDAC driver is used on the Virtual I/O Server. At the time of writing, the IBM TotalStorage DS Family include the following models:

- DS4100 (FAStT100)
- DS4200 (FAStT200)
- DS4200 (FAStT600)
- DS4400 (FAStT700)
- DS4500 (FAStT900)
- FAStT500 Storage Server

For other supported IBM Storage solutions, such as the TotalStorage SAN Volume Controller, check the official IBM Web site:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/home.html

Using MPIO or other multi-path software

There are different multi-path software products supported on the Virtual I/O Server. The following example scenarios will point out which multi-path software is supported in the different configurations.

The Virtual I/O Server uses several methods to uniquely identify a disk for use as a virtual SCSI disk. They are:

- Unique device identifier (UDID), used by MPIO
- IEEE volume identifier, used by RDAC with the DS4000 family of products
- Physical volume identifier (PVID), used by other multi-path software

Most non-MPIO disk storage multipathing software products use the PVID method instead of the UDID method. Because of the different data format associated with the PVID method, clients with non-MPIO environments should be aware that certain future actions performed in the Virtual I/O Server partition may require data migration, that is, some type of backup and restore of the attached disks. These actions may include but are not limited to the following:

- Conversion from a Non-MPIO environment to MPIO
- Conversion from the PVID to the UDID method of disk identification
- Removal and rediscovery of the Disk Storage from ODM entries
- Updating non-MPIO multipathing software under certain circumstances
- Possible future enhancements to VIO

For all configuration, we strongly recommend using MPIO with the appropriate path control module to avoid this migration effort in the future.

Note: Use the **oem_setup_env** command for installing and configuring the multi-path environment on the Virtual I/O Server. All other configurations must be done from the **padmin** command line interface to avoid invalid configurations.

Supported scenarios using one Virtual I/O Server

Figure 4-27 represents a configuration with MPIO with only one Virtual I/O Server attached to IBM TotalStorage Enterprise Storage Systems. The LUN is connected over two Fibre Channel adapters to the Virtual I/O Server to increase redundancy or throughput.

Because the disk is only attached to one Virtual I/O Server, it is possible to create logical volumes and map them to the vhost adapter that is assigned to the appropriate client partition. You can also choose to map the disk directly to the vhost adapter.

For attaching the IBM TotalStorage Enterprise Storage Server to only one Virtual I/O Server, MPIO with the SDDPCM or SDD is supported.

Attention: The preferred method to attach IBM TotalStorage Enterprise Storage Systems is MPIO. Virtual disk devices created with SDD may require a migration effort in the future.

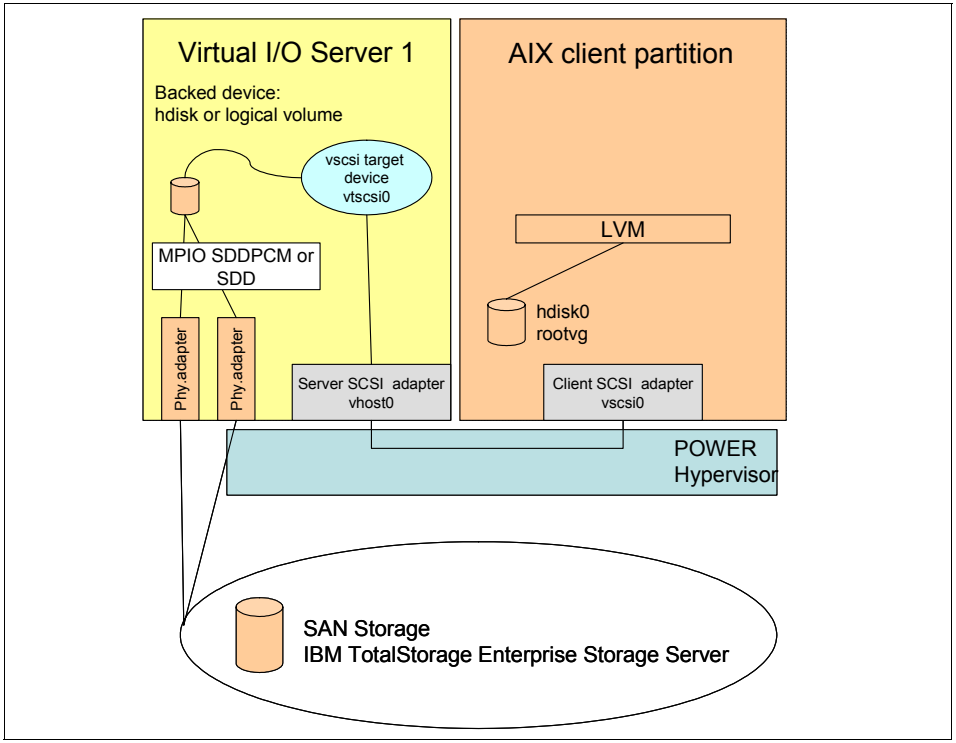


Figure 4-27 Using MPIO on the Virtual I/O Server with IBM TotalStorage

Figure 4-28 show the configuration with only one Virtual I/O Server for IBM TotalStorage DS Family using RDAC.

Only the RDAC driver is supported for attaching the IBM TotalStorage DS Family. Because the RDAC driver is using the IEEE volume identifier, there are no migration issues.

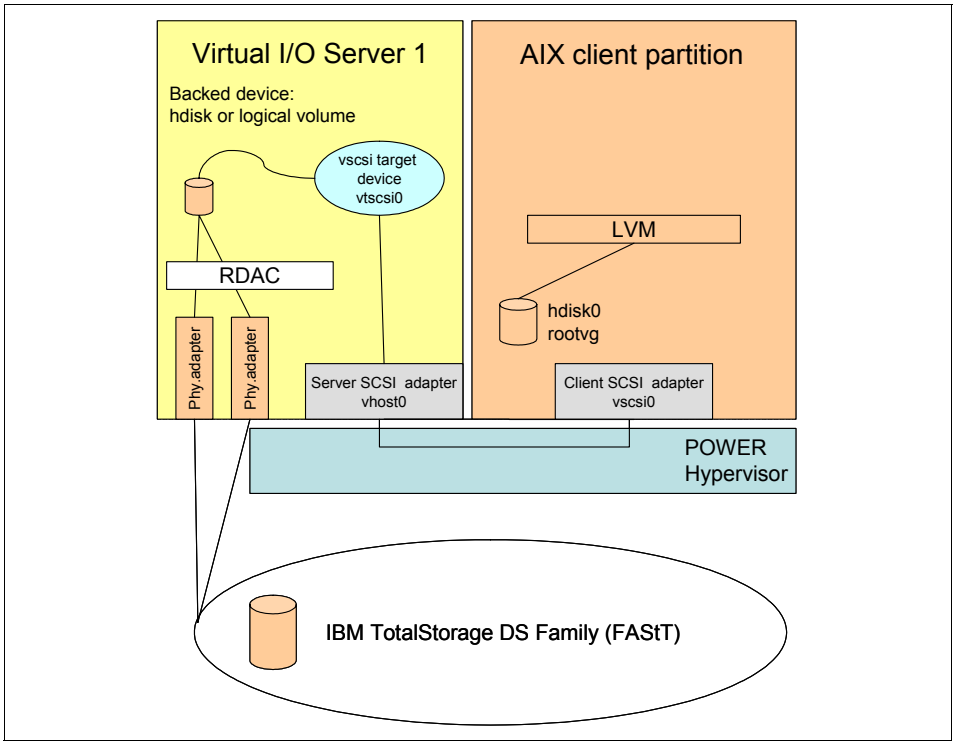


Figure 4-28 Using RDAC on the Virtual I/O Server with IBM TotalStorage

Supported scenarios using multiple Virtual I/O Servers

When configuring multiple Virtual I/O Servers, Figure 4-29 shows a supported configuration when attaching IBM TotalStorage Enterprise Storage Systems using multi-path software on the Virtual I/O Server for additional redundancy and throughput.

Attention: When using multiple Virtual I/O Servers, and exporting the same LUN to the client partitions, only mapping of hdisks is supported (logical volumes are not supported).

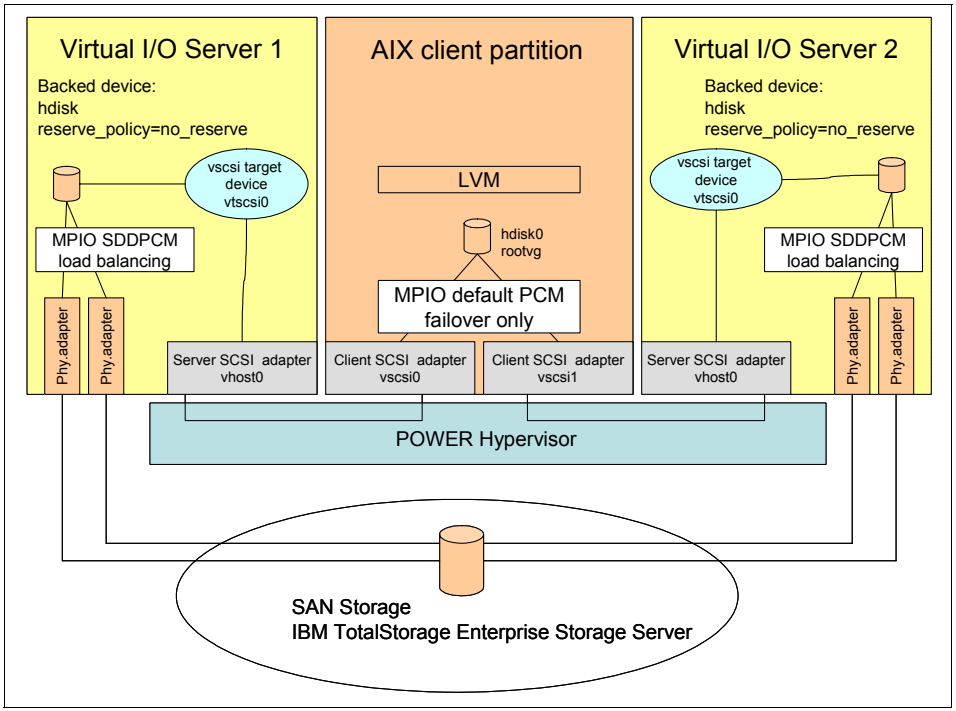


Figure 4-29 Configuration for multiple Virtual I/O Server and IBM ESS

Attaching IBM TotalStorage Enterprise Storage Servers to the Virtual I/O Servers only MPIO using the SDDPCM is supported.

With Fix Pack 6.2 or higher installed on the Virtual I/O Server, this configuration is also supported when attaching IBM TotalStorage DS Family using the RDAC driver, as shown in Figure 4-30 on page 247.

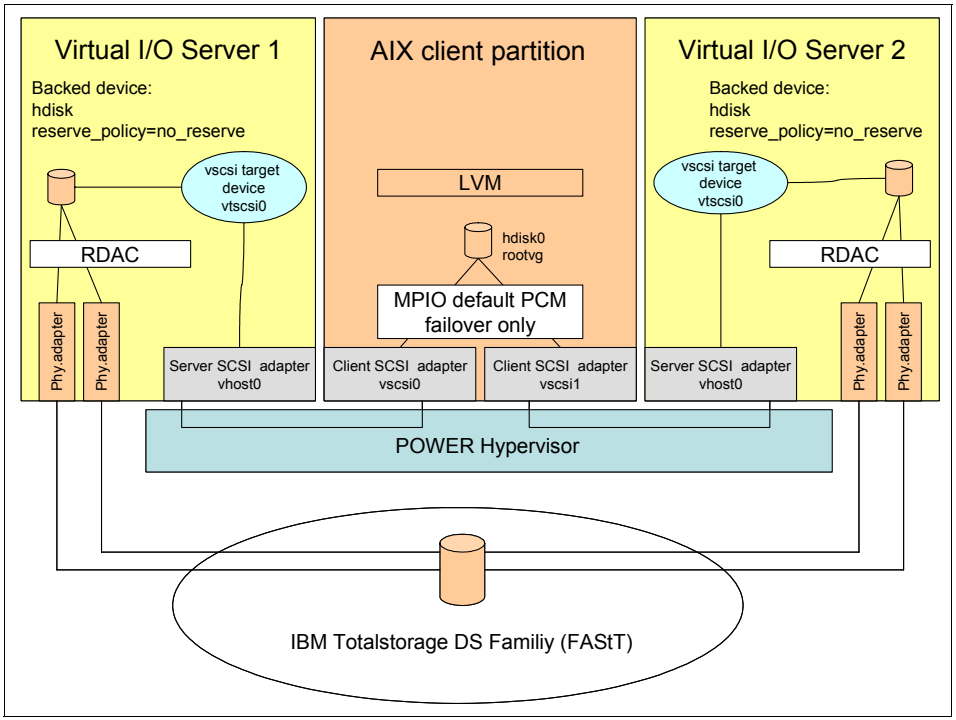


Figure 4-30 Configuration for multiple Virtual I/O Servers and IBM FAStT

On both Virtual I/O Servers, you have to set the hdisk attribute *reserve_policy* to no. This attribute prevents the Virtual I/O Server setting a reservation flag on the disk at the time of mapping. The MPIO component on the client partition will take the responsibility of managing the disk.

On the client partition MPIO, using the default PCM is supported, which only allows a failover policy and no load balancing. Only one path to the disks is active; the other path is used in the event that the active path fails, for example, when the Virtual I/O Server that serves the disk over the active path is rebooted.

It is possible to choose the active path on the client side. Users may manually configure the active paths for clients, enabling you to spread the workload evenly across the Virtual I/O Server. For detailed configuration steps, refer to 4.4, “Scenario 3: MPIO in the client with SAN” on page 218.

4.7.2 Supported Ethernet configurations

Shared Ethernet Adapter (SEA) failover was provided as a feature first in Virtual I/O Server Version 1.2. This allows two different configuration approaches when configuring high availability to external networks using SEA on multiple Virtual I/O Servers.

Figure 4-31 shows the supported configuration when using Network Interface Backup on the client partitions. This configuration is restricted to the use of PVIDs.

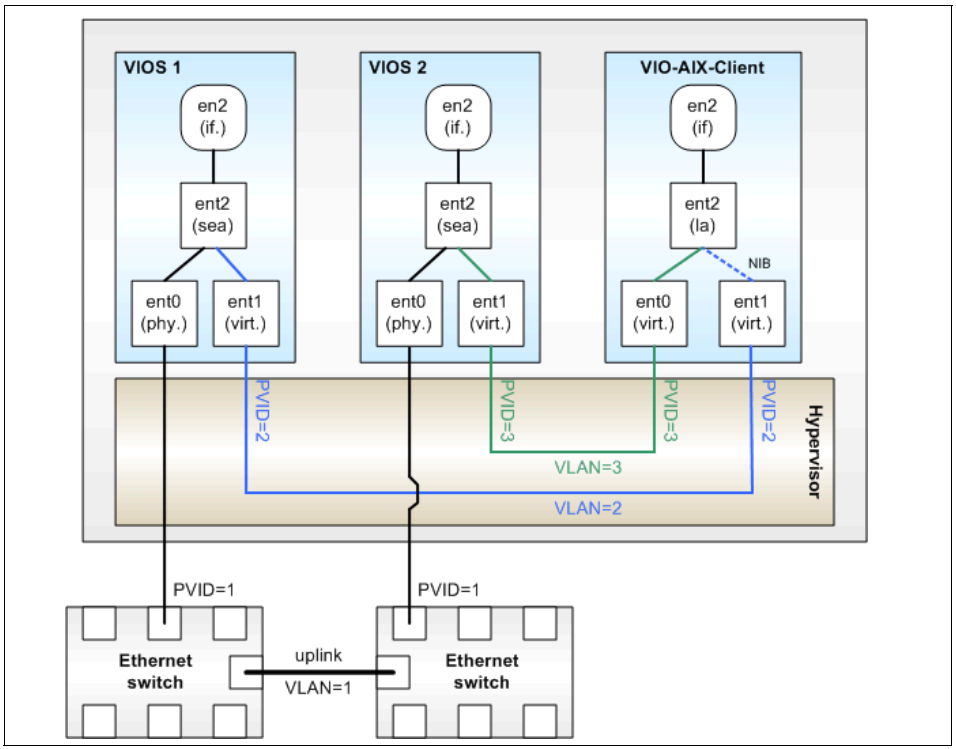


Figure 4-31 Network Interface Backup configuration

Figure 4-32 on page 249 shows the SEA Failover feature. This configuration also allows you to configure highly available external access to the network when using VLAN tagging, and simplifies the configuration on the client partition.

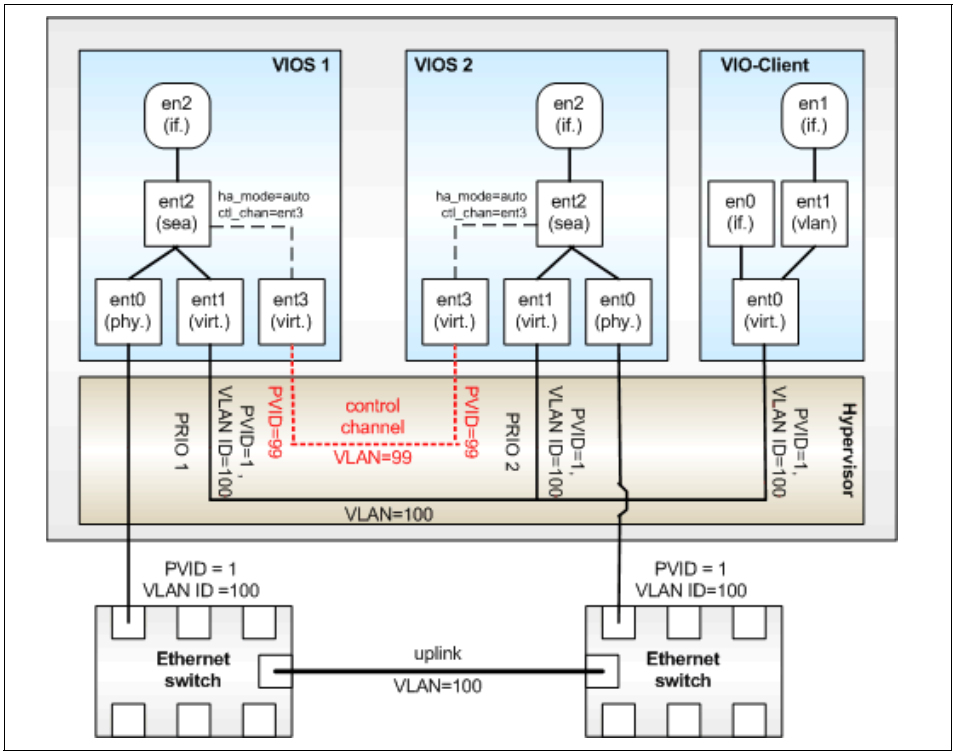


Figure 4-32 SEA Failover configuration

For a detailed discussion of the two configurations, refer to 4.1.3, “High availability for communication with external networks” on page 189. For other supported configurations on virtual Ethernet and SEA, refer to the InfoCenter documentation.

4.7.3 HACMP for virtual I/O clients

The IBM HACMP software provides a computing environment that ensures that mission-critical applications running in AIX 5L stand-alone servers or partitions can recover quickly from hardware and software failures. The HACMP software is a high availability system that ensures that critical resources for an application are available for processing in a cluster of servers or partitions. High availability combines custom software with hardware to minimize downtime by quickly restoring services when a system, component, or application fails.

In order to help clients improve the availability for their virtualized servers, IBM tested and certified HACMP with the Advanced POWER Virtualization features of virtual SCSI and virtual Ethernet. Clients can design highly available clusters of dedicated-processor partitions or micro-partitions that use services from Virtual I/O Servers for their critical applications. Information about the announcement of support for HACMP for the Advanced POWER Virtualization feature can be found at:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10390

The tasks related to maintaining active cluster storage resources in each of the Virtual I/O Servers that serve AIX 5L client partitions (cluster nodes) are handled by the combination of certified software levels and the configuration of disk drives and volume groups. The most important considerations include:

- The external storage content (volume groups and logical volumes) is handled at the AIX 5L client partition level. Virtual I/O Servers only connect external storage with HACMP/AIX 5L nodes (client partitions).
- No disk reservations apply at the hardware level. Storage access, depending on the configuration, is a combination of HACMP/AIX 5L LVM.
- All the volume groups must be Enhanced Concurrent volume groups regardless of whether they are used in concurrent access mode or not.
- If any HACMP/AIX 5L nodes access the volume groups through a Virtual I/O Server, then all nodes must access through the Virtual I/O Server.
- All the HACMP/AIX 5L nodes must use the volume groups with the same basis, concurrent or not concurrent.

Figure 4-33 on page 251 shows the basic issues for storage of AIX 5L client partitions and HACMP.

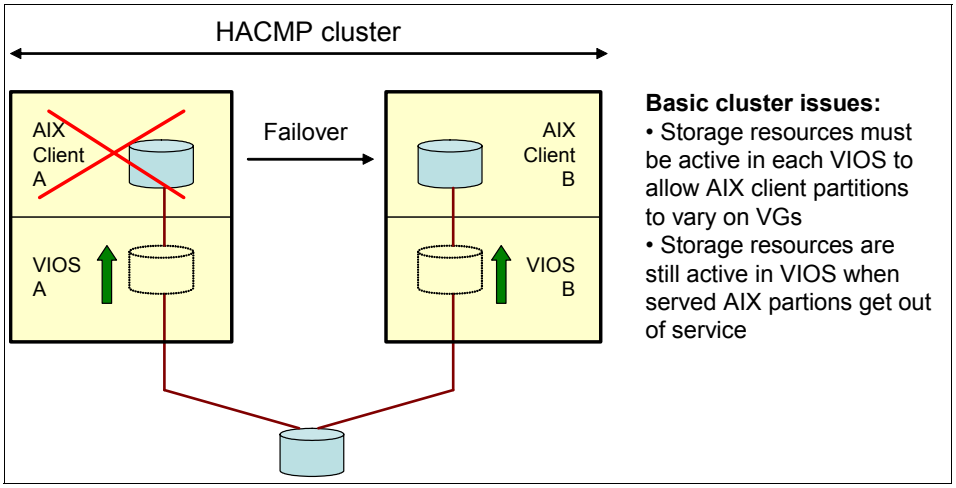


Figure 4-33 Basic issues for storage of AIX 5L client partitions and HACMP

This section outlines the supported configurations for using HACMP with the virtual Ethernet and virtual SCSI features. Such configurations should be accomplished with the minimum levels of software and specific virtual Ethernet and virtual SCSI attached disk configurations.

Software requirements

The levels of software needed in both Virtual I/O Server and AIX 5L client partitions are shown in Table 4-7.

Table 4-7 Minimum levels of software to configure HACMP with APV

Software	Version	Maintenance level	APARs/Fixes
AIX 5L V5.3	5.3	5300-02	IY70082, IY72974
rsct.basic.hacmp fileset	2.4.2.1		
rsct.basic.rte fileset	2.4.2.2		
rsct.compat.basic.hacmp fileset	2.4.2.0		
HACMP	5.1		IY66556
HACMP	5.2		IY68370, IY68387
HACMP	5.3		
Virtual I/O Server	1.1	Fix Pack 6.2	IY71303
Virtual I/O Server	1.2		
Virtual I/O Server	1.3		

Clients can find the fixes and filesets for AIX 5L and HACMP at:

http://techsupport.services.ibm.com/server/vios/download/home.html

The fixpack for the Virtual I/O Server can be found at:

http://www.ibm.com/servers/eserver/support/pseries/aixfixes.html

HACMP and virtual SCSI

The volume group must be defined as Enhanced Concurrent Mode. In general, Enhanced Concurrent Mode is the recommended mode for sharing volume groups in HACMP clusters because volumes are accessible by multiple HACMP nodes, resulting in faster failover in the event of a node failure.

If file systems are used on the standby nodes, they are not mounted until the point of failover, because the volume groups are in full active read/write mode only on the home node; the standby nodes have the volume groups in passive mode, which does not allow access to the logical volumes or file systems. If shared volumes (raw logical volumes) are accessed directly in Enhanced Concurrent Mode, these volumes are accessible from multiple nodes, so access must be controlled at a higher layer, such as databases.

If any cluster node accesses shared volumes through virtual SCSI, all nodes must do so. This means that disks cannot be shared between a partition using virtual SCSI and a server or partition directly accessing those disks.

All volume group construction and maintenance on these shared disks is done from the HACMP nodes using C-SPOC, not from the Virtual I/O Server.

Note: Partitions using virtual I/O cannot be mixed in an HACMP cluster with partitions using dedicated adapters accessing the same shared disks.

HACMP and virtual Ethernet

IP Address Takeover (IPAT) through aliasing must be used. IPAT using replacement and MAC Address Takeover are not supported. In general, IPAT using aliasing is recommended for all HACMP networks that can support it.

All virtual Ethernet interfaces defined to HACMP should be treated as single-adapter networks. In particular, configure the file netmon.cf to include a list of clients to ping. It must be used to monitor and detect failure of the network interfaces. Due to nature of virtual Ethernet, other mechanisms to detect the failure of network interfaces are not effective.

If the Virtual I/O Server has only a single physical interface on a network (instead of, for example, two interfaces with Ethernet aggregation), then a failure of that

physical interface will be detected by HACMP on the AIX 5L client partition. However, that failure will isolate the node from the network. So we recommend, in this case, a second virtual Ethernet on the AIX 5L client partition.

There are several ways to configure AIX 5L client partitions and Virtual I/O Server resources for additional high availability with HACMP. We recommend using two Virtual I/O Servers. This configuration also allows online service for Virtual I/O with improved uptime for user applications. In Figure 4-34, there is an example of a HACMP cluster between two AIX 5L client partitions.

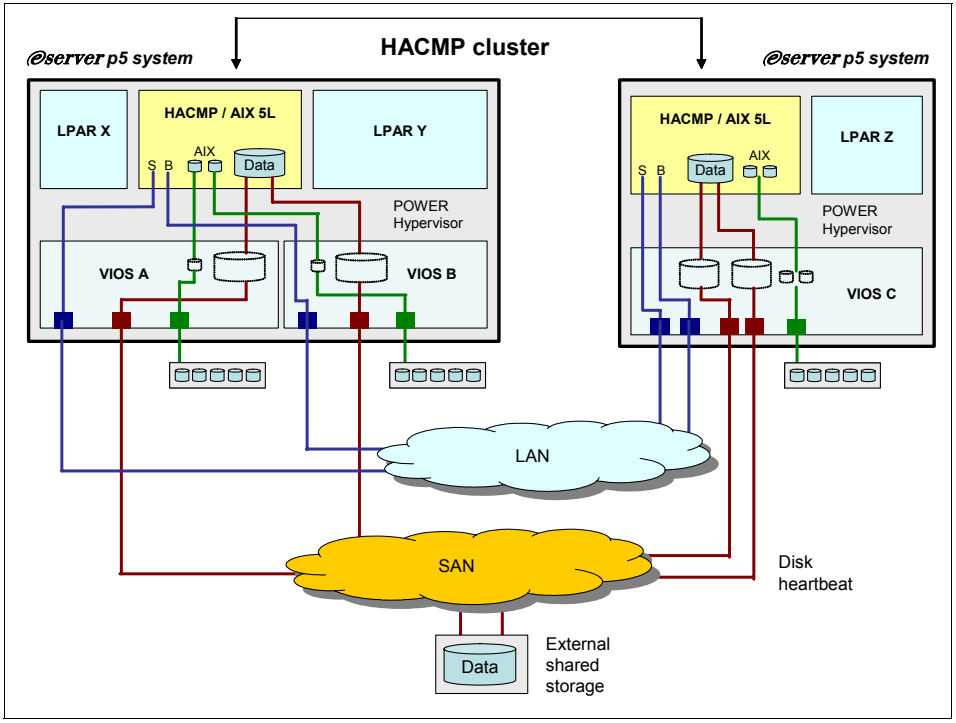


Figure 4-34 Example of HACMP cluster between two AIX 5L client partitions

Figure 4-35 shows the basic considerations when configuring an AIX 5L client partition with HACMP to be part of a high availability cluster of nodes using virtual Ethernet and virtual SCSI services from two Virtual I/O Servers.

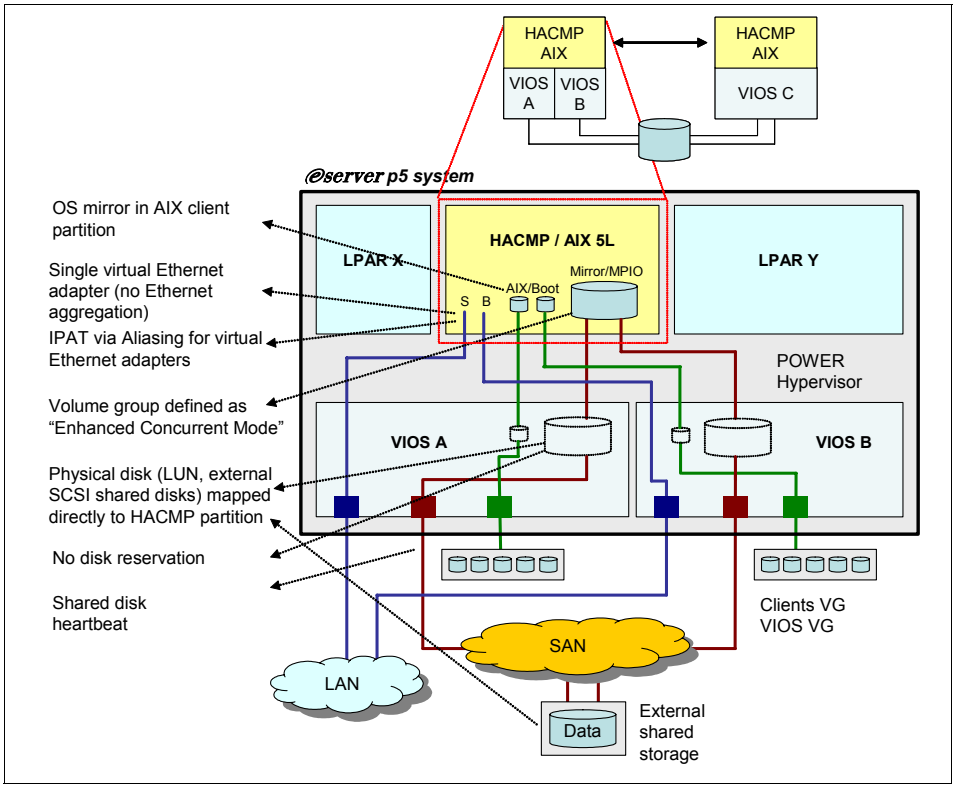


Figure 4-35 Example of an AIX 5L client partition with HACMP using two VIOSs

The following publications can be useful for clients planning to configure HACMP clusters with AIX 5L client partitions:

- *High Availability ClusterMulti-Processing for AIX: Concepts and Facilities Guide*, SC23-4864
- *High Availability ClusterMulti-Processing for AIX: Planning and Installation Guide*, SC23-4861
- *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, SG24-6769

4.7.4 General Parallel Filesystem (GPFS)

General Parallel Filesystem (GPFS) is currently not supported with virtual Ethernet or virtual SCSI disks on AIX 5L or Linux. Check the GPFS FAQ for updates:

http://publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

5

System management

This chapter provides a discussion of the following topics:

- Dynamic LPAR operations
- Backup and restore of the Virtual I/O Server
- Rebuilding the Virtual I/O Server, if no restore is possible
- System maintenance for the Virtual I/O Server
- Monitoring a virtualized environment
- Sizing considerations for Virtual I/O Servers
- Security considerations for Virtual I/O Servers

5.1 Dynamic LPAR operations

This section discusses how to move resources dynamically, which may be useful when maintaining your virtualized environment. We will look here at following operations:

- Addition of resources
- Movement of adapters between partitions
- Removal of resources
- Replacement of resources

5.1.1 Add adapters dynamically

The following steps show one way to add adapters dynamically:

1. Repeat step 1 on page 268, but this time choose **Virtual Adapter Resources** and then the **Add/Remove** button.
2. The next window will look like the one in Figure 5-1. Click the **Create client adapter** button.

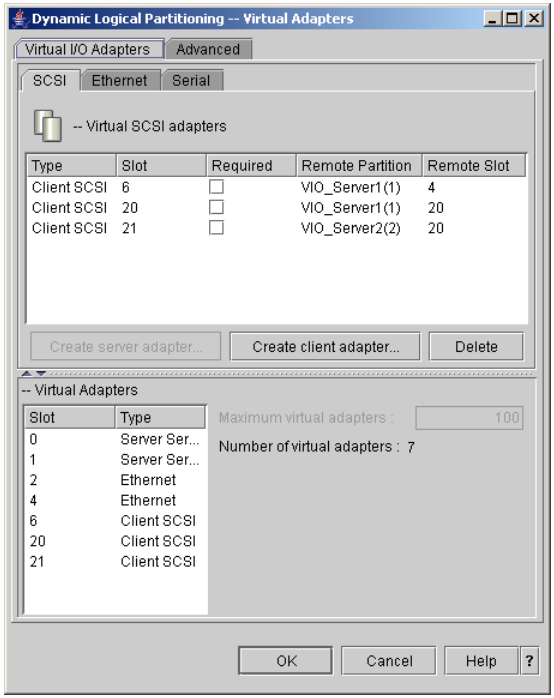


Figure 5-1 Dynamically adding virtual adapters window

3. Figure 5-2 shows you the window after clicking **Create client adapter**. Put in the slot number for the client adapter in the upper left hand portion of the window. Under the Connection settings box, choose the Virtual I/O Server partition using the drop-down arrow key and input the corresponding Virtual I/O Server slot number for the server adapter.

   Note: Step 3 assumes you already have a virtual SCSI server adapter available and you know the slot number on the server side. Then you can go ahead and click the **OK** button to create the client adapter. If not, proceed with step 4 and create the server SCSI adapter dynamically.

4. In the Virtual I/O Server dialog box (Figure 5-2), choose the Virtual I/O Server where you want to create the adapter, highlight it, and click **Create server adapter**.

   Note: The **Create server adapter** button on the lower right hand side of the window is a new feature on the HMC that allows you to dynamically create the virtual SCSI adapter on the Virtual I/O Server server. Be aware that when doing dynamic LPAR operations and clicking the **Create server adapter** button, the operation will add the adapter dynamically on the partition, but not update the partition profile. You need to make sure to edit the partition profile if you want the change to be persistent after a reboot. To ease this task, have a look at the instructions provided in Figure 3-40 on page 162.

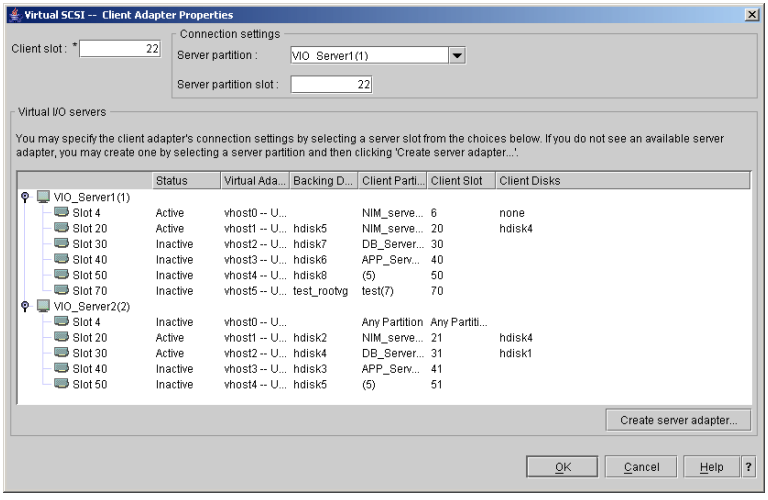


Figure 5-2 Virtual SCSI client adapter properties window

5. Figure 5-3 shows you the Virtual adapters window.

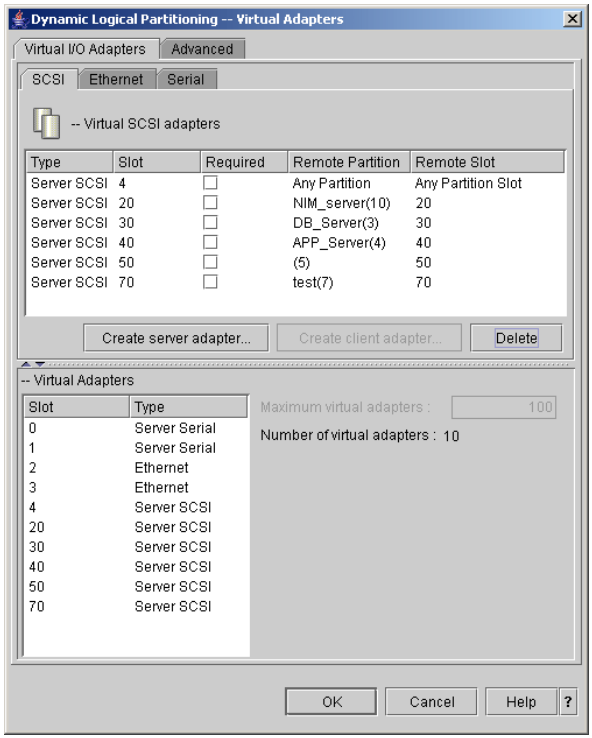


Figure 5-3 Dynamically create server adapter window

6. Click **Create server adapter** and put in the appropriate slot number.

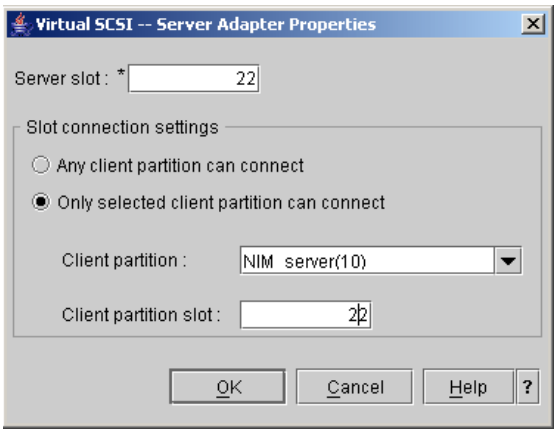


Figure 5-4 Server Adapter Properties

7. The results of this operation are shown in Figure 5-5.

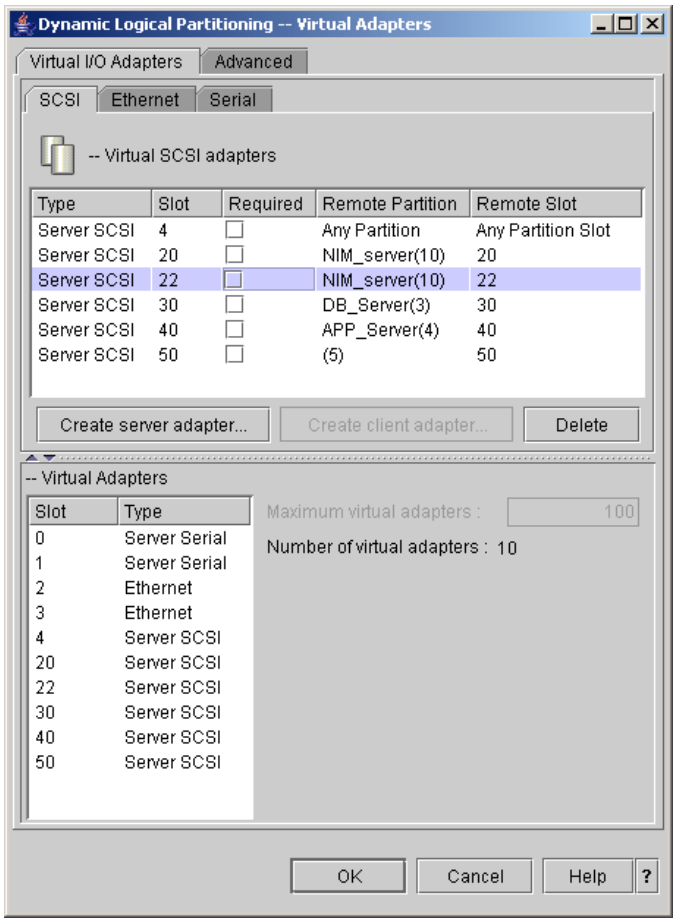


Figure 5-5 Window after creating the server virtual SCSI adapter

8. Click **OK** when done.

5.1.2 Move adapters dynamically in AIX 5L

In order to move a physical adapter, you have to release the adapter in the partition that currently owns it. Use the HMC to list which partition owns the adapter. (Right-click **Managed System** and select **Properties** → **I/O**.)

Usually, devices such as an optical drive belong to the adapter to be moved and they should be removed as well. The optical drive often needs to be moved to

another partition. Use the **lsslot -c slot** command to list adapters and their members. In the Virtual I/O Server, you can use the **lsdev -slots** command.

Use the **rmdev -l pci*n* -d -R** command to remove the adapter from the partition. In the Virtual I/O Server, you can use the **rmdev -dev pci*n* -recursive** command (n is the adapter number).

Example 5-1 shows how to remove the PCI adapter and the CD or DVD drive from a partition.

Example 5-1 Removing the adapter

```
# lsslot -c slot
Slot                     Description       Device(s)
U787B.001.DNW108F-P1-T14  Logical I/O Slot  pci3 sisioa0
U787B.001.DNW108F-P1-T16  Logical I/O Slot  pci2 ide0
U9113.550.105E9DE-V10-C0  Virtual I/O Slot  vsa0
U9113.550.105E9DE-V10-C2  Virtual I/O Slot  ent0

# rmdev -l pci2 -d -R
cd0 deleted
ide0 deleted
pci2 deleted
```

The adapter is then ready to be moved to another partition by using the HMC.

1. Right-click the partition that currently holds the adapter and select **Dynamic Logical Partitioning** → **Physical Adapter Resources** → **Move** (see Figure 5-6).

   The adapter must not be set as required in the profile. To change the setting from required to desired, you have to update the profile, and stop and start the LPAR (not just reboot).

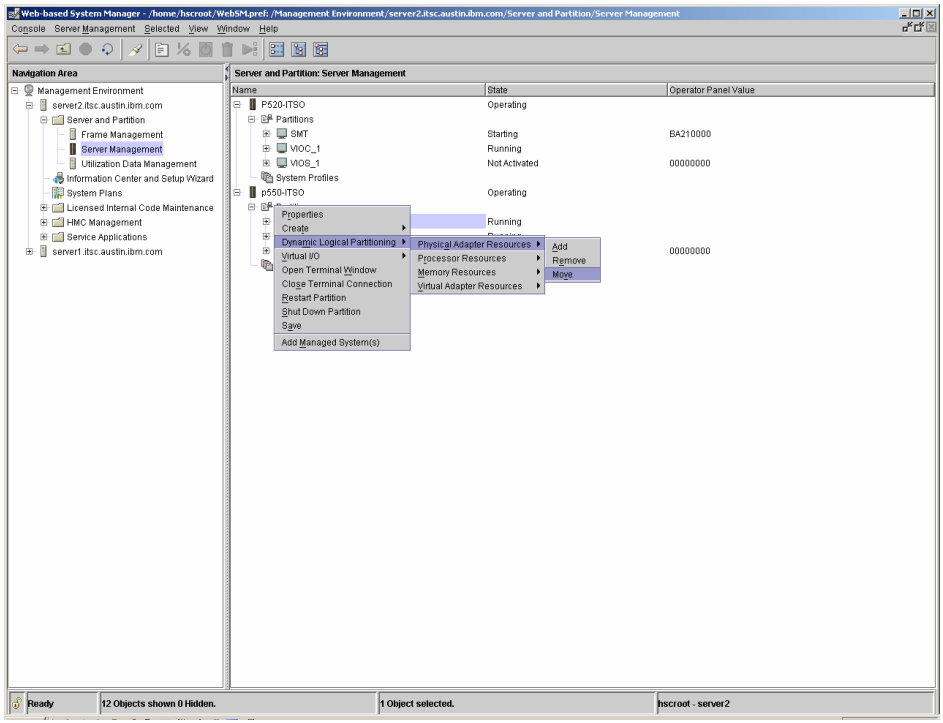


Figure 5-6 Dynamic LPAR physical adapter operation

2. Select the adapter to be moved and the receiving partition, as shown in Figure 5-7.

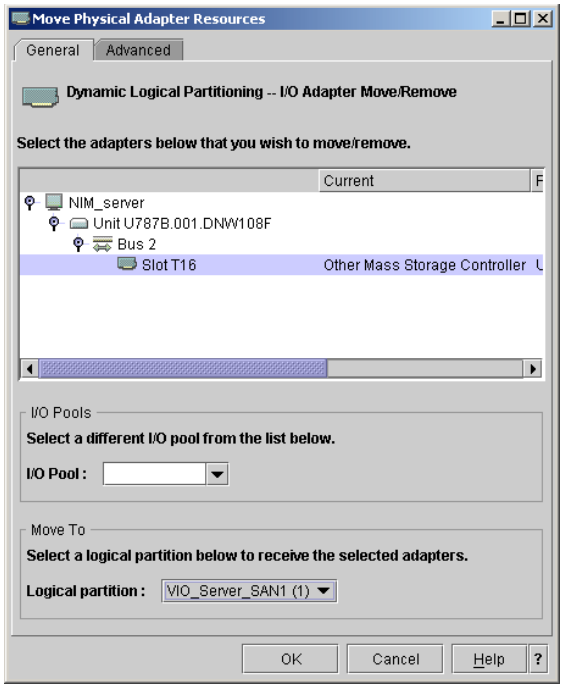


Figure 5-7 Selecting adapter T16 to be moved to partition VIO_Server_SAN1

3. Click **OK** to execute.
4. Type **cfgmgr** (**cfgdev** in the Virtual I/O Server) in the receiving partition to make the adapter and its devices available.

5. Remember to update the profiles of both partitions for the change to be reflected across restarts of the partitions. Alternatively, use the **Save** option to save the changes to a new profile. See Figure 5-8.

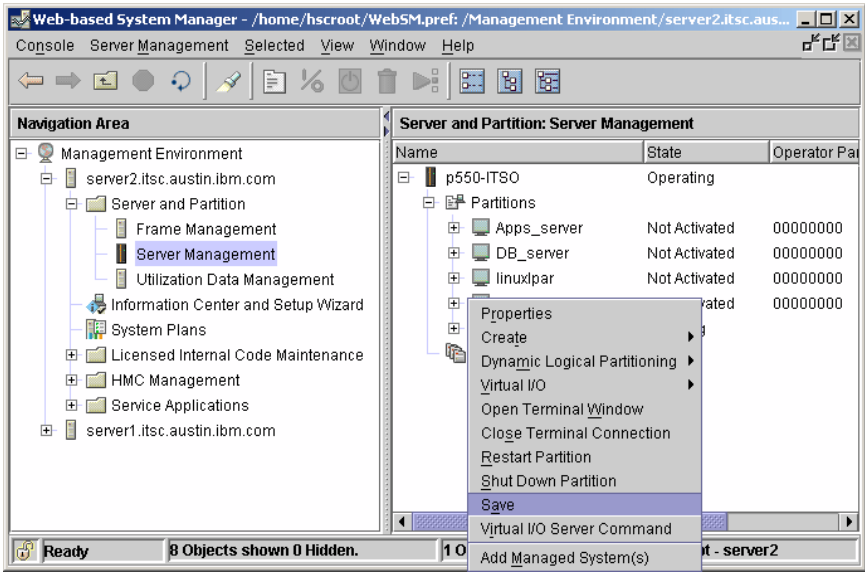


Figure 5-8 Save profile

5.1.3 Add memory dynamically in AIX 5L

Follow the steps below to dynamically add additional memory to the logical partition:

1. Right click the logical partition and select **Dynamic Logical Partitioning** → **Memory resources** → **Add** (see Figure 5-9).

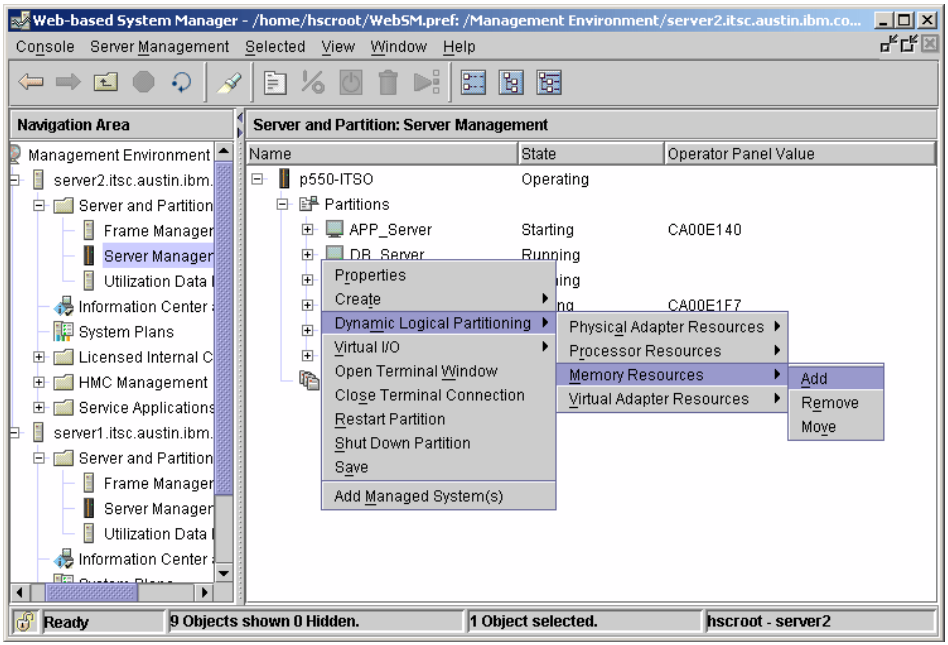


Figure 5-9 Dynamic LPAR memory operation

2. Choose the memory settings you want to add (see Figure 5-10).

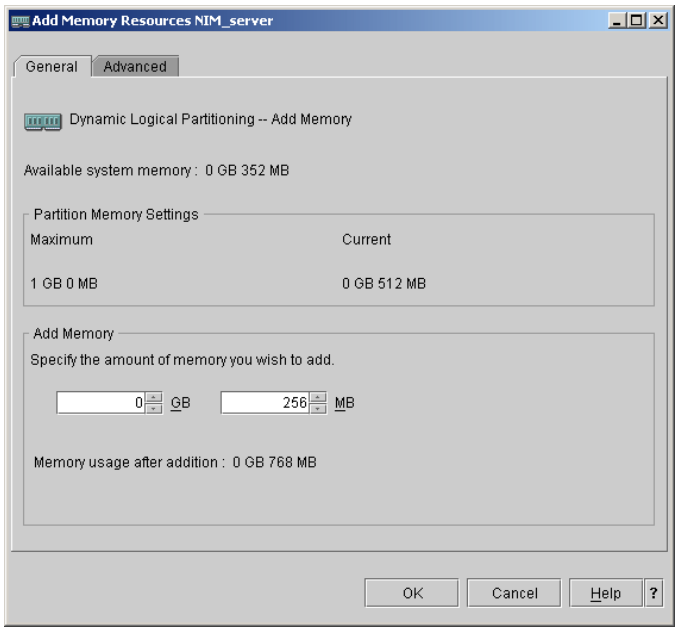


Figure 5-10 Additional 256 MB memory to be added dynamically

3. Click **OK** when done. A status window (Figure 5-11) is displayed.

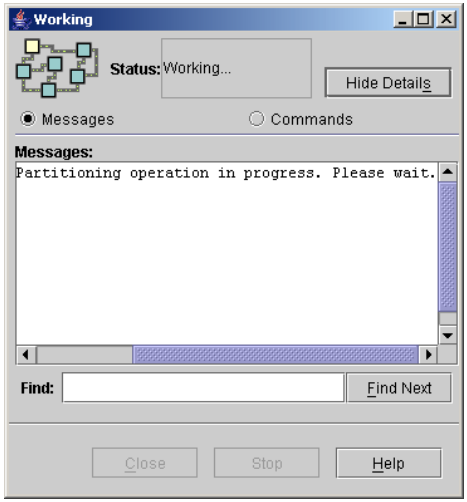


Figure 5-11 Dynamic LPAR operation in progress

5.1.4 Removing memory dynamically

The following steps provide a way to remove memory from a logical partition dynamically:

1. Right-click the logical partition where you want to initiate a dynamic LPAR operation. The first window in any dynamic LPAR operation will be similar to Figure 5-12.

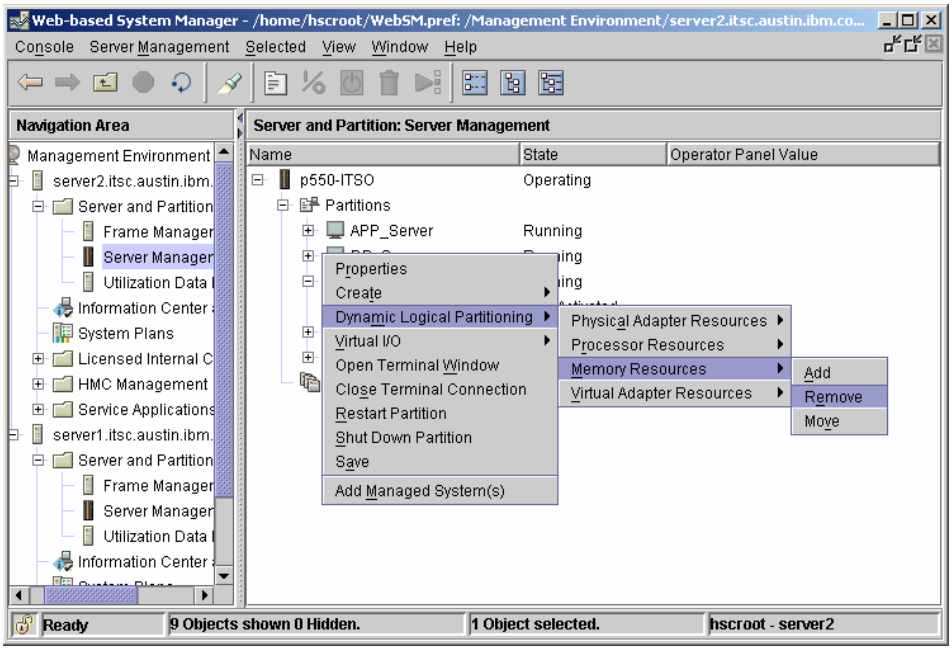


Figure 5-12 Initial dynamic LPAR window

The memory settings before the operation are:

```
# lsattr -El mem0
goodsize 512 Amount of usable physical memory in Mbytes False
size     512 Total amount of physical memory in Mbytes  False
```

Figure 5-13 shows the tab to decrease memory. Make the required changes, as indicated in the figure.

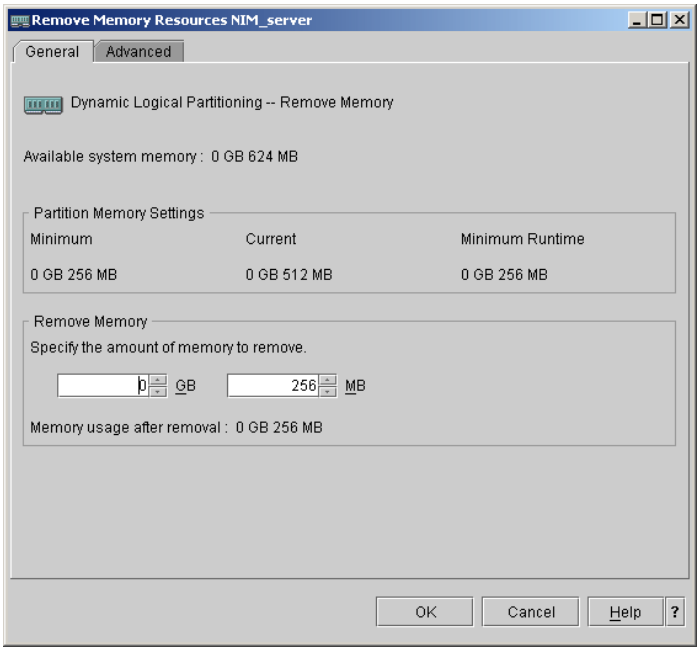


Figure 5-13 Dynamic removal of 256 MB memory

2. Click **OK** when done. A status window is shown (Figure 5-14).

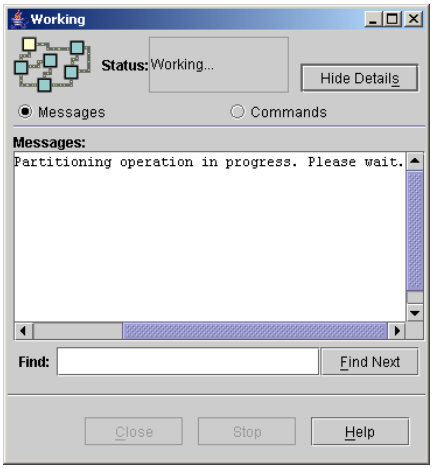


Figure 5-14 Status window

The following command shows the effect of the memory deletion:

```
# lsattr -El mem0
goodsize 256 Amount of usable physical memory in Mbytes False
size     256 Total amount of physical memory in Mbytes  False
```

5.1.5 Removing virtual adapters dynamically

The following steps provide a way to remove virtual adapters from a partition dynamically:

1. Repeat step 1 on page 268, but this time choose **Virtual Adapter Resources** and then the **Add/Remove** button.
2. Choose the adapter you want to delete (Figure 5-15) and click the **Delete** button.

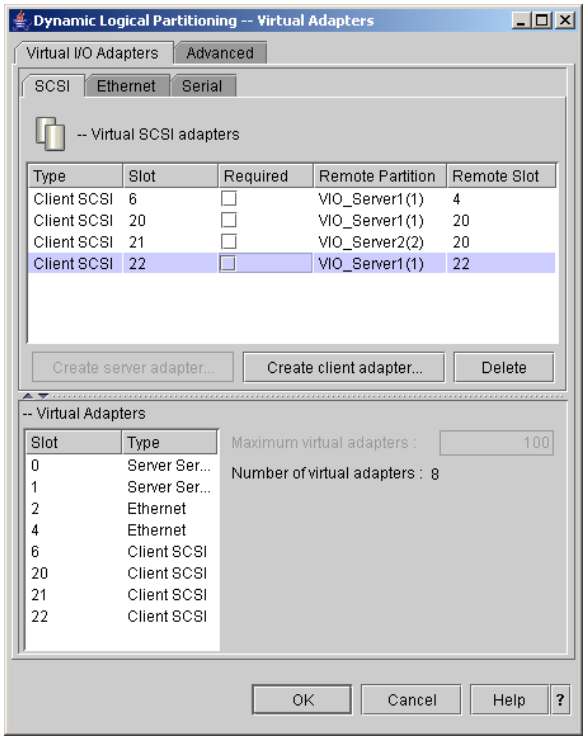


Figure 5-15 Dynamic LPAR virtual adapters window

3. Click **OK** when done.

5.1.6 Removing processors dynamically

The following steps show one way to remove processors dynamically:

1. Right-click the logical partition where you want to initiate a dynamic LPAR operation, as shown in Figure 5-16.

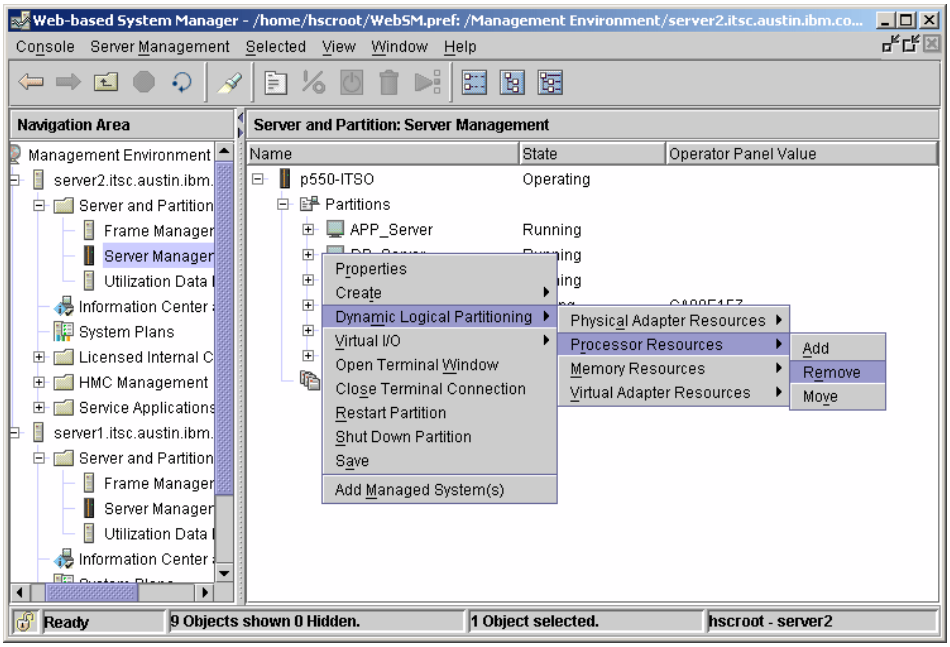


Figure 5-16 Dynamic LPAR operation CPU processing units

2. Figure 5-17 shows the current processing units and removing the 0.1 processing unit.

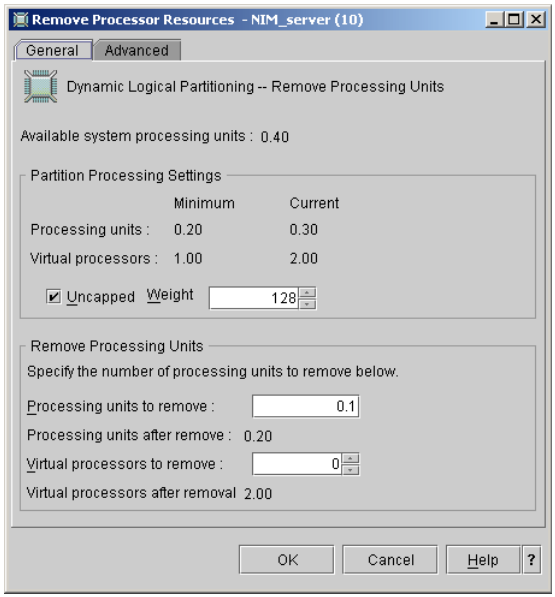


Figure 5-17 Dynamic LPAR operation to remove 0.1 processing unit

3. Click **OK** when done.

5.1.7 Removing or replacing a PCI Hot Plug adapter

Removing a PCI Hot Plug adapter is necessary when this adapter has to be replaced. Replacing an adapter could happen when, for example, you exchange 2 Gb Fibre Channel adapters for a 4 Gb Fibre Channel adapter, or other for configuration changes or updates.

Hot plugging adapters while Virtual I/O clients are using virtual devices requires, for disks, that MPIO is active and functional on the virtual device. For virtual Ethernet adapters in the Virtual I/O client, redundancy has to be enabled, either through Shared Ethernet Adapter failover enabled in the Virtual I/O Servers or Network Interface Backup configured if continuous network connectivity is required. If there is no redundancy for Ethernet, the replace operation can be done while the Virtual I/O Client is still running, but it will lose network connectivity during replacement. For Virtual I/O Clients that have no redundant paths to their virtual disks and are not mirroring these disks, it is necessary to shut them down for the time used to replace the adapter.

On the Virtual I/O Server, in both cases there will be child devices connected to the adapter as the adapter would be in use before. Therefore, the child devices will have to be unconfigured as well as the adapter before the adapter can be removed or replaced. Normally there is no need to remove the child devices as, for example, disks and mapped disks, also known as Virtual Target Devices in the case of a Fibre Channel adapter, but they have to be unconfigured (set to the defined state) before the adapter they rely on can be replaced.

5.4.2, “Hot pluggable devices” on page 305 has additional information about hot-pluggable devices.

5.1.8 Replacing an Ethernet adapter on the Virtual I/O Server

You can do the replace and remove functions by using the **diagmenu** command and selecting **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Replace/Remove a PCI Hot Plug Adapter**.

If there are still devices connected to the adapter and a replace or remove operation is performed on that device, there will be an error message in **diagmenu**:

```
The specified slot contains device(s) that are currently
configured. Unconfigure the following device(s) and try again.

ent0
```

These messages mean that devices dependent on this adapter have to be unconfigured first. We will now replace a single physical Ethernet adapter that is part of a Shared Ethernet Adapter. Here are the steps to do it:

1. Use the **diagm**enu command to unconfigure the Shared Ethernet Adapter and then select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Unconfigure a device**. You should get the following output:

```
Unconfigure a Device

Device Name                                    ¦

Move cursor to desired item and press Enter. Use arrow keys to scroll. ¦
[MORE...12] ¦
en5         Defined            Standard Ethernet Network Interface     ¦
ent0        Available 01-08    2-Port 10/100/1000 Base-TX PCI-X Adapte ¦
ent1        Available 01-09    2-Port 10/100/1000 Base-TX PCI-X Adapte ¦
ent2        Available          Virtual I/O Ethernet Adapter (l-lan)    ¦
ent3        Available          Virtual I/O Ethernet Adapter (l-lan)    ¦
ent4        Defined            Virtual I/O Ethernet Adapter (l-lan)    ¦
ent5        Available          Shared Ethernet Adapter                 ¦
et0         Defined   01-08    IEEE 802.3 Ethernet Network Interface   ¦
```

```
et1         Defined   01-09    IEEE 802.3 Ethernet Network Interface   ¦
et2         Defined            IEEE 802.3 Ethernet Network Interface   ¦
[MORE...50]                                                             ¦
```

Select the Shared Ethernet Adapter (in this example, ent5), and in the following dialogue choose to keep the information in the database:

```
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[Entry Fields]
* Device Name                                            [ent5]
  Unconfigure any Child Devices                          no
  KEEP definition in database                            yes
```

Press **Enter** to accept the changes. The system will show that the adapter is now defined:

```
ent5 Defined
```

2. Perform the same operation on the physical adapter (in this example **ent0**) with the difference that now Unconfigure any Child Devices has to be set to Yes.

3. run **diagmenu**, select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Replace/Remove a PCI Hot Plug Adapter**, and select the physical adapter ent0 to be removed or replaced. Select replace as the operation and press Enter. You will be presented with the following output:

```
COMMAND STATUS

Command: running      stdout: yes         stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.
```

Press Enter as directed and the next message will appear:

```
The visual indicator for the specified PCI slot has
been set to the action state. Replace the PCI card
in the identified slot and press Enter to continue.
Enter x to exit. Exiting now leaves the PCI slot
in the removed state.
```

4. Locate the blinking adapter, replace it, and press Enter. The system will show the message `Replace Operation Complete`.

5. Run **diagmenu** select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Configure a Defined Device**, and select the physical Ethernet adapter ent0 that was replaced. You should get the following output:

```
Configure a Defined Device

Device Name
Move cursor to desired item and press Enter. Use arrow keys to scroll.
[MORE...4]
concurrent  Available               Virtual Target Device - Disk
dac0        Available 04-08-02      3542     (200) Disk Array Controll
dar0        Available               3542     (200) Disk Array Router
db_rootvg   Available               Virtual Target Device - Disk
en0         Defined   03-08         Standard Ethernet Network Interfac
en1         Defined                 Standard Ethernet Network Interfac
en2         Available               Standard Ethernet Network Interfac
en3         Defined                 Standard Ethernet Network Interfac
en4         Defined                 Standard Ethernet Network Interface
ent0        Defined   03-08         10/100/1000 Base-TX PCI-X Adapter
[MORE...72]
```

   Press Enter to configure the adapter, thereby changing its state from Defined to Available.

6. Repeat the Configure operation for the Shared Ethernet Adapter.

This method changes if the physical Ethernet adapter is part of a Network Interface Backup configuration or a IEEE 802.3ad link aggregation.

5.1.9 Replacing a Fibre Channel adapter on the Virtual I/O Server

For Virtual I/O Servers, we recommend that you have at least two Fibre Channel adapters attached for redundant access to FC attached disks. This allows for concurrent maintenance, since the multipathing driver of the attached storage subsystem is supposed to handle any outages of a single Fibre Channel adapter. We will show the procedure to hot-plug a Fibre Channel adapter connected to a DS4200 storage device. Depending on the storage subsystem used and the multipathing driver installed, your mileage may vary.

If there are disks mapped to virtual SCSI adapters, these devices have to be unconfigured first since there has been no automatic configuration method been used to define them.

1. Use the **diagmenu** command to unconfigure devices dependent on the Fibre Channel adapter. Run **diagmenu**, select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Unconfigure a device**, select the disk or the disks in question, and set its state to Defined, as shown in the following:

```
Unconfigure a Device

Device Name

Move cursor to desired item and press Enter. Use arrow keys to
scroll.
[MORE...43]
hdisk6       Available 04-08-02      3542     (200) Disk Array
Device
hdisk9       Defined   09-08-00-4,0  16 Bit LVD SCSI Disk Drive
inet0        Available               Internet Network Extension
iscsi0       Available               iSCSI Protocol Device
lg_dumplv    Defined                 Logical volume
lo0          Available               Loopback Network Interface
loglv00      Defined                 Logical volume
lpar1_rootvg Available               Virtual Target Device - Disk
lpar2_rootvg Available               Virtual Target Device - Disk
lvdd         Available               LVM Device Driver
[MORE...34]
```

2. After that has been done for every mapped disk (Virtual Target Device), set the state of the Fibre Channel Adapter also to Defined:

```
Unconfigure a Device

Device Name                                ?
Move cursor to desired item and press Enter. Use arrow keys to scroll.
[MORE...16]
et1          Defined   05-09         IEEE 802.3 Ethernet Network Inter
et2          Defined                 IEEE 802.3 Ethernet Network Inter
et3          Defined                 IEEE 802.3 Ethernet Network Inter
et4          Defined                 IEEE 802.3 Ethernet Network Inter
fcnet0       Defined   04-08-01      Fibre Channel Network Protocol De
fcnet1       Defined   06-08-01      Fibre Channel Network Protocol De
fcs0         Available 04-08         FC Adapter
fcs1         Available 06-08         FC Adapter?
fscsi0       Available 04-08-02      FC SCSI I/O Controller Protocol D
fscsi1       Available 06-08-02      FC SCSI I/O Controller Protocol D?
[MORE...61]
```

Be sure to set Unconfigure any Child Devices to Yes, as this will unconfigure the fcnet0 and fscsi0 devices as well as the RDAC driver device dac0:

```
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]
* Device Name                                   [fcs0]
  Unconfigure any Child Devices                  yes
  KEEP definition in database                    yes
```

The following is the output of that command, showing the other devices unconfigured:

```
COMMAND STATUS

Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

fcnet0 Defined
dac0 Defined
fscsi0 Defined
fcs0 Defined
```

3. Run **diagmenu**, select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Replace/Remove a PCI Hot Plug Adapter**, and select the adapter to be replaced. Set the operation to replace, then press Enter. You will be presented with the following dialogue:

```
COMMAND STATUS

Command: running       stdout: yes          stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.
```

4. Press Enter as directed and the next message will appear.

```
The visual indicator for the specified PCI slot has
been set to the action state. Replace the PCI card
in the identified slot and press Enter to continue.
Enter x to exit. Exiting now leaves the PCI slot
in the removed state.
```

5. Locate the blinking adapter, replace it, and press Enter. The system will show the message `Replace Operation Complete`.
6. Select **`diagmenu`**, select **Task Selection** → **Hot Plug Task** → **PCI Hot Plug Manager** → **Install/Configure Devices Added After IPL**, and press Enter. This will call the **`cfgdev`** command internally and put all previously unconfigured devices back to Available.
7. If an FC adapter is replaced, the according measurements like zoning on the FC switch and definition of the WWPN of the replaced adapter to the storage subsystem have to be conducted before the replaced adapter can access the disks on the storage subsystem. For DS4000 storage subsystems, we recommend switching the LUN mappings back to their original controllers, as they may have been distributed to balance I/O load.

5.1.10 Changing TCP/IP configuration during production

Starting with Virtual I/O Server Version 1.3, you know no longer need to remove TCP/IP configuration information with the **`rmtcpip`** command and add it back with the **`mktcpip`** command; the **`chtcpip`** command now controls configuration of the TCP/IP parameters on a system.

Example 5-2 shows the syntax of the **`chtcpip`** command on the Virtual I/O Server.

Example 5-2 Syntax of the chtcpip command

```
$ chtcpip -help
Usage: chtcpip [-interface Interface -inetaddr Address -netmask SubnetMask]
       chtcpip [-interface Interface -gateway -add New_Gateway_Address -remove
OLD_Gateway_Address]

       -inetaddr    Change the IP address of the host.
                    Specify the address in dotted decimal
                    notation.

       -netmask     SubnetMask Specifies the mask the gateway
                    should use in determining the appropriate
                    subnetwork for routing.

       -interface   Interface Specifies a particular network
                    interface, for example: en0

       -gateway     Gateway Changes the gateway address for a static
                    route. Specify the current address in dotted decimal
                    notation and specify the new gateway address in
                    dotted decimal notation.
```

```
-add          New Default Gateway address to be added

-remove       Old Default Gateway address to removed
```

5.1.11 HMC topology details view

Tip: The HMC has a feature to aid administrators in looking at virtual SCSI and virtual LAN topologies within the Virtual I/O Server.

The following windows will show you a new feature that comes with the HMC:

- Right-click the Virtual I/O Server partition where you want to see the topologies. Select **Virtual I/O** → **Virtual SCSI adapter**, as shown in Figure 5-18.

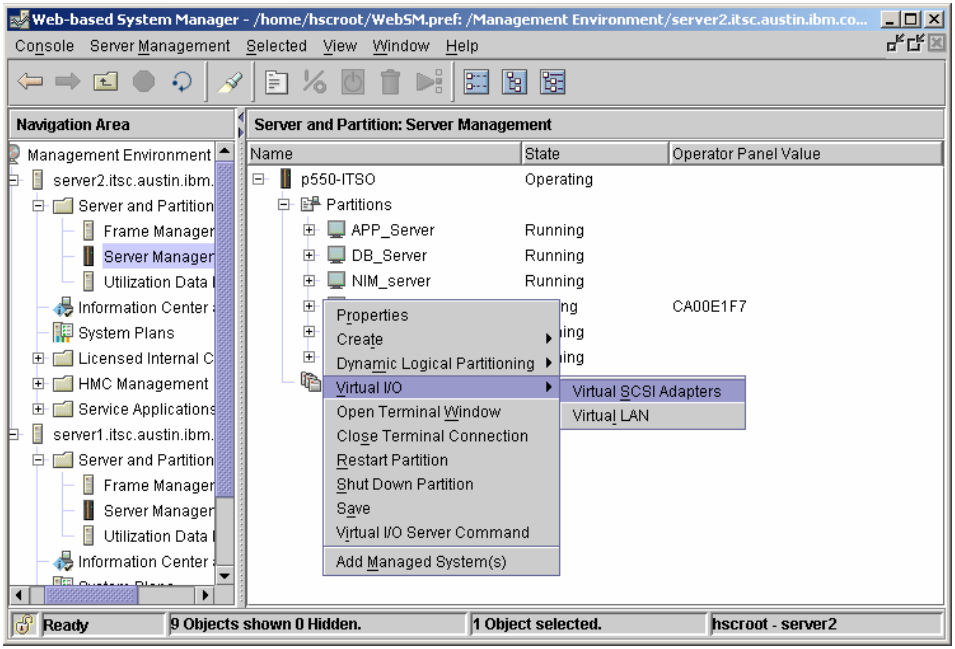


Figure 5-18 Virtual I/O topology view selection

- Figure 5-19 shows you the topology after clicking **Virtual SCSI Adapter**.

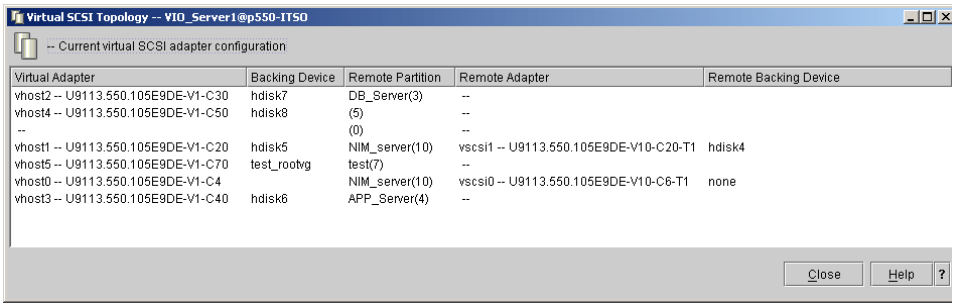


Figure 5-19 VIOS virtual SCSI adapter topology

5.2 Backup and restore of the Virtual I/O Server

This section describes a method to back up and restore the Virtual I/O Server.

5.2.1 Backing up the Virtual I/O Server

The Virtual I/O Server command line interface provides the **`backupios`** command to create an installable image of the root volume group onto either a bootable tape or a multi-volume CD/DVD. The creation of an installable NIM image on a file system is provided as well. Additionally, the system's partition configuration information, including the virtual I/O devices, should be backed up on the HMC. The client data should be backed up from the client system to ensure the consistency of the data.

The **`backupios`** command supports the following backup devices:

- Tape
- File system
- CD
- DVD

For the following three sections, we backed up the Virtual I/O Server using tape, DVD, and file system. No CD backup was performed since it is similar to DVD.

5.2.2 Backing up on tape

In Example 5-3, the result of running the **backupios** command with the -tape flag is shown.

Example 5-3 Tape backup

```
$ backupios -tape /dev/rmt0

Creating information file (/image.data) for rootvg..

Creating tape boot image.....

Creating list of files to back up.
Backing up 23622 files...................
23622 of 23622 files (100%)
0512-038 mksysb: Backup Completed Successfully.

bosboot: Boot image is 26916 512 byte blocks.

bosboot: Boot image is 26916 512 byte blocks.
```

The result of this command is a bootable tape that allows an easy restore of the Virtual I/O Server, as shown in “Restoring from tape” on page 286.

5.2.3 Backing up on DVD

There are two types of DVD media that can be used for backing up: DVD-RAM and DVD-R. DVD-RAM media can support both -cdformat and -udf format, while DVD-R media only supports the -cdformat. The DVD device cannot be virtualized and assigned to a client partition when performing **backupios**. Remove the device from the client and the virtual SCSI mapping from the server before proceeding with the backup.

Example 5-4 shows how to back up the VIOS on DVD-RAM.

Example 5-4 DVD backup

```
$ backupios -cd /dev/cd0 -udf

Creating information file for volume group datapool..

Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
Backup in progress.  This command can take a considerable amount of
time
to complete, please be patient...

Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating image.data file...
Creating temporary file system: /mkcd/mksysb_image...
Creating mksysb image...

Creating list of files to back up.
Backing up 27129 files.....................
27129 of 27129 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Populating the CD or DVD file system...
Copying backup to the CD or DVD file system...
........................................
........................................
........................................
........................................
.......................
Building chrp boot image...

Removing temporary file system: /mkcd/mksysb_image...
```

5.2.4 Backing up on a file system

The result of the **backupios** command is a backup image in the tar format. This file will be stored in the directory specified by the -file flag. Example 5-5 on page 283 shows the creation of a subdirectory named backup_loc, and the **backupios** command with its output.

Example 5-5 File system backup

```
$ mkdir /home/padmin/backup_loc
$ backupios -file /home/padmin/backup_loc

Creating information file for volume group datapool..

Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
Backup in progress.  This command can take a considerable amount of
time
to complete, please be patient...
```

The **ls** command shows that the backup was creating a tar file successfully:

```
-rw-r--r--   1 root     staff     653363200 Jan 11 21:13
nim_resources.tar
```

When only the backup image of the VIOS is required and not a file with NIM resources, the -mksysb flag is used. With this flag, a file name is required with the directory for the backup, for example:

```
$ backupios -file /home/padmin/backup_loc/VIOS.img -mksysb
```

For backups using the -file flag, backing up to an NFS mounted file system is useful for saving the backup image on another machine.

Backup to NIM

You can use NIM to restore the Virtual I/O Server. The following procedure can be used for creating the NIM resources (See Example 5-6 on page 284 and Example 5-7 on page 284):

1. Export a NIM directory for mounting on the Virtual I/O Server.
2. Create the mksysb image on the NIM server by using NFS when running the **backupios** command on the Virtual I/O Server. Use the -mksysb flag to create the mksysb image only.
3. Create the NIM client if it is not already defined.
4. Create a SPOT resource from the mksysb image unless you already have a SPOT at the correct matching level.
5. You are now ready to do a NIM install of the Virtual I/O Server.

Example 5-6 Creating a Virtual I/O Server backup on a NIM Server

```
$ mount nim:/export/nim/images /mnt
$ backupios -file /mnt/vios1_16Oct2006 -mksysb

/mnt/vios1_16Oct2006  doesn't exist.

Creating /mnt/vios1_16Oct2006

Creating information file for volume group datapool..

Creating list of files to back up.
Backing up 12 files
12 of 12 files (100%)
0512-038 savevg: Backup Completed Successfully.
Backup in progress.  This command can take a considerable amount of
time
to complete, please be patient...


Creating information file (/image.data) for rootvg.

Creating list of files to back up.
Backing up 45234 files.............................
29416 of 45234 files (65%)....
45234 of 45234 files (100%)
0512-038 savevg: Backup Completed Successfully.
```

Example 5-7 Using smit to define the NIM mksysb and spot resources

```
Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
* Resource Name                                      [vios1_16Oct2006]
* Resource Type                                       mksysb
* Server of Resource                                 [master]                +
* Location of Resource                              <mages/vios1_16Oct2006]  /
  Comments                                          []

  Source for Replication                            []                      +
                -OR-
  System Backup Image Creation Options:
    CREATE system backup image?                      no                     +
    NIM CLIENT to backup                            []                      +
```

```
    PREVIEW only?                                    no                  +
    IGNORE space requirements?                       no                  +
[MORE...10]

Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Resource Name                                     [vios1_16Oct2006_spot]
* Resource Type                                      spot
* Server of Resource                                [master]               +
* Source of Install Images                          [vios1_16Oct2006]      +
* Location of Resource                             <spots/vios1_16Oct2006]  /
  Expand file systems if space needed?               yes                  +
  Comments                                          []

  installp Flags
  COMMIT software updates?                           no                   +
  SAVE replaced files?                               yes                  +
  AUTOMATICALLY install requisite software?          yes                  +
  OVERWRITE same or newer versions?                  no                   +
  VERIFY install and check file sizes?               no                   +
```

Note: When you prepare the Virtual I/O Server for installation on the NIM server, set Remain NIM client after install to No unless you want the NIM defined interface to be configured on the installation adapter.

5.2.5 Restoring the Virtual I/O Server

The following sections describe the restore process for the Virtual I/O Server depending on your chosen backup format.

Restoring from tape

To restore the Virtual I/O Server from tape, boot the Virtual I/O Server partition to the SMS menu and select the tape drive as the install device. Then continue the same was as a normal AIX 5L installation. Example 5-8 shows the selection of a tape drive for restoring a previously made tape backup.

Example 5-8 selecting tape drive for restore of the Virtual I/O Server

```
PowerPC Firmware
 Version SF240_219
 SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
 Select Device
 Device  Current  Device
 Number  Position  Name
 1.        1      SCSI Tape
                  ( loc=U787B.001.DNW108F-P1-T14-L0-L0 )









-------------------------------------------------------------------------------
 Navigation keys:
 M = return to Main Menu
 ESC key = return to previous screen        X = eXit System Management Services

-------------------------------------------------------------------------------
 Type menu item number and press Enter or select Navigation key:
```

Restoring from DVD

To restore the Virtual I/O Server from DVD, boot the Virtual I/O Server partition to the SMS menu and select the DVD device as the install device. Continue the install similar to a normal AIX 5L installation.

Restoring from file system using the installios command

The restore of a Virtual I/O Server file system backup is done using the **installios** command of the HMC or AIX 5L. For a restore, the tar file must be located either on the HMC, an NFS-accessible directory, or a DVD. To make the tar file created with the **backupios** command accessible for restore, we performed the following steps:

1. Create a directory named backup using the **mkdir /home/padmin/backup** command.
2. Check that the NFS server is exporting a file system with the **showmount nfs_server** command.
3. Mount the NFS exported file system onto the backup directory.
4. Copy the tar file created in 5.2.4, “Backing up on a file system” on page 282 to the NFS mounted directory using the following command:

   ```
   $ cp /home/padmin/backup_loc/nim_resources.tar /home/padmin/backup
   ```

At this stage, the backup is ready to be restored to the Virtual I/O Server partition using the **installios** command on the HMC or an AIX 5L partition that is a NIM server. The restore procedure will shut down the Virtual I/O Server partition if it is still running. The following is an example of the command help:

```
hscroot@server1:~> installios -?
installios: usage: installios [-s managed_sys -S netmask -p partition
                        -r profile -i client_addr -d source_dir -m mac_addr
                        -g gateway [-P speed] [-D duplex] [-n] [-l language]]
                        | -u
```

Using the **installios** command, the -s managed_sys option requires the HMC defined system name, the -p partition option requires the name of the VIOS partition, and the -r profile option requires the partition profile you want to use to boot the VIOS partition during the recovery.

If you do not specify the -m flag and include the MAC address of the VIOS being restored, the restore will take longer as the **installios** command shuts down the VIOS and boots it in SMS to determine the MAC address. The following is an example of the command usage:

```
hscroot@server1:~> installios -s p550-ITSO -S 255.255.255.0 -p
testvios2 -r default -i 9.3.5.125 -d 9.3.5.126:/export_fs -m
00:02:55:d3:dc:34 -g 9.3.5.41
```

Following this command, NIMOL on the HMC takes over the NIM process and mounts the exported file system to process the **backupios** tar file created on the VIOS previously. The HMC NIM then proceeds with a normal install of the VIOS and one final reboot of the partition completes the install.

For an AIX 5L NIM server recovery, the same command is used:

```
# installios -?
Usage: installios [-h hmc -s managed_sys -p partition -r profile ]
        -S netmask -i client_addr -g gateway -d source_dir
        [-P speed] [-D duplex] [-n] [-N] [-l language] [-L location] | -u[f|U]
```

This command can be run either on the command line or using the **smitty installios** command. For this command to work, SSH needs to be configured between the NIM server and the HMC. Example 5-9 shows the output of the **smitty installios** command that accepts the same information as the **installios** command.

Note: If you prefer to use NIM in the same way as for other servers, you can generate the Virtual I/O Server backup using the **-mksysb** flag. See 5.2.4, "Backing up on a file system" on page 282, Backup to NIM for steps to create the required NIM resources.

Example 5-9 AIX 5L smitty installios menu

```
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
* Select or specify software source                  [cd0]                  +
  to initialize environment

  HMC Name                                           []
  Managed System Name                                []
  Partition Name                                     []
  Partition Profile Name                             []

  Primary Network Install Interface
*   IP Address Used by Machine                       []
*   Subnetmask Used by Machine                       []
*     Default Gateway Used by Machine                []
    Network Speed Setting                            [100]                  +
Network Duplex Setting                              [full]                 +

  Language to Install                                [en_US]                +
  Configure Client Network After Installation        [yes]                  +
  ACCEPT I/O Server License Agreement?               [no]                   +
[BOTTOM]
```

The AIX 5L **installios** command uses SSH commands to the HMC to shut down the LPAR and reboot it to get both the MAC address and to start the NIM

installation. Either a DVD backup or a file backup can be used to perform this restore.

You are now ready to do a NIM install of the Virtual I/O Server.

5.3 Rebuilding the Virtual I/O Server

This section describes what to do if there are no valid backup devices or backup images. In this case, you must install a new Virtual I/O Server.

In the following discussion, we assume that the partition definitions of the Virtual I/O Server and of all clients on the HMC are still available. We describe how we rebuilt our configuration of network and SCSI configurations.

It is useful to generate a System Plan on the HMC as documentation of partition profiles, settings, slot numbers, and so on. Example 5-10 shows the command to create a System Plan for a managed system. Note that the file name must have the extension .sysplan.

Example 5-10 Creating an HMC System Plan

```
hscroot@server2:~> mksysplan -f p550-ITSO.sysplan -m p550-ITSO
```

To view the System Plan, Select **System Plans** and then **Manage System Plans**. Point to the System Plan in the Manage System Plans window and click **View**. A browser window is opened where you are prompted for the user name and password of the HMC. Figure 5-20 shows a System Plan generated from a managed system.

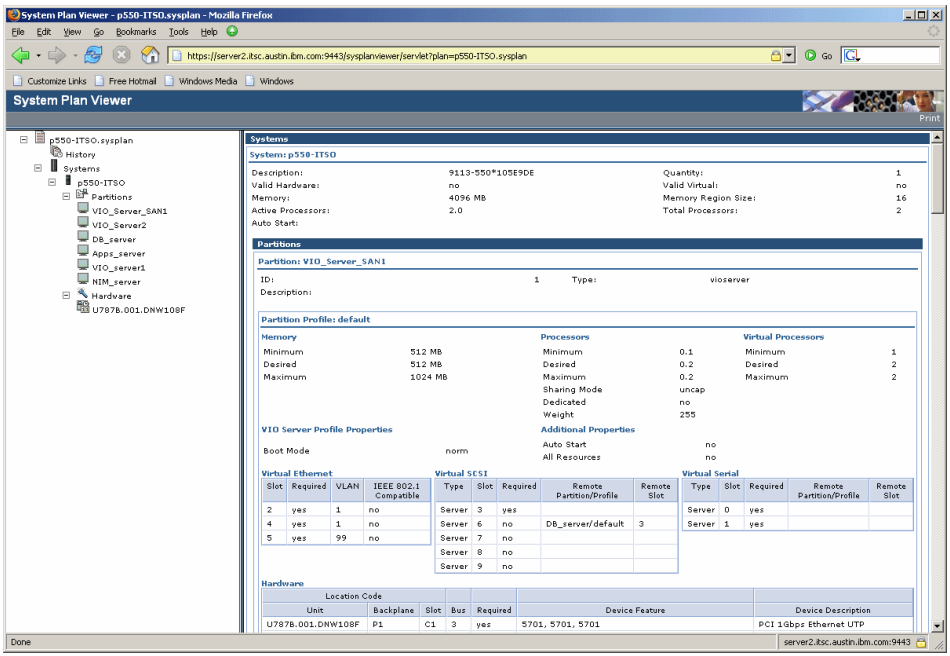


Figure 5-20 Example of a System Plan generated from a managed system

In addition to the regular backups using the **backupios** command, we recommend documenting the configuration of the following topics using the commands provided:

- Network settings

  Commands: **netstat -state**, **netstat -routinfo**, **netstat -routtable**, **lsdev -dev *Device* -attr, cfgnamsrv -ls, hostmap -ls, optimizenet -list** and **entstat -all *Device***

- All physical and logical volumes SCSI devices

  Commands: **lspv**, **lsvg**, and **lsvg -lv *VolumeGroup***

- All physical and logical adapters

  Command: **lsdev -type adapter**

- The mapping between physical and logical devices and virtual devices

  Commands: **lsmap -all** and **lsmap -all -net**

- Code levels, users and security

  Commands: **ioslevel, viosecure -firewall view, viosecure -view -nonint**

With this information, you are able to reconfigure your Virtual I/O Server manually. In the following sections, we describe the commands we needed to get the necessary information and the commands that rebuilt the configuration. The important information from the command outputs is highlighted. Depending on your environment, the commands may differ from those shown as examples.

To start rebuilding the Virtual I/O Server, you must know which disks are used for the Virtual I/O Server itself and for any assigned volume groups for virtual I/O.

The **lspv** command shows us that the Virtual I/O Server was installed on hdisk0. The first step is to install the new Virtual I/O Server from the installation media onto disk hdisk0. This command can be run from **diag** command environment or another AIX 5L environment:

```
hdisk0          00cddedc01300ed3                    rootvg          active
hdisk1          00cddedc143815fb                    None
hdisk2          00cddedc4d209163                    client_disks    active
hdisk3          00cddedc4d2091f8                    datavg          active
```

See 3.3, “Virtual I/O Server software installation” on page 142 for the installation procedure. The further rebuild of the Virtual I/O Server is done in two steps:

1. Rebuild the SCSI configuration.
2. Rebuild the network configuration.

5.3.1 Rebuild the SCSI configuration

The **lspv** command also shows us that there are two additional volume groups located on the Virtual I/O Server (client_disks and datavg):

```
hdisk0          00cddedc01300ed3                    rootvg          active
hdisk1          00cddedc143815fb                    None
hdisk2          00cddedc4d209163                    client_disks    active
hdisk3          00cddedc4d2091f8                    datavg          active
```

The following commands import this information into the new Virtual I/O Server system’s ODM:

```
importvg -vg client_disks hdisk2
importvg -vg datavg hdisk3
```

In Example 5-11, we look to the mapping between the logical and physical volumes and the virtual SCSI server adapters.

Example 5-11 lsmap -all

```
$ lsmap -all
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9111.520.10DDEDC-V4-C30                     0x00000000

VTD                   vtscsi2
LUN                   0x8100000000000000
Backing device        datalv
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U9111.520.10DDEDC-V4-C10                     0x00000006

VTD                   vtscsi0
LUN                   0x8100000000000000
Backing device        rootvg_ztest0
Physloc

VTD                   vtscsi1
LUN                   0x8200000000000000
Backing device        hdisk1
Physloc               U787A.001.DNZ00XY-P1-T10-L4-L0
```

Virtual SCSI server adapter vhost0 (defined on slot 30 in HMC) is mapped to Logical Volume datalv by Virtual Target Device vtscsi2.

Virtual SCSI server adapter vhost2 has two Virtual Target Devices, vtscsi0 and vtscsi1. They are mapping Logical Volume rootvg_ztest0 and Physical Volume hdisk1 to vhost2 (is defined on slot 10 in HMC).

The following commands are used to create our needed Virtual Target Devices:

```
mkvdev -vdev datalv -vadapter vhost0
mkvdev -vdev rootvg_ztest0 -vadapter vhost2
mkvdev -vdev hdisk1 -vadapter vhost2
```

Note: The names of the Virtual Target Devices are generated automatically, except when you define a name using the -dev flag of the **mkvdev** command.

5.3.2 Rebuild network configuration

After successfully rebuilding the SCSI configuration, we now are going to rebuild the network configuration.

The **netstat -state** command shows us that en2 is the only active network adapter:

```
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en2   1500  link#2      0.d.60.a.58.a4      2477     0      777     0     0
en2   1500  9.3.5       9.3.5.147           2477     0      777     0     0
lo0   16896 link#1                           153     0      158     0     0
lo0   16896 127         127.0.0.1            153     0      158     0     0
lo0   16896 ::1                             153     0      158     0     0
```

With the **lsmap -all -net** command, we determine that ent2 is defined as a shared Ethernet adapter mapping physical adapter ent0 to virtual adapter ent1:

```
SVEA   Physloc
------ --------------------------------------------
ent1   U9111.520.10DDEDC-V4-C2-T1

SEA                 ent2
Backing device      ent0
Physloc             U787A.001.DNZ00XY-P1-C2-T1
```

The information for the default gateway address is provided by the **netstat -routinfo** command:

```
Routing tables
Destination      Gateway           Flags     Wt  Policy  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.5.41          UG        1    -     en2     0         0
9.3.5.0          9.3.5.147         UHSb      1    -     en2     0         0 =>
9.3.5/24         9.3.5.147         U         1    -     en2     0         0
9.3.5.147        127.0.0.1         UGHS      1    -     lo0     0         0
9.3.5.255        9.3.5.147         UHSb      1    -     en2     0         0
127/8            127.0.0.1         U         1    -     0       0         0
```

To list the subnet mask, we use the **lsdev -dev en2 -attr** command:

```
netmask       255.255.255.0 Subnet Mask
True
```

The last information we need is the default virtual adapter and the default PVID for the shared Ethernet adapter. This is shown by the **lsdev -dev ent2 -attr** command:

```
attribute     value description                                              user_settable

pvid          1     PVID to use for the SEA device                                    True
pvid_adapter  ent1  Default virtual adapter to use for non-VLAN-tagged packets       True
real_adapter  ent0  Physical adapter associated with the SEA                         True
virt_adapters ent1  List of virtual adapters associated with the SEA (comma separated) True
```

The following commands re-created our network configuration:

```
mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1
mktcpip -hostname p51iosrv2 -inetaddr 9.3.5.147 -interface en2 -start
-netmask 255.255.255.0 -gateway 9.3.5.41
```

These steps complete the basic rebuilding of the Virtual I/O Server.

5.4 System maintenance for the Virtual I/O Server

In this section, system maintenance for the VIOS is covered. Our topics include hot swapping devices and recovering from a disk failure on both the VIOS and client partition. We start with methods for updating VIOS software on an active VIOS configuration without disrupting VIOS client availability to VIOS services.

5.4.1 Concurrent software updates for the VIOS

In this section, the steps of updating the Virtual I/O Servers in a multiple Virtual I/O Server environment are covered. A simple software update is covered in “Updating the VIOS” on page 313.

We set up two Virtual I/O Server partitions and two client partitions to show the software update of the Virtual I/O Server in a mirrored and in a MPIO environment. In this configuration, it is possible for clients to continue 24x7 operations without requiring a client outage during a VIOS software update.

The client partition DB_Server is configured to have one virtual disk from VIO_Server1 and a different virtual disk from VIO_Server2. On this client partition, we set up an LVM mirror. Detailed steps for the configuration of this scenario can be found in 4.2, “Scenario 1: Logical Volume Mirroring” on page 207.

The client partition, APP_server, is configured as an MPIO client. Both Virtual I/O Servers provide paths to the same SAN disk for this client partition. Detailed

steps for this configuration can be found in 4.4, “Scenario 3: MPIO in the client with SAN” on page 218.

The configuration on the system is shown in Figure 5-21.

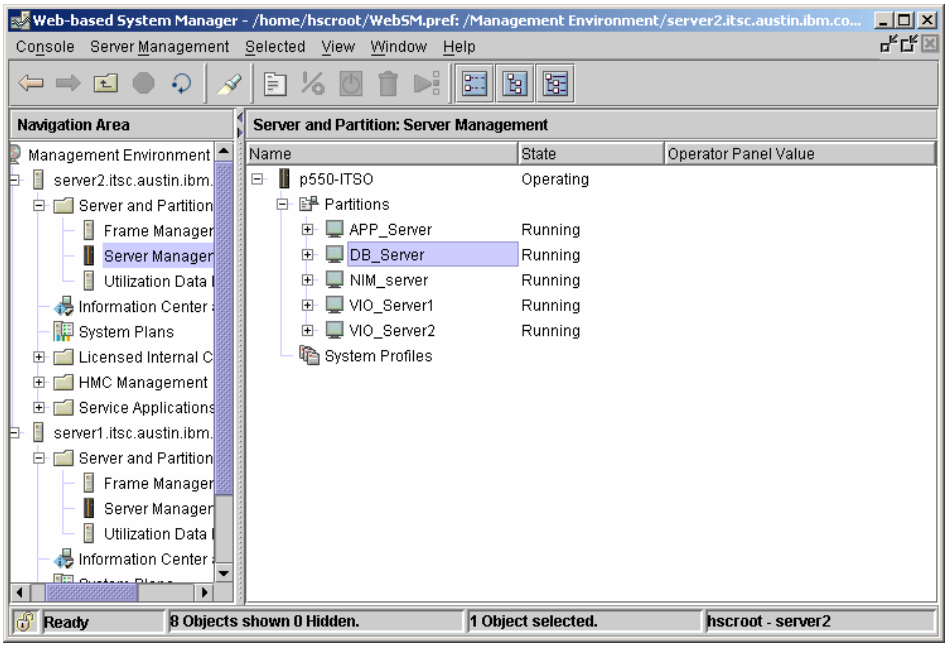


Figure 5-21 Concurrent software update configuration

In this test setup, both Virtual I/O Servers have Version 1.2.1.0 installed and will be updated to Version 1.3 using Fix Pack 8. Virtual I/O Server updates can be found at the following URL:

`http://techsupport.services.ibm.com/server/vios/download/home.html`

The client partitions are running AIX 5L V5.3 TL 05. The steps shown here are typical for all updates of the Virtual I/O Servers in an multiple Virtual I/O Server environment regardless of the OS level.

Before we start to update the Virtual I/O Server, we checked our client partitions to be certain that no previous activity, such as rebooting the Virtual I/O Server, shows stale partition on the client partition or paths disabled. In these cases, the paths may not be operational, or the disk mirrors similar.

In a mirrored environment, issue the following steps for preparation:

1. On the client partition, in our case the DB_Server partition, issue the **lsvg** command to check whether there are any stale partitions:

```
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     4     8     2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/stale    N/A
hd4                 jfs2       1     2     2    open/stale    /
hd2                 jfs2       5     10    2    open/stale    /usr
hd9var              jfs2       1     2     2    open/stale    /var
hd3                 jfs2       1     2     2    open/stale    /tmp
hd1                 jfs2       1     2     2    open/stale    /home
hd10opt             jfs2       1     2     2    open/stale    /opt
```

2. If you see a stale partition, resynchronize the volume group using the **varyonvg** command. Make sure that the Virtual I/O Server is up and running and all mappings are in the correct state:

```
# varyonvg rootvg
```

3. After running this command, check again using the **lsvg** command to show that the volume group is synchronized:

```
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     4     8     2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/syncd    N/A
hd4                 jfs2       1     2     2    open/syncd    /
hd2                 jfs2       5     10    2    open/syncd    /usr
hd9var              jfs2       1     2     2    open/syncd    /var
hd3                 jfs2       1     2     2    open/syncd    /tmp
hd1                 jfs2       1     2     2    open/syncd    /home
hd10opt             jfs2       1     2     2    open/syncd    /opt
```

In a MPIO environment, issue the following steps for preparation:

1. On the client partition, in our case the APP_Server partition, check the state using the **lspath** command. Both paths are in an enabled state. If one is missing or in a disabled state, check for possible reasons. If the Virtual I/O Server was rebooted earlier and the health_check attribute is not set, you may need to enable it.

2. Issue the following command to check the paths:

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

3. After ensuring both paths function without problems, disable the path to the Virtual I/O Server that will receive the first software update. In our example, we start by updating VIO_Server1. The **lspath** command shows us that one path is connected using the vscsi0 adapter and one using the vscsi1 adapter.

   Issue the **lscfg** command to determine the slot number of the vscsi0 adapter:

```
# lscfg -vl vscsi0
  vscsi0 U9113.550.105E9DE-V3-C30-T1  Virtual SCSI Client Adapter

        Device Specific.(YL)........U9113.550.105E9DE-V3-C30-T1
```

4. To discover to which Virtual I/O Server the vscsi0 adapter is connected, access the active profile of the client partition on the HMC.

   Go to Server Management and choose the active profile of the client partition, as shown in Figure 5-22.

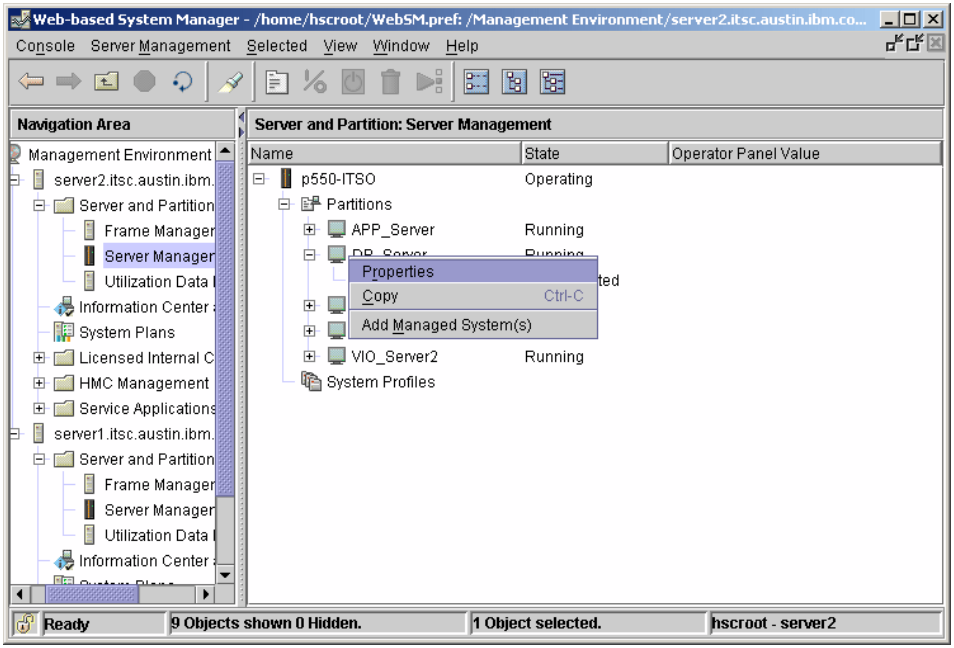


Figure 5-22 HMC profile properties

Right-click the profile and choose **Properties**. In the Properties window, choose the **Virtual I/O Adapters** tab.

Figure 5-23 shows that the virtual SCSI adapter in slot 30 is connected to the VIO_Server1 partition.

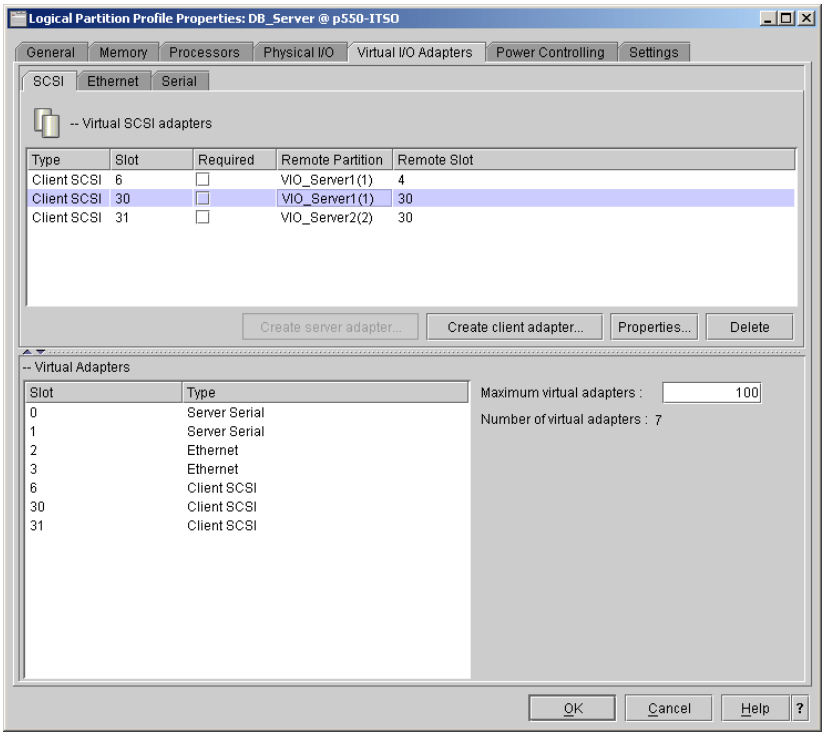


Figure 5-23 LPAR properties window

5. Disable the vscsi0 path for the hdisk0 using the **chpath** command.

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
# chpath -l hdisk0 -p vscsi0 -s disable
paths Changed
# lspath
Disabled hdisk0 vscsi0
Enabled  hdisk0 vscsi1
```

Now we can start to update the VIO_Server1 partition.

1. The Fix Pack 8 has been downloaded from the Virtual I/O Server support Web site at:

   http://techsupport.services.ibm.com/server/vios/download/home.html

2. On an AIX system with sufficient space, the file fixpack80.tar.gz has been extracted into a new directory via the command

   ```
   # gunzip -dc ../fixpack80.tar.gz | tar -xvf-
   ```

3. On the VIO_Server1 partition, a new directory /home/padmin/updates has been created and the contents of the extracted Fix Pack were transmitted using the **ftp** command from the AIX system to this directory.

4. On the VIO_Server1 partition, to upgrade, we issued the **updateios** command; the results are shown in Example 5-12.

Example 5-12 updateios command output

```
$ ioslevel
1.2.1.0
$ updateios -dev /home/padmin/updates -install -accept


Validating RPM package selections ...


********************************************************************************
installp PREVIEW:  installation will not actually occur.
********************************************************************************

+-------------------------------------------------------------------------------+
                    Pre-installation Verification...
+-------------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

WARNINGS
--------
  Problems described in this section are not likely to be the source of any
  immediate or serious failures, but further actions may be necessary or
  desired.

  Already Installed
  -----------------
  The following filesets which you selected are either already installed
  or effectively installed through superseding filesets.

    rpm.rte 3.0.5.39                          # RPM Package Manager
    ios.ldw.gui 1.1.0.0                       # LPAR Deployment Wizard Graph...
    ios.cli.man.fr_FR 5.3.0.0                 # Virtual I/O Technologies Man...
    devices.vdevice.IBM.vmc.rte 5.3.0.0       # Virtual Management Channel
    devices.pci.O21O5e51.rte 5.3.0.0          # Native Display Adapter Softw...
    csm.dsh 1.5.0.0                           # Cluster Systems Management Dsh
```

```
    csm.diagnostics 1.5.0.0                  # Cluster Systems Management P...
    csm.core 1.5.0.0                         # Cluster Systems Management Core
    csm.client 1.5.0.0                       # Cluster Systems Management C...
    devices.pci.a8135201.rte 5.3.0.0         # Native 2-Port Asynchronous E...
    devices.pci.a8135201.diag 5.3.0.0        # 2-port Asynchronous EIA-232 ...
    devices.pci.77102224.com 5.3.0.0         # PCI-X FC Adapter (77102224) ...
    devices.pci.77102224.rte 5.3.0.0         # PCI-X FC Adapter (77102224) ...
    devices.pci.77102224.diag 5.3.0.0        # PCI-X FC Adapter (77102224) ...
    devices.pci.1410ec02.rte 5.3.0.0         # 10 Gigabit Ethernet-LR PCI-X...
    devices.pci.1410ec02.diag 5.3.0.0        # 10 Gigabit Ethernet-LR PCI-X...
    devices.pci.1410eb02.rte 5.3.0.0         # 10 Gigabit-SR Ethernet DDR P...
    devices.pci.1410eb02.diag 5.3.0.0        # 10Gb Ethernet -SR PCI-X 2.0 ...
    devices.pci.1410e202.rte 5.3.0.0         # IBM 1 Gigabit-SX iSCSI TOE P...
    devices.pci.1410e202.diag 5.3.0.0        # IBM 1 Gigabit-SX iSCSI TOE P...
    devices.pci.1410bf02.rte 5.3.0.0         # PCI-XDDR Quad Channel U320 S...
    devices.pci.1410bf02.diag 5.3.0.0        # PCI-X DDR Quad Channel U320 ...
    devices.pci.14108d02.rte 5.3.0.0         # PCI-XDDR Dual Channel SAS RA...
    devices.pci.14108d02.diag 5.3.0.0        # PCI-X DDR Dual Channel SAS R...
    devices.pci.14102203.rte 5.3.0.0         # IBM 1 Gigabit-TX iSCSI TOE P...
    devices.pci.14102203.diag 5.3.0.0        # IBM 1 Gigabit-TX iSCSI TOE P...
    devices.pci.14101403.rte 5.3.0.0         # Gigabit Ethernet-SX Adapter ...
    devices.pci.14101403.diag 5.3.0.0        # Gigabit Ethernet-SX Adapter ...
    devices.pci.14101103.rte 5.3.0.0         # 4-Port 10/100/1000 Base-TX P...
    devices.pci.14101103.diag 5.3.0.0        # 4-Port 10/100/1000 Base-TX P...
    devices.chrp.AT97SC3201_r.rte 5.3.0.0    # Trusted Platform Module Devi...
    invscout.com 2.2.0.1                     # Inventory Scout Microcode Ca...
    devices.usbif.03000008.rte 5.3.0.0       # USB 3D Mouse Client Driver
    devices.pci.df1000fd.rte 5.3.0.0         # 4Gb PCI-X FC Adapter Device ...
    devices.pci.df1000fd.diag 5.3.0.0        # FC PCI-X Adapter Device Diag...
    devices.pci.77102e01.diag 5.3.0.0        # 1000 Base-TX PCI-X iSCSI TOE...
    devices.pci.77102e01.rte 5.3.0.0         # PCI-X 1000 Base-TX iSCSI TOE...
    devices.pci.1410d402.rte 5.3.0.0         # PCI-X Dual Channel U320 SCSI...
    devices.pci.1410d402.diag 5.3.0.0        # PCI-X Dual Channel U320 SCSI...
    devices.pci.1410d302.rte 5.3.0.0         # PCI-X Dual Channel U320 SCSI...
    devices.pci.1410d302.diag 5.3.0.0        # PCI-X Dual Channel Ultra320 ...
    devices.pci.1410c002.rte 5.3.0.0         # PCI-XDDR Dual Channel U320 S...
    devices.pci.1410c002.diag 5.3.0.0        # PCI-X DDR Dual Channel U320 ...
    devices.pci.1410be02.rte 5.3.0.0         # PCI-XDDR Dual Channel U320 S...
    devices.pci.1410be02.diag 5.3.0.0        # PCI-X DDR Dual Channel U320 ...
    devices.chrp.IBM.HPS.rte 1.2.0.0         # IBM eServer pSeries High Per...
    devices.chrp.IBM.HPS.hpsfu 1.2.0.0       # IBM pSeries HPS Functional U...
    devices.msg.en_US.common.IBM.sni.rte 1.2.0.0 # Switch Network Interface
Runtime Messages - U.S. English
    devices.msg.en_US.common.IBM.sni.ml 1.2.0.0 # Multi Link Interface Runtime
- U.S. English
    devices.msg.en_US.common.IBM.sni.ntbl 1.2.0.0 # Network Table Runtime
Messages - U.S. English
    devices.msg.en_US.chrp.IBM.HPS.rte 1.2.0.0 # pSeries HPS Rte Msgs - U.S.
English
```

```
    devices.msg.en_US.chrp.IBM.HPS.hpsfu 1.2.0.0 # pSeries HPS Func Util Msgs -
U.S. English
    devices.common.IBM.sni.rte 1.2.0.0        # Switch Network Interface Run...
    devices.common.IBM.sni.ml 1.2.0.0         # Multi Link Interface Runtime
    devices.common.IBM.sni.ntbl 1.2.0.0       # Network Table Runtime
    devices.pci.1410cf02.rte 5.3.0.0          # 1000 Base-SX PCI-X iSCSI TOE...
    devices.pci.1410cf02.diag 5.3.0.0         # 1000 Base-SX PCI-X iSCSI TOE...
    invscout.ldb 2.2.0.2                      # Inventory Scout Logic Database
    Java14.license 1.4.2.0                    # Java SDK 32-bit License
    devices.msg.en_US.common.IBM.sni.ntbl 1.2.0.1 # Network Table Runtime
Messages - U.S. English
    bos.rte.net 5.3.0.40                      # Network
    devices.common.IBM.usb.diag 5.3.0.40      # Common USB Adapter Diagnostics
    bos.rte.iconv 5.3.0.40                    # Language Converters
    devices.scsi.tape.diag 5.3.0.30           # SCSI Tape Device Diagnostics
    devices.scsi.disk.diag.rte 5.3.0.30       # SCSI CD_ROM, Disk Device Dia...
    devices.pci.14106602.ucode 5.3.0.30       # PCI-X Dual Channel SCSI Adap...
    bos.net.tcp.smit 5.3.0.30                 # TCP/IP SMIT Support
    devices.pci.14105300.rte 5.3.0.30         # IBM PCI ATM 25MBPS Adapter S...
    devices.pci.14107c00.com 5.3.0.30         # Common ATM Adapter Software
    devices.pci.14106402.rte 5.3.0.30         # PCI-X Quad Channel U320 SCSI...
    devices.pci.df1000f9.rte 5.3.0.30         # 64-bit PCI FC Adapter Device...
    devices.pci.df1080f9.rte 5.3.0.30         # PCI-X FC Adapter Device Soft...
    devices.pci.df1000f7.rte 5.3.0.30         # PCI FC Adapter Device Software
    devices.usbif.08025002.diag 5.3.0.30      # USB CD-ROM Diagnostics
    devices.pci.331121b9.diag 5.3.0.30        # PCI 2-Port Multiprotocol Ada...
    ifor_ls.base.cli 5.3.0.30                 # License Use Management Runti...
    devices.scsi.scarray.rte 5.3.0.30         # 7135 RAIDiant Array DA Devic...
    devices.scsi.ses.rte 5.3.0.30             # SCSI Enclosure Device Software
    devices.iscsi.disk.rte 5.3.0.30           # iSCSI Disk Software
    devices.pci.14108c00.rte 5.3.0.30         # ARTIC960Hx 4-Port Selectable...
    devices.scsi.tm.rte 5.3.0.30              # SCSI Target Mode Software
    devices.pci.14107802.ucode 5.3.0.30       # PCI-X Dual Channel Ultra320 ...
    devices.pci.2b101a05.diag 5.3.0.20        # GXT120P Graphics Adapter Dia...
    devices.pci.14103302.diag 5.3.0.20        # GXT135P Graphics Adapter Dia...
    devices.pci.14105400.diag 5.3.0.20        # GXT500P/GXT550P Graphics Ada...
    devices.pci.2b102005.diag 5.3.0.20        # GXT130P Graphics Adapter Dia...
    devices.pci.isa.rte 5.3.0.10              # ISA Bus Bridge Software (CHRP)
    devices.pci.331121b9.rte 5.3.0.10         # IBM PCI 2-Port Multiprotocol...
    devices.pci.14107c00.diag 5.3.0.10        # PCI ATM Adapter (14107c00) D...
    devices.pci.1410e601.diag 5.3.0.10        # IBM Cryptographic Accelerato...
    devices.pci.14101800.diag 5.3.0.10        # PCI Tokenring Adapter Diagno...
    devices.pci.14103e00.diag 5.3.0.10        # IBM PCI Tokenring Adapter (1...
    devices.pci.23100020.diag 5.3.0.10        # IBM PCI 10/100 Mb Ethernet A...
    devices.pci.22100020.diag 5.3.0.10        # PCI Ethernet Adapter Diagnos...
    devices.isa_sio.chrp.ecp.rte 5.3.0.10     # CHRP IEEE1284 Parallel Port ...
    perl.libext 2.1.0.10                      # Perl Library Extensions
    devices.pci.77101223.rte 5.3.0.10         # PCI FC Adapter (77101223) Ru...
    devices.pci.5a107512.rte 5.3.0.10         # IDE Adapter Driver for Promi...
```

```
    bos.txt.spell 5.3.0.10                  # Writer's Tools Commands
    Java14.license 1.4.2.0                  # Java SDK 32-bit License
    devices.iscsi.tape.rte 5.3.0.30         # iSCSI Tape Software
    devices.scsi.ses.diag 5.3.0.30          # SCSI Enclosure Services Devi...
    devices.pci.14106402.ucode 5.3.0.30     # PCI-X Quad Channel U320 SCSI...
    devices.pci.00100100.com 5.3.0.10       # Common Symbios PCI SCSI I/O ...
    devices.pci.14100401.diag 5.3.0.10      # Gigabit Ethernet-SX PCI Adap...

  NOTE:  Base level filesets may be reinstalled using the "Force"
  option (-F flag), or they may be removed, using the deinstall or
  "Remove Software Products" facility (-u flag), and then reinstalled.


  Multiple install types for same fileset
  ---------------------------------------
  The following filesets have both a base-level and an update fileset on the
  installation media at the same level.  Since they have the same name and
  level, the last fileset on the installation media will be used.

    Java14.license 1.4.2.0

  << End of Warning Section >>

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Mandatory Fileset Updates
  -------------------------
  (being installed automatically due to their importance)
  bos.rte.install 5.3.0.50                  # LPP Install Commands

  << End of Success Section >>

FILESET STATISTICS
------------------
  430  Selected to be installed, of which:
        1  Passed pre-installation verification
      105  Already installed (directly or via superseding filesets)
      324  Deferred (see *NOTE below)
  ----
    1  Total to be installed

*NOTE  The deferred filesets mentioned above will be processed after the
       installp update and its requisites are successfully installed.

RESOURCES
---------
```

```
  Estimated system resource requirements for filesets being installed:
               (All sizes are in 512-byte blocks)
      Filesystem                    Needed Space             Free Space
      /usr                               11664                 1098296
      -----                           --------                  ------
      TOTAL:                             11664                 1098296

  NOTE:  "Needed Space" values are calculated from data available prior
  to installation. These are the estimated resources required for the
  entire operation. Further resource checks will be made during
  installation to verify that these initial estimates are sufficient.

********************************************************************************
End of installp PREVIEW.  No apply operation has actually occurred.
********************************************************************************

Continue the installation [y|n]?
```

Type y and press Enter to start the update. The output of the **updateios** command is very verbose and left out of this example.

5. After a successful update, you must accept the license again:

   ```
   $ license -accept
   ```

   To check the update, run the **ioslevel** command:

   ```
   $ ioslevel
   1.3.0.0
   ```

6. Reboot the Virtual I/O Server.

   After the Virtual I/O Server is up and running again, we need to log in to the client partitions using the first updated VIOS to resynchronize the volume group on the mirrored client partition and to change the path status on the MPIO client partition. This will update the mirror to include any disk I/O that took place during the software upgrade.

   In a mirrored environment, issue the **lsvg** command on the client partitions to check the status of your volume group:

   ```
   # lsvg -l rootvg
   rootvg:
   LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
   POINT
   hd5                 boot       1     2     2    closed/syncd  N/A
   hd6                 paging     4     8     2    open/syncd    N/A
   hd8                 jfs2log    1     2     2    open/stale    N/A
   hd4                 jfs2       1     2     2    open/syncd    /
   hd2                 jfs2       5     10    2    open/syncd    /usr
   hd9var              jfs2       1     2     2    open/stale    /var
   ```

```
hd3                 jfs2       1     2     2    open/syncd    /tmp
hd1                 jfs2       1     2     2    open/syncd    /home
hd10opt             jfs2       1     2     2    open/syncd    /opt
```

Run the **varyonvg** command to synchronize the volume group:

```
# varyonvg rootvg

# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     4     8     2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/syncd    N/A
hd4                 jfs2       1     2     2    open/syncd    /
hd2                 jfs2       5     10    2    open/syncd    /usr
hd9var              jfs2       1     2     2    open/syncd    /var
hd3                 jfs2       1     2     2    open/syncd    /tmp
hd1                 jfs2       1     2     2    open/syncd    /home
hd10opt             jfs2       1     2     2    open/syncd    /opt
```

7. In a MPIO environment, log on to your client partitions and check the path status with the **lspath** command:

```
# lspath
Disabled hdisk0 vscsi0
Enabled  hdisk0 vscsi1
```

8. Enable the vscsi0 path and disable the vscsi1 path:

```
# chpath -l hdisk0 -p vscsi0 -s enable
paths Changed
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
# chpath -l hdisk0 -p vscsi1 -s disable
paths Changed
# lspath
Enabled  hdisk0 vscsi0
Disabled hdisk0 vscsi1
```

9. Update the second Virtual I/O Server by repeating the steps above.
10. After the update is complete and the second Virtual I/O Server partition is rebooted, log on to the client partition to synchronize the rootvg using the **varyonvg** command and enable the path using the **chpath** command, as shown in the examples above.

This completes the update of a redundant VIOS configuration while maintaining client application availability.

5.4.2 Hot pluggable devices

Similar to AIX 5L, the VIOS includes a feature that accepts hot plugging devices, such as disks and PCI adapters into the server, and activating them for the partition without a reboot.

Prior to starting, an empty system slot must be assigned to the VIOS partition on the HMC. This task can be done through dynamic LPAR operations, but the VIOS partition profile must also be updated so that the new adapter is configured to the VIOS after a reboot.

To begin, use the **diagmenu** command to get into the VIOS diagnostic menu. This menu is very similar to the AIX 5L diagnostic menu and gives you the same four options at the beginning screen:

- Diagnostic Routines
- Advanced Diagnostic Routines
- Task Selection
- Resource Selection

The Hot Plug Tasks selection is under the Task Selection option of the menu. Under this menu selection, the choice of PCI hot plug tasks, RAID hot plug devices, and the SCSI and SCSI RAID hot plug manager are presented, as shown in Example 5-13.

Example 5-13 Hot Plug Task diagmenu

```
Hot Plug Task
801004

Move cursor to desired item and press Enter.

  PCI Hot Plug Manager
  RAID Hot Plug Devices
  SCSI and SCSI RAID Hot Plug Manager
```

The PCI menu is used for adding, identifying, or replacing PCI adapters in the system that are currently assigned to the VIOS. The RAID hot plug devices option is used for adding RAID enclosures that will be connected to a SCSI RAID adapter. The SCSI and SCSI RAID manager menu is used for disk drive addition or replacement and SCSI RAID configuration.

Adding a PCI hot plug adapter

The test VIOS has the following PCI Adapter configuration:

```
# Slot                    Description                          Device(s)
U787B.001.DNW0974-P1-C1   PCI-X capable, 64 bit, 133MHz slot   ent0
U787B.001.DNW0974-P1-C2   PCI-X capable, 64 bit, 133MHz slot   Empty
U787B.001.DNW0974-P1-C3   PCI-X capable, 64 bit, 133MHz slot   Empty
U787B.001.DNW0974-P1-C4   PCI-X capable, 64 bit, 133MHz slot   sisioa0
U787B.001.DNW0974-P1-C5   PCI-X capable, 64 bit, 133MHz slot   pci5 lai0
```

For this VIOS, slots C2 and C3 are both empty hot-plug PCI adapter slots. Slots C1, C4, and C5 are also hot-plug adapter slots with an Ethernet card in C1, a SCSI RAID adapter in C4, and a graphics card in C5.

To add an adapter, choose Add a PCI Hot Plug Adapter from the menu and a list of available slots will be presented, as in Example 5-14.

Example 5-14 Add a PCI Hot Plug Adapter screen

```
                        Add a PCI Hot Plug Adapter

Move cursor to desired item and press Enter. Use arrow keys to scroll.

  # Slot                    Description                          Device(s)
  U787B.001.DNW0974-P1-C2   PCI-X capable, 64 bit, 133MHz slot   Empty
  U787B.001.DNW0974-P1-C3   PCI-X capable, 64 bit, 133MHz slot   Empty
```

For this example, slot C2 is chosen and the output is displayed in Example 5-15.

Example 5-15 Add a PCI Hot Plug Adapter to slot 2

```
Command: running       stdout: yes           stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.
```

The rest of the adapter addition is performed exactly as in a stand-alone AIX 5L LPAR with the following tasks occurring:

- Indicator light for adapter placement flashes
- Adapter installation
- Finish adapter installation task using the **diagmenu** command

When the adapter has been added successfully, the **diagmenu** command will display the following:

```
Add Operation Complete.
```

After this step, the **cfgdev** command must be run to configure the device for the VIOS. Then, when a list of the available hot-plug adapters is selected in the **diagmenu** output, the new Fibre Channel card appears in slot C2:

```
# Slot                    Description                        Device(s)
U787B.001.DNW0974-P1-C1   PCI-X capable, 64 bit, 133MHz slot  ent0
U787B.001.DNW0974-P1-C2   PCI-X capable, 64 bit, 133MHz slot  fcs0
U787B.001.DNW0974-P1-C3   PCI-X capable, 64 bit, 133MHz slot  Empty
U787B.001.DNW0974-P1-C4   PCI-X capable, 64 bit, 133MHz slot  sisioa0
U787B.001.DNW0974-P1-C5   PCI-X capable, 64 bit, 133MHz slot  pci5 lai0
```

This Fibre Channel card is now ready to be attached to a SAN and have LUNs assigned to the VIOS for virtualization.

Adding a SCSI hot swappable disk

To add a hot swappable SCSI disk to the RAID enclosure of a VIOS, you need to use the **diagmenu** command again. Under the task selection, Hot Plug Tasks menu, choose the SCSI and SCSI RAID Hot Plug Manager menu. This gives the output shown in Example 5-16.

Example 5-16 SCSI and SCSI RAID Hot Plug menu

```
Make selection, use Enter to continue.

  List Hot Swap Enclosure Devices
    This selection lists all SCSI hot swap slots and their contents.
  Identify a Device Attached to a SCSI Hot Swap Enclosure Device
    This selection sets the Identify indication.
  Attach a Device to an SCSI Hot Swap Enclosure Device
    This selection sets the Add indication and prepares
    the slot for insertion of a device.
  Replace/Remove a Device Attached to an SCSI Hot Swap Enclosure Device
    This selection sets the Remove indication and prepares
    the device for removal.
  Configure Added/Replaced Devices
    This selection runs the configuration manager on the
    parent adapter where devices have been added or replaced.
```

After selecting the Attach a Device to a SCSI Hot Swap Enclosure Device menu, you will be presented with a list of free slots in the RAID enclosure in which to plug the SCSI disk.

The following is a list of empty SCSI Hot Swap Enclosure device slots:

```
slot  3                              [empty slot]
```

In this example, slot 3 is the only available empty slot. When you choose the empty slot, a screen will come up telling you to attach the device and hit Enter to return the LED to the normal state of operation

After this, return to the previous menu and choose the List Hot Swap Enclosure Devices option in the menu and you will see the output shown in Example 5-17.

Example 5-17 Hot Swap Enclosure Devices list

```
The following is a list of SCSI Hot Swap Enclosure Devices. Status information
about a slot can be viewed.

Make selection, use Enter to continue.

                 U787B.001.DNW0974-
  ses0             P1-T14-L15-L0
     slot  1       P1-T14-L8-L0          hdisk3
     slot  2       P1-T14-L5-L0          hdisk2
     slot  3                             [populated]
     slot  4       P1-T14-L3-L0          hdisk0
```

Slot 3 still shows up without an hdisk in it, but as populated. Run the **cfgdev** command to initialize the disk on the VIOS command line; after returning to this menu, you will see that hdisk1 now appears in slot 3:

```
                 U787B.001.DNW0974-
  ses0             P1-T14-L15-L0
     slot  1       P1-T14-L8-L0          hdisk3
     slot  2       P1-T14-L5-L0          hdisk2
     slot  3       P1-T14-L4-L0          hdisk1
     slot  4       P1-T14-L3-L0          hdisk0
```

5.4.3 Recovering from a failed VIOS disk

If a disk failure occurs on a Virtual I/O Server, there are several steps that you need to perform before full recovery is achieved. While getting the failed disk replaced is the key step to recovery, cleaning up the VIOS and the client partition also needs to happen before the rebuilding step is taken.

The **errlog** command is used to view what errors have occurred the VIOS. In this case, the following report is produced:

```
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
613E5F38   0114201370 P H LVDD           I/O ERROR DETECTED BY LVM
8647C4E2   0114201370 P H hdisk1         DISK OPERATION ERROR
613E5F38   0114201370 P H LVDD           I/O ERROR DETECTED BY LVM
8647C4E2   0114201370 P H hdisk1         DISK OPERATION ERROR
613E5F38   0114201370 P H LVDD           I/O ERROR DETECTED BY LVM
8647C4E2   0114201370 P H hdisk1         DISK OPERATION ERROR
```

As displayed in the error report, hdisk1 has failed on the VIOS. This disk was not part of a RAID set and was not mirrored, so its failure constitutes a fatal error on the VIOS, as logical volumes were lost.

Note: The use of a SCSI RAID set for client volume groups is recommended to prevent possible failures due to the loss of a SCSI disk.

The volume group clientvg had logical volumes for the root volume groups of clients plus a logical volume for the NIM volume group of a client.

The virtual SCSI target and device are still available on the system, as they are not directly connected to the hard disk that failed:

```
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
vcl1nimvg       Available  Virtual Target Device - Logical Volume
vcl1rootvg      Available  Virtual Target Device - Logical Volume
vcl2rootvg      Available  Virtual Target Device - Logical Volume
```

However, the logical volumes that were located on the failing disk are now gone, and rebuilding of the logical volumes and the virtual SCSI target devices is required.

Run **diagmenu** and select **Task Selection** → **Hot Plug Task** → **SCSI and SCSI RAID Hot Plug Manager** to see the following option:

```
Replace/Remove a Device Attached to an SCSI Hot Swap Enclosure Device
```

This option starts the procedure for replacing the faulty disk. The sequence of identifying the disk, replacing, and completing the task is completed in a similar fashion to an AIX 5L partition.

After the disk is replaced and brought into the volume group, new logical volumes for the client root volume groups need to be created. The old virtual SCSI target devices need to be removed using the **rmdev -dev vscsi -target-device** command and then, using the **mkvdev** command, new target devices need to be created similar to the original setup:

```
mkvdev -vdev {new logical volume} -vadapter {vhost}
```

On the client partition, the following output displays the state of the root volume group:

```
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     32    64    2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/syncd    N/A
hd4                 jfs2       1     2     2    open/syncd    /
hd2                 jfs2       37    74    2    open/stale    /usr
hd9var              jfs2       1     2     2    open/stale    /var
hd3                 jfs2       3     6     2    open/stale    /tmp
hd1                 jfs2       1     2     2    open/stale    /home
hd10opt             jfs2       3     6     2    open/stale    /opt
```

As in an AIX 5L partition, a cleanup of the volume group is required. Remove the mirror from the missing disk followed by removing the disk from the volume group itself. Remove the definition of the hdisk and then run the **cfgmgr** command to configure the disk back to the machine. This cleanup procedure is required because the logical volume that has been created on the VIOS contains no volume group data and will show up as a new hdisk after the **cfgmgr** command completes.

Since the mapping has been completed on the VIOS, the virtual SCSI disk will be available and ready for use. A standard rootvg mirroring process is applied from this point.

Recovering from a failed VIOS

If for some reason a VIOS reboots without the clients being prepared, a volume group cleanup procedure will need to be performed. Unlike a failed SCSI device on a VIOS, the client volume group and logical volumes are still intact.

When the VIOS comes back online, the previously missing client virtual SCSI disk returns. Volume groups that used virtual SCSI disks from this VIOS will show logical volumes in a open/stale state, as the disks hosted by the VIOS will have be marked as missing.

The correct way to recover this situation is to run the **varyonvg** command for all volume groups affected, including rootvg. This command restores the previously missing disk to a PV State of active and starts synchronizing the logical volumes.

No other cleanup on the client partition is required after this task.

Failed paths with MPIO

If either a VIOS or its Fibre Channel card has a failure and MPIO is configured on the client partition, you will notice a failed path when you run the **lspath** command:

```
# lspath
Failed  hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

When the VIOS is rebooted or the Fibre Channel adapter is replaced, the failed path automatically changes its status to Enabled. If this does not occur, you need to issue the **chpath -s Enabled** command to force the enablement of the path. Without this command, if the second VIOS should experience a failure, the I/O to the client partition will fail, as there are no valid paths to its disks.

5.4.4 Maintenance considerations and recommendations

The following section covers general maintenance of the VIOS with some basic day-to-day scenarios that are often encountered.

Increasing a client partition volume group

Virtual disks exported from a Virtual I/O Server can be either a logical volume or a whole disk. Whole disks could be local or SAN disks but only SAN disks can be expanded.

Important: At the time of writing, a logical volume that is used for creating virtual SCSI devices and that spans multiple physical disks is not recommended.

1. Use the **extendlv** command to expand a logical volume.

   ```
   Example to extend the logical volume db_lv by 5 Gigabytes:
   extendlv db_lv 5G
   ```

   a. Use your storage manager application to extend a SAN disk.

From the client side, additional virtual disk space could be achieved in the following ways:

1. Get additional disks from the Virtual I/O Server.

2. Get a larger disk from the Virtual I/O Server and use the **migratepv** command to migrate from the smaller disk to the larger.
3. Have the back end logical volume or SAN disk extended and use the **chvg -g <volumegroup name>** command to get the LPAR to recognize the size change.

See Example 5-18 for the client operation.

Note: VIOS V1.3 will automatically discover a change in the size of a disk in the Virtual I/O Server. There is no need to unconfigure and configure the virtual device to pick up the size change.

Note: The rootvg volumegroup cannot be changed using the **chvg -g** command. If you want a larger disk for rootvg, you must either add a disk or use the **migratepv** command to migrate to a larger disk. However, this could slow down the LPAR during the operation.

Note: A reboot will not pick up an extended size of a rootvg disk (even if the **bootinfo -s** command will report the new size.

Example 5-18 Client operation for extending a disk.

```
# lsvg dbdatavg
VOLUME GROUP:       dbdatavg                VG IDENTIFIER:
00c5e9de00004c000000010e14c01dde
VG STATE:           active                  PP SIZE:        64 megabyte(s)
VG PERMISSION:      read/write              TOTAL PPs:      79 (5056 megabytes)
MAX LVs:            256                     FREE PPs:       68 (4352 megabytes)
LVs:                2                       USED PPs:       11 (704 megabytes)
OPEN LVs:           2                       QUORUM:         2
TOTAL PVs:          1                       VG DESCRIPTORS: 2
STALE PVs:          0                       STALE PPs:      0
ACTIVE PVs:         1                       AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                    MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)         AUTO SYNC:      no
HOT SPARE:          no                      BB POLICY:      relocatable

# chvg -g dbdatavg

# lsvg dbdatavg
VOLUME GROUP:       dbdatavg                VG IDENTIFIER:
00c5e9de00004c000000010e14c01dde
VG STATE:           active                  PP SIZE:        64 megabyte(s)
VG PERMISSION:      read/write              TOTAL PPs:      95 (6080 megabytes)
MAX LVs:            256                     FREE PPs:       84 (5376 megabytes)
```

```
LVs:                2                   USED PPs:       11 (704 megabytes)
OPEN LVs:           2                   QUORUM:         2
TOTAL PVs:          1                   VG DESCRIPTORS: 2
STALE PVs:          0                   STALE PPs:      0
ACTIVE PVs:         1                   AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)     AUTO SYNC:      no
HOT SPARE:          no                  BB POLICY:      relocatable
```

Attention: You do not have to do a **varyoffvg** and a subsequent **varyonvg** of the volumegroup to pick up the size change. If you do a **varyonvg <volumegroup name>**, you will a message similar to this:

```
0516-1434 varyonvg: Following physical volumes appear to be grown in
size.
```

Run the **chvg** command to activate the new space.

Important: At the time of writing, a logical volume that is used for creating virtual SCSI devices and that spans multiple physical disks is not recommended.

Updating the VIOS

The **updateios** command is used for applying upgrades to the VIOS software. An example of its use is shown in 5.4.1, “Concurrent software updates for the VIOS” on page 294.

Example 5-19 shows the usage message of the command.

Example 5-19 updateios command example

```
$ updateios
Command requires option "-accept -cleanup -dev -remove -reject".

Usage: updateios -dev Media [-f] [-install] [-accept]
       updateios -commit | -reject [-f]
       updateios -cleanup
       updateios -remove {-file RemoveListFile | RemoveList}
```

The **updateios** command does not commit the upgrade it is installing, but rather puts it in an applied state. Before the next upgrade can be applied, the previous upgrade needs to be either committed or rejected, which is done with the -commit and -reject flags with the command.

In some cases, a reboot of the VIOS partition may be required to complete the upgrade. Other updates may also require the upgrade of the system firmware, which could include an entire system outage for a reboot to complete the firmware upgrade.

For an example of how to update the VIOS without affecting client application availability, see 5.4.1, “Concurrent software updates for the VIOS” on page 294.

Snap collection for IBM support

When contacting IBM support, there are two things you should have ready prior to actually opening a call:

- The exact configuration of your system.
- Run the **snap** command from the VIOS.

The rest of the snap collection proceeds similar to a standard AIX 5L **snap** command collection. The file produced is the standard snap.pax.Z file, which can be transferred to the IBM support site given to you by your IBM support representative.

When the -general flag is used with the **snap** command, only the general configuration of the system is collected. Without the -general flag, a full system scan is performed and all information, including storage layout, security, install results, network configuration and virtual configurations, are captured in the snap.pax.Z file.

Dual Virtual I/O Server considerations

With the redundancy of dual Virtual I/O Servers comes the added system maintenance requirement to keep the redundancy working. Just as with HACMP, all changes made to the dual VIOS that are related to providing redundancy to a client partition should be tested. While the strictness of HACMP testing does not need to be followed, testing after major changes or additions should be performed to verify that the redundancy required is actually achieved.

Backing up the dual VIOS also requires some thought. When backing up to a optical device, where VIOS maintains control of the DVD, requires some planning, as the client partitions will also use that VIOS for access to the optical device.

Assigning one VIOS as a client to the other for the DVD is not an option, because as of Virtual I/O Server 1.3 there can be only server SCSI adapters created in a Virtual I/O Server.

5.4.5 Checking and fixing the configuration

When using LVM mirroring between disks from two Virtual I/O Servers, a reboot of one Virtual I/O Server makes one disk *missing* and stale partitions have to be synchronized with the **varyonvg** command when the server is rebooted.

When using NIB for network redundancy, the backup does not fall back to the primary adapter until the backup adapter fails. This holds for virtual adapters and AIX 5L V5.3-ML03 or higher even if Automatically Recover to Main Channel is set to yes due to the fact that no link up event is received during reactivation of the path. This is because virtual Ethernet adapters are always up. If you configure NIB to do load balancing, you may want the NIB to be on the primary channel.

If the settings are correct, MPIO will not require any special attention when a Virtual I/O Server is restarted, but a failed path should be checked.

Checking and fixing these things in a system with many partitions is time consuming and prone to errors.

Figure 5-21 on page 295 is a *sample* script that you can fit to your needs that will check and fix the configuration for:

- Redundancy with dual Virtual I/O Servers and LVM mirroring
- Redundancy with dual Virtual I/O Servers, Fibre Channel SAN disks and AIX MPIO
- Network redundancy using Network Interface Backup

The script should reside on each VIO Client and if using **dsh** (distributed shell), it should also be located in the same directory on each VIO Client.

Since it is local to the VIO Client, it can be customized for the individual client.

It could be executed at regular intervals using **cron**.

A better way would be to run the script on all required target partitions in parallel using **dsh** (distributed shell) from a NIM or admin server after a Virtual I/O Server reboot.

Note: The **dsh** command is installed by default in AIX and is part of CSM. However, use of the full function clustering offered by CSM requires a license. See the AIX 5L documentation for **dsh** command information.

Note: You can use **dsh** based on **rsh**, **ssh**, or Kerberos authentication as long as **dsh** can run commands without being prompted for a password.

See Example 5-20 for information about how to run fixdualvios.ksh in parallel on partitions dbserver, appserver, and nim.

Example 5-20 Using a script to update partitions

```
# dsh -n dbserver,appserver,nim /tmp/fixdualvios.ksh | dshbak >\
/tmp/fixdualvio.out
```

Tip: Use the DSH_LIST=<file listing lpars> variable so you do not have to type in the names of the target LPARs when using **dsh**.

Tip: Use the DSH_REMOTE_CMD=/usr/bin/ssh variable if you use **ssh** for authentication.

Tip: The output file /tmp/fixdualvio.out will reside on the system running the **dsh** command.

The **dshbak** command will group the output from each server.

Example 5-21 shows how to run the script and the output listing from the partitions dbserver, appserver, and nim.

Example 5-21 Running the script and listing output

```
# export DSH_REMOTE_CMD=/usr/bin/ssh
# export DSH_LIST=/root/nodes
# dsh /tmp/fixdualvios.ksh|dshbak > /tmp/fixdualvios.out

HOST: appserver
---------------
1 Checking if Redundancy with dual VIO Server and LVM mirroring is
being used.
Redundancy with dual VIO Server and LVM mirroring is NOT used.
No disk has missing status in any volumegroup.

2 Checking if Redundancy with dual VIO Server, Fiber Channel SAN disks
and AIX MPIO is being used.
Status:
```

```
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
hdisk1 has vscsi0 with Failed status. Enabling path.
paths Changed
New status:
Enabled hdisk1 vscsi0
Enabled hdisk1 vscsi1

3 Checking if Network redundancy using Network interface backup is
being used.
EtherChannel en2 is found.
Backup channel is being used. Switching back to primary.
Active channel: primary adapter

HOST: dbserver
--------------
1 Checking if Redundancy with dual VIO Server and LVM mirroring is
being used.
Redundancy with dual VIO Server and LVM mirroring is NOT used.
No disk has missing status in any volumegroup.

2 Checking if Redundancy with dual VIO Server, Fiber Channel SAN disks
and AIX MPIO is being used.
hdisk0 has vscsi0 with Failed status. Enabling path.
paths Changed
New status:
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1

3 Checking if Network redundancy using Network interface backup is
being used.
EtherChannel en2 is found.
Backup channel is being used. Switching back to primary.
Active channel: primary adapter

HOST: nim
---------
1 Checking if Redundancy with dual VIO Server and LVM mirroring is
being used.
Redundancy with dual VIO Server and LVM mirroring is being used.
Checking status.
No disk in rootvg has missing status.
No disk has missing status in any volumegroup.
```

```
2 Checking if Redundancy with dual VIO Server, Fiber Channel SAN disks
and AIX MPIO is being used.
Redundancy with dual VIO Server, Fiber Channel SAN disks and AIX MPIO
is NOT used.

3 Checking if Network redundancy using Network interface backup is
being used.
EtherChannel en2 is found.
Backup channel is being used. Switching back to primary.
Active channel: primary adapter
```

Note: The reason for the Failed status of the paths is that the hcheck_interval parameter had not been set on the disks yet.

Listing of the fixdualvio.ksh script

```
#!/bin/ksh
#set -x
#
# This script will check and restore the dual VIO Server
# configuration for partitions served from two VIO Servers after
# one VIO Server has been unavailable.
# The script must be tailored and TESTED to your needs.
#
# Disclaimer
# IBM DOES NOT WARRANT OR REPRESENT THAT THE CODE PROVIDED IS COMPLETE OR UP-TO-DATE.
IBM DOES NOT WARRANT, REPRESENT OR IMPLY RELIABILITY, SERVICEABILITY OR FUNCTION OF THE
CODE.  IBM IS UNDER NO OBLIGATION TO UPDATE CONTENT NOR PROVIDE FURTHER SUPPORT.

# ALL CODE IS PROVIDED "AS IS," WITH NO WARRANTIES OR GUARANTEES WHATSOEVER.  IBM
EXPRESSLY DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ALL EXPRESS, IMPLIED,
STATUTORY AND OTHER WARRANTIES, GUARANTEES, OR REPRESENTATIONS, INCLUDING, WITHOUT
LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND
NON-INFRINGEMENT OF PROPRIETARY AND INTELLECTUAL PROPERTY RIGHTS.  YOU UNDERSTAND AND
AGREE THAT YOU USE THESE MATERIALS, INFORMATION, PRODUCTS, SOFTWARE, PROGRAMS, AND
SERVICES, AT YOUR OWN DISCRETION AND RISK AND THAT YOU WILL BE SOLELY RESPONSIBLE FOR
ANY DAMAGES THAT MAY RESULT, INCLUDING LOSS OF DATA OR DAMAGE TO YOUR COMPUTER SYSTEM.

# IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY TYPE WHATSOEVER RELATED TO OR ARISING
FROM USE OF THE CODE FOUND HEREIN, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS
INTERRUPTION, LOST SAVINGS, LOSS OF PROGRAMS OR OTHER DATA, EVEN IF IBM IS EXPRESSLY
ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS EXCLUSION AND WAIVER OF LIABILITY
APPLIES TO ALL CAUSES OF ACTION, WHETHER BASED ON CONTRACT, WARRANTY, TORT OR ANY OTHER
LEGAL THEORIES.
#
# Assuming that the configuration may be using one or more of:
#   1 Redundancy with dual VIO Server and LVM mirroring.
#   2 Redundancy with dual VIO Server, Fiber Channel SAN disks and
```

```
#     AIX MPIO.
#   3 Network redundancy using "Network interface backup".
#
# Syntax: fixdualvio.ksh
#
#

#   1 Redundancy with dual VIO Server and LVM mirroring.
#
echo 1 Checking if "Redundancy with dual VIO Server and LVM mirroring" is being used.

# Check if / (hd4) has 2 copies
MIRROR=`lslv hd4|grep COPIES|awk '{print $2}'`
if [ $MIRROR -gt 1 ]
then
        # rootvg is most likely mirrored
        echo "Redundancy with dual VIO Server and LVM mirroring" is being used.
        echo Checking status.
        # Find disk in rootvg with missing status
        MISSING=`lsvg -p rootvg|grep missing|awk '{print $1}'`
        if [ "$MISSING" = "" ]
        then
                echo No disk in rootvg has missing status.
        else
                echo $MISSING has missing status.
#
# Restore active status and sync of rootvg
                echo Fixing rootvg.
                varyonvg rootvg
        fi
else
        echo "Redundancy with dual VIO Server and LVM mirroring" is NOT used.
fi
# Check now if ANY disk has missing status.
        ANYMISSING=`lsvg -o|lsvg -ip|grep missing|awk '{print $1}'`
        if [ "$ANYMISSING" = "" ]
        then
                echo No disk has missing status in any volumegroup.
        else
                echo $ANYMISSING has missing status. CHECK CAUSE!
        fi

#   2 Redundancy with dual VIO Server, Fiber Channel SAN disks and
#     AIX MPIO.
echo
echo 2 Checking if "Redundancy with dual VIO Server, Fiber Channel SAN disks and AIX
MPIO" is being used.
# Check if any of the disks have more than one path (listed twice)
MPIO=`lspath | awk '{print $2}' | uniq -d`
if [ $MPIO ]
then
        for n in $MPIO
        do
                # Check if this disk has a Failed path.
```

```
                STATUS=`lspath -l $n | grep Failed | awk '{print $1}'`
                if [ $STATUS ]
                then
                        ADAPTER=`lspath -l $n | grep Failed | awk '{print $3}'`
                        echo $n has $ADAPTER with Failed status. Enabling path.
                        chpath -s ena -l $n -p $ADAPTER
                        # Check new status
                        echo New status:
                        lspath -l $n
                else
                echo Status:
                        lspath -l $n
                fi

        done
else

echo "Redundancy with dual VIO Server, Fiber Channel SAN disks and AIX MPIO
"is NOT used.
fi


#   3 Network redundancy using "Network interface backup".
# Find out if this is being used and if so which interface number(s).

echo
echo 3 Checking if Network redundancy using "Network interface backup" is being used.

ECH=`lsdev -Cc adapter -s pseudo -t ibm_ech -F name | awk -F "ent" '{print $2}'`

if [ -z "$ECH" ]
then
echo No EtherChannel is defined.
else
# What is the status
        for i in $ECH
                do
                echo EtherChannel en$i is found.

                ETHCHSTATUS=`entstat -d en$i | grep Active | awk '{print $3}'`
                        if [ "$ETHCHSTATUS" = "backup" ]
                        then
                        # switch back to primary (requires AIX5.3-ML02 or higher)
                        echo Backup channel is being used. Switching back to primary.

                        /usr/lib/methods/ethchan_config -f en$i

                        # Check the new status
                        NEWSTATUS=`entstat -d en$i | grep Active | awk '{print $3}'`
                        echo Active channel: $NEWSTATUS adapter
                        #
                        else
                        echo Active channel: $ETHCHSTATUS adapter.
                        fi
```

```
#
                done

fi
exit
end
```

5.5 Monitoring a virtualized environment

Oscar Wilde once said “The truth is rarely pure and never simple”. This statement could be applied to monitoring system resource usage in a virtualized environment. In these environments, the amount of resources owned by a partition can change on-the-fly and this presents new challenges to both the developers of performance tools and those trying to interpret the results.

This section starts with some theory on how tools measure virtual resource usage before diving into the practicalities of using the tools and presenting some of the new AIX 5L performance-related commands.

5.5.1 Ask the right questions

Here are few of the questions that must be answered when designing performance monitoring tools in a virtualized environment. In many cases, there are several correct answers.

When using SMT, how do you measure the resource usage of the two logical processors? Is a logical processor that is using 50 percent of the physical processor 100% busy? When is an uncapped virtual-processor 100 percent busy? What should be reported if the configuration changes during the monitoring?

To help give an answer to these questions and others, the POWER5 family of processors implements a new performance-specific register called the Process Utilization Resource Register (PURR). The PURR tracks the real processor resource usage on a per thread or per partition level. The AIX 5L performance tools have been updated in AIX 5L V5.3. It had been modified to show these new statistics.

Traditional performance measurements were based on sampling, typically with a 100 Hz sample rate (each sample corresponds to a 10ms *tic*). Each sample is sorted into one of four categories:

user The interrupted code is outside the AIX 5L kernel.

sys The interrupted code is inside the AIX 5L kernel and the currently running thread is not waitproc.

iowait The currently running thread is waitproc and there is an I/O pending.

idle The currently running thread is waitproc and there is no I/O pending.

This traditional mechanism must stay unchanged to preserve binary compatibility with earlier tools.

This sampling-based approach breaks down in a virtualized environment, as the assumption that the dispatch cycle of each virtual processor is the same no longer holds true. A similar problem exists with SMT; if one thread is consuming 100 percent of the time on a physical CPU, sample-based reporting would report the system 50 percent busy (one processor at 100 percent, the other at 0 percent), but in fact the processor is really 100 percent busy.

5.5.2 Process Utilization Resource Register (PURR)

The PURR is simply a 64-bit counter with the same units for the timebase and decrementer registers that provide per-thread processor utilization statistics. Figure 5-24 on page 323 shows the logical CPUs and the relationship of the PURR registers within a single POWER5 processor (core) and the two hardware threads. With SMT enabled, each hardware thread is seen as a logical processor.

The *timebase* register shown in Figure 5-24 on page 323 is simply a hardware register that is incremented at each tic. The *decrementer* register provides periodic interrupts.

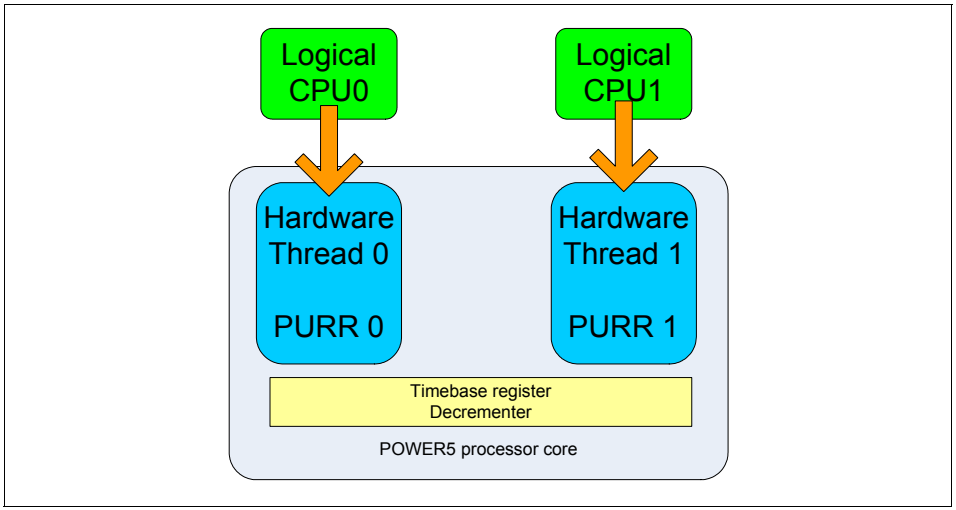


Figure 5-24 Per-thread PURR

At each processor clock cycle, one of the PURRs is incremented, either:

- The thread dispatching instructions
- The thread that last dispatched an instruction

The sum of the two PURRs equals the value in the timebase register. This approach is an approximation, as SMT allows both threads to run in parallel. It simply provides a reasonable indication of which thread is making use of the POWER5 resources; however, it does not provide a mechanism to distinguish the performance difference operating with SMT on from that with SMT off.

New PURR-based metrics

The new registers provide some new statistics.

SMT statistics

The ratio of *(delta PURR)/(delta timebase)* over an interval indicates the fraction of physical processor consumed by a logical processor. This is the figure returned by the **`sar -P ALL`** and **`mpstat`** commands.

The figure *(delta PURR/delta TB)*100* over an interval gives the previous figure as a percentage and can be interpreted as the percentage of dispatch cycles given to a logical processor or the percentage of physical processor consumed by a logical processor. This figure is returned by the **`mpstat -s`** command, which shows the SMT statistics.

CPU statistics in shared-processors partitions

In a shared-processor environment, the PURR measures the time that a virtual processor runs on a physical processor. The partition time, just like the processor, is virtual, with the POWER Hypervisor maintaining the virtual Time Base as the sum of the two PURRs. In shared processors with SMT off, the virtual time base is simply the value stored in the PURR.

Capped shared processors

For capped shared processors the calculation is as follows:

The *entitled PURR* over an interval is given as *entitlement * time base*.

The %user time over an interval is then given as:

*%user = (delta PURR in user mode/entitled PURR) * 100*

Uncapped shared processors

For uncapped shared processors, the calculations take the variable capacity into account. The *entitled PURR* in the above formula is replaced by the *consumed PURR* whenever the latter is greater than the entitlement.

Physical processor consumption for a shared processor

A partition's consumption of physical processor measured over an interval is simply the sum of the consumption of all its logical processors:

SUM(delta PURR/delta TB)

Partition entitlement consumption

A partition's entitlement consumption is simply the ratio of its physical processor consumption (PPC) to its entitlement expressed as a percentage:

*(PPC/ENT) * 100*

Shared-processor pool spare capacity

Unused cycles in the shared-processor pool are spent in the POWER Hypervisor's idle loop. The POWER Hypervisor enters this loop when all partition entitlements are satisfied and there are no partitions to dispatch. The time spent in the Hypervisor's idle loop, measured in tics, is called the Pool Idle Count or PIC. The shared-processor pool spare capacity over an interval is expressed as:

(delta PIC/delta TB)

and is measured in numbers of processors. Only partitions with shared-processor pool authority will be able to display this figure. An example is given in Example 5-36 on page 336 using the **lparstat** command.

Logical processor utilization

This is simply the sum of the traditional 10 ms, tic-based sampling of the time spent in %sys and %user. If this figure starts approaching 100 percent, it may indicate the partition could make use of additional virtual processors.

5.5.3 System-wide tools modified for virtualization

The AIX 5L tools providing system wide information, such as the **iostat**, **vmstat**, **sar**, and **time** commands, use the PURR-based statistics whenever SMT is enabled for the %user, %system, %iowait, and %idle figures.

When executing on a shared-processor partition, these commands adds two extra columns of information with:

- Physical processor consumed by the partition, shown as pc or %physc.
- Percentage of entitled capacity consumed by the partition, shown as ec or %entc.

This is shown in Example 5-22 and Example 5-23.

Example 5-22 iostat with SMT in capped shared-processor partition

```
# iostat -t 2 4

System configuration: lcpu=2 ent=0.50

tty:    tin      tout    avg-cpu: % user % sys % idle % iowait physc %
entc
        0.0      19.3                8.4  77.6   14.0      0.1   0.5
99.9
        0.0      83.2                9.9  75.8   14.2      0.1   0.5
99.5
        0.0      41.1                9.5  76.4   13.9      0.1   0.5
99.6
        0.0      41.0                9.4  76.4   14.1      0.0   0.5
99.7
```

Example 5-23 sar with SMT in capped shared-processor partition

```
# sar -P ALL 2 2

AIX dbserver 3 5 00C5E9DE4C00    10/11/06

System configuration: lcpu=2 ent=0.20

20:13:48 cpu    %usr    %sys    %wio   %idle   physc   %entc
20:13:50  0       19      71       0       9    0.31    61.1
          1        2      75       0      23    0.19    38.7
          -       13      73       0      15    0.50    99.8
20:13:52  0       21      69       0       9    0.31    61.1
          1        2      75       0      23    0.20    39.0
          -       14      71       0      15    0.50   100.2

Average   0       20      70       0       9    0.31    61.1
          1        2      75       0      23    0.19    38.9
          -       13      72       0      15    0.50   100.0
```

Logical processor tools

The logical processor tools are the **mpstat** and the **sar -P ALL** commands. When running in a partition with SMT enabled, these commands add the column Physical Processor Fraction Consumed - *(delta PURR/delta TB)*, shown as

physc. This shows the relative split of physical processor time of each of the two logical processors.

When running in shared processor partition, these commands add a new column, Percentage of Entitlement Consumed *((PPFC/ENT)*100)* shown as %entc. This figure gives relative entitlement consumption for each logical processor expressed as a percentage.

The **mpstat -s** command in Example 5-24 displays the virtual and logical processors and their load.

Example 5-24 mpstat SMT mode

```
# mpstat -s 2 2

System configuration: lcpu=8 ent=0.5

     Proc0            Proc2            Proc4            Proc6
     49.94%            0.03%            0.03%            0.03%
  cpu0     cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 24.98%  24.96%   0.01%   0.01%   0.01%   0.01%   0.01%   0.01%
--------------------------------------------------------------------------
--------
     Proc0            Proc2            Proc4            Proc6
     49.90%            0.03%            0.03%            0.03%
   cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 25.01%  24.89%   0.02%   0.01%   0.01%   0.01%   0.01%   0.01%
```

5.5.4 The topas command

The **topas** command CPU screen now includes information about the physical processor (Physc) and entitlement (%Entc) consumption by the partition, as shown in Example 5-25.

Example 5-25 topas default screen

```
Topas Monitor for host:    dbserver            EVENTS/QUEUES    FILE/TTY
Wed Oct 11 13:54:50 2006   Interval:  2        Cswitch     181  Readch    0.0G1
                                               Syscall     249  Writech   0.0G6
Kernel   48.1   |##############              |  Reads        45  Rawin        0
User      0.5   |#                           |  Writes       49  Ttyout      272
Wait      8.7   |###                         |  Forks         1  Igets        0
Idle     42.7   |#############               |  Execs         0  Namei        7
Physc =  0.11                   %Entc=  55.7  Runqueue    1.0  Dirblk       0
                                               Waitqueue   1.0
Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out
en2       0.9     6.1     3.5     0.4     0.5  PAGING           MEMORY
lo0       0.1     1.0     1.0     0.0     0.0  Faults      884  Real,MB     768
                                               Steals        0  % Comp     39.2
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  PgspIn        0  % Noncomp  11.4
hdisk1  100.1   9903.4    39.9     0.0  9903.4  PgspOut       0  % Client   11.4
                                               PageIn        0
Name            PID  CPU%  PgSp Owner           PageOut    4848  PAGING SPACE
topas        307238   0.1   1.3 root            Sios       5968  Size,MB     512
getty        188564   0.0   0.4 root                             % Used      1.0
ksh          290976   0.0   0.6 root            NFS (calls/sec)  % Free     98.9
gil           65568   0.0   0.1 root            ServerV2       0
rmcd         323782   0.0   2.4 root            ClientV2       0   Press:
nfsd         368824   0.0   0.2 root            ServerV3       0   "h" for help
rpc.lock     356540   0.0   0.2 root            ClientV3       0   "q" to quit
```

The **topas** command has a new split-screen mode with the -L switch or the **L** command. The upper section shows a subset of the **lparstat** command statistics while the lower part shows a sorted list of logical processors with a number of the **mpstat** command figures, as shown in Example .

Example 5-26 topas LPAR monitor screen

```
Interval:    2    Logical Partition: DB_Server         Wed Oct 11 14:32:52 2006
Psize:       -               Shared SMT  ON           Online Memory:   768.0
Ent: 0.20                     Mode: UnCapped          Online Logical CPUs:  4
Partition CPU Utilization                             Online Virtual CPUs:  2
%usr %sys %wait %idle physc  %entc %lbusy    app    vcsw    phint   %hypv   hcalls
   0    2     0    98   0.0   2.80   0.00      -     308        0       0        0
================================================================================
LCPU  minpf majpf  intr    csw   icsw runq lpa  scalls usr sys _wt idl   pc   lcsw
Cpu0      0     0   183    158     81    0 100      52   6  72   0  22 0.00    134
Cpu1      0     0    11      0      0    0   0       0   0   5   0  95 0.00    134
Cpu2      0     0    10      0      0    0   0       0   0  26   0  74 0.00     20
```

```
Cpu3      0     0    10      0      0    0   0       0   0  28   0  72 0.00     20
```

The **topas** command also has a new -D switch or **D** command to show disk statistics that take virtual SCSI disks into account, as shown in Example 5-27.

Example 5-27 topas disk monitor screen

```
Topas Monitor for host: dbserver Interval:   2    Wed Oct 11 14:01:00 2006
================================================================================
Disk    Busy%  KBPS    TPS   KB-R   ART  MRT   KB-W   AWT  MWT   AQW   AQD
hdisk0  47.1   3.2K  138.4   1.9K  4.7  21.1   1.3K  7.2  17.6   9.5   7.6
```

A cross-partition view of system resources is available with the **topas -C** command. This command will only see partitions running AIX 5L V5.3 ML3 or later; without specific configuration it will not show Virtual I/O Server partitions. Linux partitions cannot be shown at all at the time of writing. An example of the output of this command is shown in Example 5-28.

Example 5-28 topas cross-partition monitor screen

```
Topas CEC Monitor             Interval:  10 Wed Oct 11 14:10:30 2006
Partitions     Memory (GB)               Processors
Shr:  3        Mon: 2.0  InUse: 1.0      Shr:0.6  PSz:  2   Shr_PhysB:  0.02
Ded:  0        Avl:   -                  Ded:  0  APP:  2.0 Ded_PhysB:  0.00

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared-----------------------------------
appserver    A53 U 0.8 0.3  4   0  2  0 97   0.01  0.20   3.9  312   0
nim          A53 C 0.5 0.3  4   0  1  0 98   0.01  0.20   3.2  342   1
dbserver     A53 U 0.8 0.3  4   0  1  0 98   0.01  0.20   2.9  336   1
-----------------------------------dedicated--------------------------------
```

The **topas -C** command screen is split into header and partition regions. The partition region is distinguishes between shared and dedicated processor partitions.

The header section contains the global CEC information is comprised of three columns. The Partitions column lists how many partitions of each type **topas** found on the CEC. The Memory column shows the total and used memory in the monitored partitions measured in gigabytes. The Processor column shows the monitored processor types, shared and dedicated. The PSz shows the shared processor pool size when the partition has shared processor pool authority. The Shr_PhysB and Ded_PhysB values are comparable to the physc column in the **lparstat** output, but it excludes the idle time.

The partition section lists all the partitions the topas command can find in the CEC. The OS column indicates the type of operating system. In Example , the A53 indicates AIX 5L V5.3. The M column shows the partition mode: shared,

capped, and uncapped. The Mem and the InU columns show the configured and in use memory, respectively, measured in gigabytes. The Lp column show how many logical processors are configured. In Example on page 329, all partitions have one virtual processor with SMT enabled and so have two logical processors. The Us, Sy, Wa, and Id columns show the percent user, system, wait and idle times. The PhysB column is the same as in the summary section. The Ent, %Ent, Vcsw, and PhI columns are only applicable to shared-processor partitions. The Ent column shows the entitled capacity, the %EntC shows the percentage of the entitled capacity used, the Vcsw shows the virtual context switch rate, and the PhI column shows the phantom interrupt rate.

Typing the letter g in the **topas** command window expands the global information, as shown in Example 5-29. A number of fields are not completed in this example, such as the total available memory. These have been reserved for an update to this command that will allow **topas** to interrogate the HMC to determine their values. It is possible to manually specify some of these values on the command line.

Example 5-29 topas -C global information with the g command

```
Topas CEC Monitor             Interval:  10             Wed Oct 11 14:12:40 2006
Partition Info    Memory (GB)        Processor
Monitored  :  3   Monitored  : 2.0   Monitored  :0.6   Shr Physical Busy:  0.02
UnMonitored:  -   UnMonitored:   -   UnMonitored:  -   Ded Physical Busy:  0.00
Shared     :  3   Available  :   -   Available  :  -
Dedicated  :  0   UnAllocated:   -   UnAllocated:  -   Hypervisor
Capped     :  1   Consumed   : 1.0   Shared     :0.6   Virt. Context Switch:1245
Uncapped   :  2                      Dedicated  :  0   Phantom Interrupts  :   0
                                      Pool Size  :  2
                                      Avail Pool :  2.0

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared-------------------------------------
dbserver     A53 U 0.8 0.3  4   1  1  0 96   0.01  0.20   4.7  596   0
nim          A53 C 0.5 0.3  4   0  1  0 97   0.01  0.20   3.3  347   0
appserver    A53 U 0.8 0.3  4   0  1  0 98   0.00  0.20   2.4  302   0
-----------------------------------dedicated-----------------------------------
```

The **topas** command is also available on the Virtual I/O Server but the command options differ noticeably from those available for AIX. Example 5-30 on page 331 shows the options valid on the Virtual I/O Server.

Example 5-30 topas options on the Virtual I/O Server

```
$ topas -h
Usage: topas [-disks num_of_monitored_hot_disks]
             [-interval monitoring_interval_in_seconds]
             [-nets num_of_monitored_hot_networks_interfaces]
             [-procs num_of_monitored_hot_processes]
             [-wlms num_of_monitored_hot_WLM classes]
             [-cpus num_of_monitored_hot_CPUs]
             [-procsdisp | wlmdisp | cecdisp]

       Reports selected local system statistics.

       -disks        Specifies the number of disks to be monitored.
                     The default is 2.

       -interval     Sets the monitoring interval in seconds.
                     The default is 2 seconds.

       -nets         Specifies the number of hot network interfaces to be
                     monitored. The default is 2.

       -procs        Specifies the number of hot processes to be monitored.

       -wlms         Specifies the number of hot Work Load Management (WLM)
                     classes to be monitored.  The default is 2.

       -cpus         Specifies the number of hot CPUs to be monitored.
                     The default is 2.

       -procsdisp    Displays the full-screen process display.

       -wlmdisp      Displays the full-screen WLM class display.

       -cecdisp      Displays the full-screen cross-partition display.
```

Tip: The output of **topas -cecdisp** will probably not show values for the Virtual I/O Server itself unless the following has been done:

1. Log on to the Virtual I/O Server and execute the **oem_setup_env** command.
2. Run **# vi /etc/inetd.conf**

   and uncomment the xmquery stanza in this file. It should look like this:

   Before change

   ```
   # xmquery dgram   udp     wait    root    /usr/bin/xmtopas
   xmtopas -p3
   ```

   After change

   ```
   xmquery dgram   udp     wait    root    /usr/bin/xmtopas xmtopas
   -p3
   ```

3. Reload the inetd configuration with:

   ```
   # refresh -s inetd
   ```

The output of the **topas -cecdisp** command will now include the Virtual I/O Server, as shown in the output of Example 5-31.

Example 5-31 Output of **topas -cecdisp**

```
Topas CEC Monitor           Interval:  10              Fri Oct 13 14:25:52 2006
Partitions     Memory (GB)               Processors
Shr:  5        Mon: 3.0  InUse: 1.7      Shr:1.3  PSz:  2   Shr_PhysB:  0.03
Ded:  0        Avl:   -                  Ded:  0  APP:  2.0 Ded_PhysB:  0.00

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared-------------------------------------
-
nim          A53 C 0.5 0.3  4   0  1  0 98   0.01  0.20   3.1  478   0
VIO_Server2  A53 U 0.5 0.3  4   0  1  0 98   0.01  0.20   2.7  414   0
VIO_Server1  A53 U 0.5 0.4  4   0  0  0 99   0.01  0.50   1.1  369   0
dbserver     A53 U 0.8 0.3  4   0  1  0 98   0.01  0.20   2.5  302   0
appserver    A53 U 0.8 0.3  4   0  1  0 98   0.00  0.20   2.4  299   0
-----------------------------------dedicated----------------------------------
```

Continuous data collection and monitoring with topas

Data collection for continuous monitoring can also be enabled, so that statistics can be collected and an idea of the load on the system can be developed. There are a few steps necessary to enable that. Since this has to be done as root, on AIX 5L the procedure differs slightly from that one to be used on the Virtual I/O Server. Remember to activate the Shared Processor Pool Authority (in the

Partition Property screen on the HMC) for at least one partition, preferably the Virtual I/O Server partition if not already done. To enable data collection on the Virtual I/O Server, execute the **oem_setup_env** command and follow the guidelines for AIX 5L. We outline them below.

Go to the /usr/lpp/perfagent directory and execute the script **config_topas.sh** with the parameter add:

```
# cd /usr/lpp/perfagent
# ./config_topas.sh add
AIX Topas Recording (topas -R) enabled
# Sending nohup output to nohup.out.
```

This will add an entry to /etc/inittab and additionally start **topas** with the -R option for continuous data recording, collecting cross-partition CPU usage data.

For daily performance records, go to the /usr/lpp/perfagent directory and execute the script **config_aixwle.sh** with the parameter add:

```
# cd /usr/lpp/perfagent
# ./config_aixwle.sh add
Sending nohup output to nohup.out.
AIX Daily Recording Agent (xmwlm -L) enabled
#
```

This will add an entry to /etc/inittab as well as to the crontab and additionally start the **xmwlm** command with the -L option for daily data recording, which includes data from CPU usage as well as memory, disk, network, and kernel statistics.

The collected data of the **topas -R** command will be written to a file in /etc/perf. The name of the file is dependent on the date, so it should look like the one in Example 5-32

Example 5-32 Listing of directory /etc/perf after enabling topas data recording

```
# ls
   daily             wlm               xmtopas.log1      xmwlm.log1
   topas_cec.061013  xmservd.log1      xmtopas.log2      xmwlm.log2
```

These files can now be processed with the **topasout** command. The **topasout** command accepts the parameters shown in Example 5-33.

Example 5-33 Usage message of the topasout command

```
# topasout
Error, topasout: no filename or options specified
        topasout [-c|-s|-R daily|-R weekly] [-R detailed|-R summary|-R disk \
                 [ -i MM -b HHMM -e HHMM]] [xmwlm_recording|topas_recording]
        flags:  -c comma separated output format
                -s spreadsheet import format
                -R daily | weekly  WLE output report
                -R detailed | summary | disk (local recordings)
                -R detailed | summary (topas recordings)
                -i MM   split the recording reports into equal size time\
                        periods.Allowed Values (in minutes) are 5, 10, 15, 30,60
                -b HHMM  begin time in hours (HH) and minutes (MM).Range is \
                        between 0000 and 2400]
                -e HHMM  end time in hours (HH) and minutes (MM).Range is \
                        between 0000 and 2400 and is greater than the begin time
```

Example 5-34 shows an example of how to do the processes.

Example 5-34 topasout processing of recorded data - topas_recording

```
# topasout -c -i 5 -m min,max,mean,stdev,set,exp topas_recording
/etc/perf/topas_cec.061013
#ls - l /etc/perf
daily                wlm                  xmtopas.log2
topas_cec.061013     xmservd.log1         xmwlm.log1
topas_cec.061013_01  xmtopas.log1         xmwlm.log2
#
```

The file produced will always have a name based on the name of the file used as an argument to the **topasout** command and will be located in the directory /etc/perf. The data recorded to that file will contain cross-partition CPU measurements.

For daily performance statistics taken by the **xmwlm** command, performance data will be recorded to the directory /etc/perf/daily with a file name starting with **xmwlm** and can be processed by **topasout** as well. These files contain CPU, memory, disk, and network measurements as well as some kernel statistics.

Example 5-35 on page 335 shows how you can use the statistics.

Example 5-35 topasout processing of recorded data - xmwlm_recording

```
# topasout -c -i 5 -m min,max,mean,stdev,set,exp xwlm_recording
/etc/perf/daily/xmwlm.061016
# ls  /etc/perf/daily
xmwlm.061016     xmwlm.061016_01  xmwlm.061017
#
```

These files can be used as input files for further processing, for example, by the Java™ utility pGraph, or the Performance Graph Viewer by Federico Vagnini, an unsupported freeware tool available from:

http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/Performance+Graph+Viewer

This tool is able to produce HTML pages in batch mode and show graphs in interactive mode. Nmon recordings can be processed as well by this tool.

5.5.5 New monitoring commands on AIX 5L V5.3

AIX 5L V5.3 introduces some new monitoring commands specific to virtualized environments. These are discussed in this section.

The lparstat command

The **lparstat** command shows configuration and performance information for the partition in which it is run. This command works on all AIX 5L V5.3 systems, even those that do not support logical partitioning (though obviously without LPAR information). It supports dedicated and shared partitions, with or without SMT.

The **lparstat** command has four operational modes:

Monitoring The default, without any flags

Hypervisor summary With the -h switch

Hypervisor hcalls With the -H switch

System configuration With the -i switch

In all modes, except with the -i switch, **lparstat** prints a one-line summary of the system configuration before displaying the corresponding information.

The lparstat command monitoring mode

With no switches, the **lparstat** command monitors the partition's resource usage. Like many of the *stat* commands, the **lparstat** command takes optional

interval and count parameters. The output of the **lparstat** command is shown in Example 5-36.

Example 5-36 lparstat command monitoring mode

```
# lparstat 2 4

System configuration: type=Shared mode=Capped smt=On lcpu=2 mem=512
psize=6 ent=0.50

%user  %sys  %wait  %idle physc %entc  lbusy   app  vcsw phint
-----  ----  -----  ----- ----- -----  ------   ---  ---- -----
 14.2  85.3    0.6    0.0  0.50  99.6   97.7  5.49   311     1
 13.7  85.9    0.3    0.0  0.50  99.8   98.5  5.49   321     0
 13.9  85.9    0.2    0.0  0.50 100.2   98.2  5.49   319     0
 14.7  85.0    0.3    0.0  0.50 100.0   98.7  5.49   353     0
```

The first line, showing the system configuration, tells us that this partition is a capped shared-processor partition, type=Shared and mode=Capped, SMT is enabled, smt=On, and that there are two logical CPUs, lcpu=2, from which we can conclude that there is one virtual processor configured; the partition has 512 MB of memory, mem=512, and the partition's CPU entitlement is 0.5 of a physical CPU, ent=0.50.

The psize=6 output indicates that there are six processors in the shared pool. This information is only available to partitions that have shared-processor pool authority checked on their HMC configuration.

The %user, %sys, %wait, and %idle times show the traditional UNIX CPU statistics.

The physc column shows how many physical processors the partition is consuming. The %entc shows the percentage of the entitlement consumed by the partition and the lbusy column shows the percent occupation of the logical CPUs at the user and system level. From these three columns in Example 5-36, along with the information in the configuration summary line, we can deduce without too much difficulty that the partition is CPU bound, consuming all of its entitlement.

The vcsw column shows the number of virtual context switches. These correspond to a hardware preemption of a virtual processor. This figure gives one indication of the POWER Hypervisor load.

The phint column shows the number of phantom interrupts that the partition received. A phantom interrupt is an interrupt targeted to another partition that

shares the same physical processor. For example, one partition starts an I/O operation. While the partition is waiting for the I/O to complete, it cedes the physical processor to another partition. The I/O operation completes and the controller sends an interrupt to the requesting processor, but as the interrupted partition running on the processor is not the intended destination, the partition says “this is not for me” and the interrupt is queued by the POWER Hypervisor. Phantom interrupts are relatively cheap operations in terms of computing time and have little effect on system performance except when they occur in very large numbers.

The lparstat Hypervisor summary

With the -h flag, the **lparstat** command displays a summary of the POWER Hypervisor activity, as shown in Example 5-37.

Example 5-37 lparstat command Hypervisor summary

```
# lparstat -h 2 4

System configuration: type=Shared mode=Capped smt=On lcpu=2 mem=512 psize=6
ent=0.50

%user  %sys  %wait  %idle physc %entc  lbusy   app  vcsw phint  %hypv hcalls
-----  ----  -----  ----- ----- ----- ------   ---  ---- -----  ----- ------
 14.0  72.0    0.1   13.9  0.50 100.1   50.2  5.48   504     1   15.1   8830
 14.1  71.7    0.1   14.1  0.50  99.9   48.8  5.50   511     2   15.2   8696
 14.8  71.2    0.1   13.9  0.50 100.0   49.8  5.49   507     0   15.5   8826
 13.9  71.5    0.1   14.4  0.50  99.4   47.2  5.49   559     0   17.5   9381
```

The -h flag adds two columns to the monitoring (no flags) version of the **lparstat** command.

The %hypv column shows the amount of time spent in the POWER Hypervisor in each interval. Example 5-37 shows that between 15 and 17 percent of time was spent in the Hypervisor. If this figure becomes too high, you can use the -H to see which Hypervisor calls are causing the problem, as shown in the next section.

The hosted operating system uses Hypervisor calls or hcalls to communicate with the POWER Hypervisor. (Refer to 2.6, “Introduction to the POWER Hypervisor” on page 49). The hcalls column shows the total number of calls AIX 5L made to the POWER Hypervisor. In Example 5-37, there are approximately 4,500 calls per second. You can use this figure in conjunction with the %hypv column and the **lparstat -H** command shown in the next section to track the cause of high Hypervisor usage.

The lparstat command POWER Hypervisor hcalls

The -H flag of the **lparstat** command provides a detailed breakdown of the time spent in the POWER Hypervisor. For each hcall, it shows:

- The number of times it was called.
- The percentage of wall-clock time spent in the call.
- The percentage of POWER Hypervisor time spent in the call.
- The average and maximum time spent in the call.

In Example 5-38, it is the h_cede call that completely dominates the time spent in the POWER Hypervisor (94.8%); this is a fairly common scenario. Though the numbers appear large, the max call time for h_cede is given as 15463728, but this time is in nanoseconds, so it is only 15.5 ms and the average h_cede call is less than 0.2 ms.

Example 5-38 lparstat command Hypervisor hcalls

```
# lparstat -H 2 1

System configuration: type=Shared mode=Capped smt=On lcpu=2 mem=512 psize=6 ent=0.50

           Detailed information about Hypervisor Calls

Hypervisor        Number of    %Total Time   %Hypervisor   Avg Call    Max Call
  Call              Calls         Spent      Time Spent    Time(ns)    Time(ns)

remove                4052            0.2           1.1         430        5877
read                  2785            0.1           0.4         194        5322
nclear_mod               0            0.0           0.0           1           0
page_init             2418            0.2           1.4         906        6863
clear_ref              306            0.0           0.0         114        1159
protect                  0            0.0           0.0           1           0
put_tce                142            0.0           0.1        1140        2071
xirr                    67            0.0           0.0         874        3313
eoi                     66            0.0           0.0         729        1067
ipi                      0            0.0           0.0           1         405
cppr                    66            0.0           0.0         390         685
asr                      0            0.0           0.0           1           0
others                   0            0.0           0.0           1           0
enter                 6404            0.2           1.2         290        5641
cede                   834           14.5          94.8      173511    15463728
migrate_dma              0            0.0           0.0           1           0
put_rtce                 0            0.0           0.0           1           0
confer                   0            0.0           0.0           1        2434
prod                   152            0.0           0.0         463         777
get_ppp                  1            0.0           0.0        1980        2583
set_ppp                  0            0.0           0.0           1           0
purr                     0            0.0           0.0           1           0
pic                      1            0.0           0.0        2912        3849
bulk_remove            809            0.1           0.7        1381        7114
```

```
send_crq                61            0.0           0.1       2415        6143
copy_rdma                0            0.0           0.0          1           0
get_tce                  0            0.0           0.0          1           0
send_logical_lan         2            0.0           0.0       2600        6384
add_logicl_lan_buf       6            0.0           0.0        521        1733
--------------------------------------------------------------------------------
```

The lparstat command system configuration

The -i flag of the **lparstat** command produces an output in a significantly different format to that of the other **lparstat** commands. It gives a list of the partition configuration, as defined on the HMC. Example 5-39 shows the use of the this flag.

Example 5-39 lparstat command Hypervisor summary

```
# lparstat -i
Node Name                                  : vio_client2
Partition Name                             : VIO_client2
Partition Number                           : 1
Type                                       : Shared-SMT
Mode                                       : Capped
Entitled Capacity                          : 0.50
Partition Group-ID                         : 32769
Shared Pool ID                             : 0
Online Virtual CPUs                        : 1
Maximum Virtual CPUs                       : 32
Minimum Virtual CPUs                       : 1
Online Memory                              : 512 MB
Maximum Memory                             : 1024 MB
Minimum Memory                             : 128 MB
Variable Capacity Weight                   : 0        Capped partition
Minimum Capacity                           : 0.10
Maximum Capacity                           : 2.00
Capacity Increment                         : 0.01
Maximum Physical CPUs in system            : 16
Active Physical CPUs in system             : 8
Active CPUs in Pool                        : 6
Unallocated Capacity                       : 0.00
Physical CPU Percentage                    : 50.00%
Unallocated Weight                         : 0
--------------------------------------------------------------------------------
-------
```

Note: The variable capacity weight is zero in this example, as the partition mode is capped, as can be seen from line five of the output.

Note: The output format of the **lparstat** command varies, depending on the partition configuration (SMT on/off, shared or dedicated processors). This makes its parsing with **sed**, **awk**, **cut**, **perl**, and the like somewhat problematic.

The mpstat command

The **mpstat** command collects and displays performance statistics for all logical CPUs in a partition. The interpretation of many of the figures displayed by this command requires an understanding of POWER Hypervisor and the POWER5 processor. The **mpstat** command flags are shown in Table 5-1.

Table 5-1 mpstat command flags

Command	Flags	Function
mpstat	none	Displays the default statistics; a subset of the -a flag.
	-a	Displays all 29 logical processor statistics.
	-i	Displays the interrupt statistics; a subset of the -a flag.
	-d	Displays the processor affinity and migration statistics for AIX 5L threads and dispatching statistics for logical processors. A subset of the -a flag.
	-s	Shows the SMT usage statistics when SMT is enabled. This data is not shown with the -a flag.

When the **mpstat** command is invoked, it displays two sections of statistics. The first section displays the system configuration, which is displayed when the command starts and whenever there is a change in the system configuration. The second section displays the utilization statistics, which will be displayed in intervals and at any time the values of these metrics are deltas from a pervious interval.

An optional -w flag switches on wide screen output.

Example 5-40 shows the output of the **mpstat -a** command. Because the output is very wide, the result has been split into three sets of columns, with the CPU column repeated in each output. The meaning of each column is given in Table 5-2.

Example 5-40 mpstat command

```
# mpstat -a

System configuration: lcpu=2 ent=0.5

cpu    min    maj   mpcs   mpcr    dev   soft    dec     ph     cs   ics  bound
  0    134      3      0      0      2      0    105      0     90    47      0
  1     88      0      0      0      2     54    122      0      9     6      0
  U      -      -      -      -      -      -      -      -      -     -      -
ALL    222      3      0      0      4     54    227      0     99    53      0

cpu   rq   push S3pull  S3grd  S0rd  S1rd  S2rd  S3rd  S4rd  S5rd
  0   0      0      0      0  98.8   1.2   0.0   0.0   0.0   0.0
  1   0      0      0      0  90.8   9.2   0.0   0.0   0.0   0.0
  U   -      -      -      -     -     -     -     -     -     -
ALL   0      0      0      0  97.7   2.3   0.0   0.0   0.0   0.0

cpu    sysc   us    sy    wa    id    pc   %ec   ilcs   vlcs
  0    205   5.1  84.2   0.3  10.4  0.01   2.1     11    173
  1     47   6.3  72.7   0.2  20.8  0.01   1.4      4    157
  U      -     -     -   0.3  96.1  0.48  96.4      -      -
ALL    252   0.2   2.8   0.4  96.6  0.02   3.6     15    330
```

Table 5-2 mpstat output interpretation

Column	Measured parameter	Comments
cpu	The logical CPU ID.	The U shows the unused CPUs.
min/maj	Minor and major page faults.	A minor page fault causes no disk I/O, a major page fault does.
mpcr mpcs	Number of mpc interrupts sent (mpcs) and received (mpcr).	mpc interrupts is used for inter-processor communication.
dev	Number of device initiated interrupts.	Hardware interrupt.
soft	Number of software initiated interrupts.	
dec	Number of decrementer interrupts.	Timer interrupt.

Column	Measured parameter	Comments
ph	Number of phantom interrupts.	Number of interrupts received that were targeted to another partition that is sharing the same processor.
cs	Context switches.	
ics	Involuntary context switches.	
bound	The number of threads bound to the processor.	Through resource sets or the bindprocessor call.
rq	The length of the run queue.	The number of threads waiting to be run.
push	The number of threads migrated to a different processor due to load balancing.	
mig	Total number of thread migrations (to another logical processor).	Only shown on the default (no flags) version of the command.
s3pull	Number of migrations of the logical processor to a different physical processor on a different MCM.	Measures the migration of threads across MCM boundaries due to idle stealing (one MCM with no work).
s3grd	Number of dispatches from the global runqueue.	
s0rd	Percentage of thread re-dispatch that occurs on the same logical processor.	This is the optimum case; the thread uses the same registers. This should have a high value.
s1rd	Percentage of thread re-dispatch that occurs on the same physical processor but on a different logical processor.	This is a very close second-best; the thread uses the same L1 instruction and data caches. If the s0rd has a low value, then this metric should be high.
s2rd	Percentage of thread re-dispatch that occurs on the same chip, but not on the same core (POWER5 processor).	The L1 caches are different, but for POWER5, the L2 caches are shared between cores so the L2 and L3 caches may still be *hot* when the thread is re-dispatched.
s3rd	Percentage of thread re-dispatch that occurs on the same MCM but not on the same chip.	The thread is kept close to the physical memory it is using.

Column	Measured parameter	Comments
s4rd	Percentage of thread re-dispatch that occurs on the same book (collection of MCMs) but not on the same MCM.	Here the thread is starting to get a long way from home. All caches are cold and data paths to previously allocated data are getting longer. This metric should have a low value.
s5rd	Percentage of thread re-dispatch that occurs in a different book (collection of MCMs).	This is the worst-case scenario. This metric should have a low value. In performance terms, this is only marginally worse than the s4rd state.
sysc	Number of system calls.	
us, sy, wa, id	Logical cpu utilization in user, system, wait io, and idle.	
pc	Physical processor consumed.	Only available with SMT and shared-processor partitions.
%ec	Percent of entitled capacity used.	Shared-processor partitions only.
icls	Number of involuntary logical processor context switches.	Occurs when the logical processor's time slice expires.
vcls	Number of voluntary logical processor switches.	Voluntary context switches are initiated with the h_cede and the h_confer family of hcalls.

5.5.6 New monitoring commands on the Virtual I/O Server

There are a few commands that are designed to monitor system performance and throughput on the Virtual I/O Server. These include the **sysstat**, **viostat** and **topas** commands discussed in 5.5.4, "The topas command" on page 328.

The sysstat command

The **sysstat** command tells you briefly about the overall performance of the system and displays user sessions. Example 5-41 shows the syntax.

Example 5-41 Usage message of the sysstat command

```
$ sysstat -h
Usage: sysstat [-long | -short] [user]

       Prints a summary of current system activity.

       -long   Prints the summary in long form. This is the default.

       -short  Prints the time of day, amount of time since last system startup,
               number of users logged on, and number of processes running.
```

Example 5-42 provides an example output of the **sysstat** command.

Example 5-42 Example output of the **sysstat** *command*

```
$ sysstat -long
  03:29PM   up   1:20,  2 users,  load average: 0.51, 0.56, 0.50
User     tty          login@       idle      JCPU      PCPU what
padmin   vty0        02:09PM         22         1         0 -rksh
padmin   pts/0       02:16PM          0         0         0 -rksh
```

The viostat command

The **viostat** command can be very helpful in tracing system activity with regard to questions related to I/O. It allows for relatively fine-grained measurements of different type of adapters and attached disks as well as the usage of paths to redundant attached disks, including virtual adapters and virtual disks as well as their backing devices.

The usage message of this command is shown below in Example 5-43

Example 5-43 Usage message of the viostat command

```
$ viostat -h
Usage: viostat [-sys] [-adapter] [-disk | -extdisk | -tty] [-path] [-time]
               [PhysicalVolume...] [Interval [Count]]
       viostat

       Reports Central Processing Unit statistics, asynchronous input/output,
       input/output statistics for entire system, adapters, tty devices,
       disks and CD-ROMs.

       -sys       A system-header row is displayed followed by a line
                  of statistics for the entire system.  The hostname of
```

```
                the system is printed in the system-header row.

        -adapter    An adapter-header row is displayed followed by a line
                    of statistics for the adapter.
                    This will be followed by a disk-header row and the statistics
                    of all the disks/CD-ROMs connected to the adapter.

        -disk       A disk-header row is displayed followed by a line of
                    statistics for each disk that is configured.
                    If the PhysicalVolume parameter is specified, then only that
                    PhysicalVolume statistics is displayed.
                    The -disk, -extdisk and -tty flags are exclusive.

        -extdisk    A disk-header row is displayed followed by detailed
                    statistics for each disk that is configured.
                    If the PhysicalVolume parameter is specified, then only that
                    PhysicalVolume statistics is displayed.
                    The -disk, -extdisk and -tty flags are exclusive.

        -path       Displays the throughput for all paths to that device
                    followed by the throughput for that device.

        -tty        Display the statistics for the tty and cpu usage.
                    The -disk, -extdisk and -tty flags are exclusive.

        -time       Prints the time-stamp next to each line of output of viostat.
                    The time-stamp displays in the HH:MM:SS format.
```

The **viostat** command resembles the **iostat** command available in AIX 5L and is of comparable use to spot hot I/O paths and drill down to the related components. Here is the output of a measurement while disk I/O on one client with a virtual disk occurred. Example 5-44 shows the command output

Example 5-44 The viostat command output

```
$ viostat -extdisk
System configuration: lcpu=4 drives=7 paths=0 vdisks=9

dac0           xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0    374.8K     3.5       17.3K     357.4K
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         1.2      5.2      0.5      5.7S          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         2.4     47.0      2.0     34.6S          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         5.0      0.0     15.4S     0.0        0.0          0
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.9      4.8K     0.6        3.3K       1.5K
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.3      6.2      0.5    269.4          0          0
```

```
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          0.3     31.3      2.0    34.6S 0 0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                        16.7      0.0      5.2S     0.0        0.0        301
hdisk1         xfer:  %tm_act      bps      tps      bread      bwrtn
                          0.0      0.0      0.0        0.0        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                          0.0      0.0      0.0      0.0        0.0          0
hdisk2         xfer:  %tm_act      bps      tps      bread      bwrtn
                          0.0      0.0      0.0        0.0        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                          0.0      0.0      0.0      0.0        0.0          0
hdisk3         xfer:  %tm_act      bps      tps      bread      bwrtn
                          0.4    621.4      0.1       74.5      546.9
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                          0.0     60.6      0.6      5.7S          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          0.1     79.5      2.1      5.2S          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         50.3      0.0     15.4S     0.0        0.0          0
hdisk4         xfer:  %tm_act      bps      tps      bread      bwrtn
                          3.6    369.3K     2.8       14.0K     355.4K
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                          0.9      4.1      0.5    234.5          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          1.9     47.5      2.0    419.1          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                          0.2      0.0      5.7S     0.0        0.0          0
hdisk5         xfer:  %tm_act      bps      tps      bread      bwrtn
                          0.0      0.0      0.0        0.0        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                          0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                          0.0      0.0      0.0      0.0        0.0          0
--------------------------------------------------------------------------------
```

The workload management commands

Starting with Virtual I/O Server Version 1.3, there are commands that allow for monitoring partitions and collecting data in passive mode. The activation of workload management also allows you to view workload data using the **topas** command with the W switch. Collected data is stored in the /home/ios/perf/wlm

directory on a daily basis. The files are kept for two days and are then removed automatically. With the **wkldout** command, it is possible to extract data from the collected set related to CPU usage.

To start data collection, the **wkldmgr -start** command has to be issued first, then the **wkldagent -start** command can be used. Example 5-45 shows how to use the **wkldout** command to extract data in ASCII format from that collected set.

Example 5-45 Extracting performance data with the wkldout command

```
$ ls /home/ios/perf/wlm
xmwlm.061014  xmwlm.061015  xmwlm.log1    xmwlm.log2
$ wkldout -filename /home/ios/perf/wlm/xmwlm.061015 | tee xmwlm_061015
.
.
.
Time="2006/10/15 10:20:49", PagSp/%totalused=0.69
Time="2006/10/15 10:20:49", Proc/swpque=0.03
Time="2006/10/15 10:20:49", Proc/runque=1.01
Time="2006/10/15 10:20:49", CPU/glwait=0.22
Time="2006/10/15 10:20:49", CPU/gluser=0.07
Time="2006/10/15 10:20:49", CPU/glkern=0.78
Time="2006/10/15 10:23:49", WLM/wlmstate=1.00
Time="2006/10/15 10:23:49", PagSp/%totalused=0.69
Time="2006/10/15 10:23:49", Proc/swpque=0.08
Time="2006/10/15 10:23:49", Proc/runque=1.02
Time="2006/10/15 10:23:49", CPU/glwait=0.25
Time="2006/10/15 10:23:49", CPU/gluser=0.07
Time="2006/10/15 10:23:49", CPU/glkern=0.78
Time="2006/10/15 10:26:49", WLM/wlmstate=1.00
$ ls
backup              nohup.out           smit.transaction
config              smit.log            viosfirewall.rules
ioscli.log          smit.script         xmwlm_061015
$
```

As shown, to save the data into a file, the **tee** command has to be used since redirection of output to a file is not allowed in the restricted shell environment of the Virtual I/O Server. The file can then be transferred onto another machine and analyzed or displayed by other utilities. As also can be seen from the output, the data recorded by the **wkldmgr** and **wkldagent** commands contains only measurements for CPU and paging space. To collect data containing measurements for disk and network usage, use **topas** recording or **nmon** recording.

5.5.7 Monitoring with PLM

The Partition Load Manager (PLM) can be used in either monitoring or managing mode. In both modes, PLM provides summary information of the partitions that it is managing. The PLM command for monitoring partition state is **xlpstat**; an example of its output is shown in Example 5-46. The syntax of the command is:

```
xlpstat  [ -r ] { -p | -f } filename [ interval ] [ count ]
```

The common form is shown in Example 5-46; the -p switch specifies that the list of managed partitions will be retrieved from the given policy file; you usually use the policy file used when starting the PLM server. Alternatively you can provide a list of managed partitions in a text file, one partition per line, and use the -f flag to specify this file. The **xlpstat** command will query the status of the listed partitions. The output of this command does not distinguish between those partitions actively managed by PLM and those that are not.

The -r switch prints the output in raw mode, which is easier to parse by scripting languages.

Example 5-46 xlpstat command

```
# xlpstat -p 2_groups
                   CPU                        MEM
        ------------------------    ------------------
  STAT  TYP    CUR    PCT   LOAD     CUR    PCT  PGSTL  HOST
group2:
  up    S     0.5    4.00   0.10     512  75.17      0  plmserver
  up    S     0.50  85.45   0.44     512  99.17    129  vio_client2
group1:
  up    D     1.00  95.09   0.19     512  99.23    129  app_server
  up    D     1.00   0.39   0.09     512  74.73      0  db_server
```

The display shows the state of each managed partition on a separate line; the partitions are grouped into PLM groups. In the above example, there are two groups.

The STAT column indicates whether the partition is up or down. In the above example, all partitions are up.

The TYP column shows whether the partition uses shared processor (S), dedicated processors (D), or if **xlpstat** cannot query the partition and the state is unknown (this column is shown as U; this is usually a sign of connection problems). In Example 5-46, the partitions in group 2 are shared and those in group 1 are dedicated.

The following six columns are split into two groups of three: one for CPU usage, and the other for memory usage. The CUR column gives the current entitlement for CPU and memory and the PCT column gives the percent utilization. The LOAD column indicates the CPU load as measured by PLM and the PGSTL column indicates the memory load measured with the page steal rate.

The HOST column gives the name of the managed partition.

5.5.8 Performance workbench

The performance workbench is a graphical interface to monitor the system activity and processes. It has two windows; the first shows the partition configuration and the CPU and memory consumptions, and the second lists the top processes that can be sorted on the different provided metrics.

The performance workbench is a plug-in for the Eclipse development environment. It is in the bos.perf.gtools.perfwb lpp. The Eclipse IDE is available for AIX 5L and is found in the eclipse2.rte lpp. The Eclipse graphical desktop requires that X11 and Motif be installed too.

Use the **perfwb** command to start the Performance Workbench. Figure 5-25 shows the Procmon window of the performance workbench.

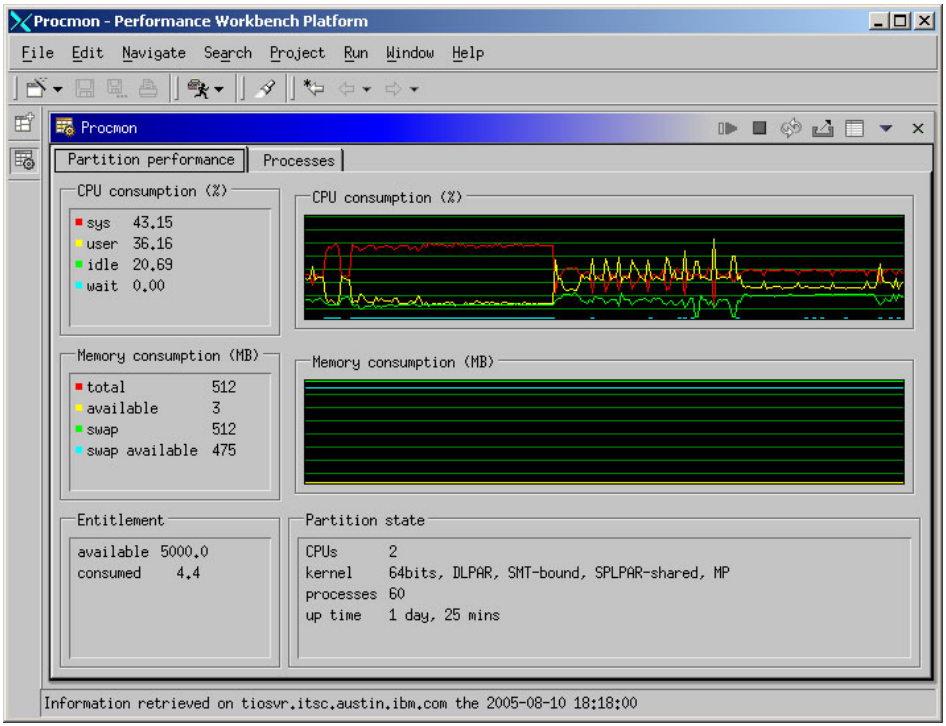


Figure 5-25 Performance workbench: Procmon window

5.5.9 The nmon command

The **nmon** command is a freeware monitoring tool for AIX 5L. The tool provides a text-based summary, similar to the **topas** command, of key system metrics. An additional tool, nmon analyzer, provides an easy way of transforming the text base output to a graphical format or for incorporation into spreadsheet software.

As of version 11, the **nmon** command is SMT and partition aware.

The **nmon** command is available from:

http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmon

Extract and install the nmon_aix53 file, typically under /usr/sbin/nmon. Optionally, change and verify the iostat flags to continuously maintain the disk I/O history using the following commands:

```
# chdev -l sys0 -a iostat=true
```

```
# lsattr -D -l sys0 -a iostat
```

You may get a warning if you do not position this flag to true, but **nmon** will continue to show non-null values for disk usage.

If you have a large number of disks (more than 200), then setting iostat to true will start consuming CPU time, around 2 percent. Once the measurement campaign is completed you should set the flag back to false.

Using the r and p switches after startup, as shown in Figure 5-26, presents the logical partition information.

```
-nmon--------p=Partitions-------Host=VIO_Server1----Refresh=2 secs---17:55.25-
Resources
-------------------------------------------------------------------
Machine has 4 PowerPC_POWER5 (64 bit) CPUs with 4 CPUs active
CPU Speed 1654.3 MHz
Logical partition=Dynamic
Virtual I/O Server Version=1.3.0.0 Kernel=64 bit Multi-Processor
Hardware-Type(NIM)=CHRP=Common H/W Reference Platform Bus-Type=PCI
CPU Architecture  =PowerPC Implementation=POWER5
CPU Level 1 Cache is Combined Instruction=65536 bytes & Data=32768
bytes
Level 2 Cache size=0
Shared-CPU-Logical-Partition---------------------------------------
Partition:Number=1 "VIO_Server1"
Flags:     LPARed DRable SMT-bound Shared UnCapped PoolAuth
Summary: Entitled= 0.50  Used     0.00 (  0.9%)  0.2% of CPUs in
System
         PoolCPUs= 2     Unused   1.99           0.2% of CPUs in
Pool CPU-Stats---------   Capacity----------------
ID-Memory---------------------- Physical(+CUoD)  2   Cap. Processor
Min  0.20  LPAR ID Group:Pool  32769:0 Active (in_sys)  2   Cap.
Processor Max  0.80  Memory(MB) Min:Max  128:1152 Virtual Online   2
Cap. Increment      0.01  Memory(MB) Online    512 Logical Online
4   Cap. Unallocated    0.00  Memory Region LMB   16MB min Physical
pool    2   Cap. Entitled       0.50  Time------------------Seconds
SMT threads/CPU  2    -MinReqVirtualCPU  0.10  Time Dispatch Wheel
0.0100 CPU--------Min-Max   Weight------------------  MaxDispatch
Latency    0.0150 Virtual      1   2   Weight Variable      255
Time Pool Idle         1.9879 Logical      1   4   Weight
Unallocated     0  Time Total Dispatch    0.0046
-------------------------------------------------------------------
--------
```

Figure 5-26 nmon Resources and LPAR screen

nmon uses the NMON environment variable for the default display format. For example, you can use:

```
# export NMON=cmt
# nmon
```

This will show CPU and memory usage screens and the top processes screens one after the other.

The nmon tool on the VIOS

This latest version of the **nmon** command can monitor resource usage on the Virtual I/O Server.

To install **nmon** on the VIOS, use the padmin account to FTP the **nmon** executable file extracted from the downloaded package onto the VIOS partition, renaming it nmon:

```
# ftp vio_server
  (user padmin)
> put nmon_aix53 nmon
```

Log on to the VIOS using the **telnet** command or using the HMC and log in as user padmin. Since the installation of software is not allowed for the padmin user, you have to switch to a root shell using the **oem_setup_env** command. First, change the execution bits on the file:

```
# chmod 550 nmon
```

Then, move the command to a directory in the PATH variable of the padmin user, usually /usr/ios/oem:

```
# mv nmon /usr/ios/oem
```

Leave the root shell with **exit**. Change the system attributes to maintain the disk I/O history; the command is not the same as for AIX 5L:

```
# chdev -dev sys0 -attr iostat=true
# lsdev -dev sys0 -attr iostat
```

You can now use **nmon** in interactive mode.

```
# NMON=cmt nmon
```

You can use the standard **nmon** switches.

Recording with the nmon tool

You can record resource usage using **nmon** for subsequent analysis with the **nmon** analyzer tool. This will work on both standard AIX 5L partitions and the VIOS:

```
# nmon -f -t -s <interval> -c <count>
```

The **nmon** process runs in the background and you can log off the partition if you wish. For best results, the count should not be greater than 1,500. The command will create a file with a name in the following format:

```
<nom-part>_<date>_<hour>
```

Once the recording process has finished, you must run the file through nmon2csv, which is contained in the nmon_analyzer package and then transfer the file to a machine that runs Microsoft® Excel® spreadsheet software to run the **nmon** analyzer tool.

When monitoring system behavior in a virtualized environment, you should monitor the VIOS and the client partitions at the same time. Set the “Allow shared processor pool utilization authority” attribute to true for at least one of the monitored partitions on the HMC.

5.5.10 AIX Performance Toolbox

The AIX Performance Toolbox (PTX) supports shared and dedicated processor partitions. It includes monitors for virtualized environments, such as entitlement and consumed entitlement.

5.5.11 Dynamic Reconfiguration Awareness

The **vmstat**, **iostat**, **sar**, **mpstat**, and **lparstat** commands are all *dynamic reconfiguration* aware. This means that they are able to detect when the system has been changed with a dynamic reconfiguration (dynamic LPAR operation). These commands start their output with a pre-header that has a summary of the configuration, and each time the configuration changes, it prints a warning and the pre-header information with the new configuration and continues.

Example 5-47 on page 355 shows how this works for the **vmstat** command and the addition of a virtual processor. Because SMT is enabled, the addition of one virtual processor results in two additional logical processors, so there are two dynamic reconfiguration events. The lines in bold font show the **vmstat** output caused by the configuration change.

Example 5-47 Dynamic reconfiguration and the vmstat command

```
# vmstat 2 6

System configuration: lcpu=2 mem=512MB ent=0.50

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
 0  0 92678 16964   0   2   0   0    0   0   2   25 182  0  0 97  2  0.00   1.0
 1  0 92678 16964   0   0   0   0    0   0   1    5 147  0  0 99  0  0.00   0.8
System configuration changed. The current iteration values may be inaccurate.
 1  0 93091 15932   0 207   0   0    0   0   3 68741 1147 3 15 59 23 0.14  28.7

System configuration: lcpu=3 mem=512MB ent=0.50

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
System configuration changed. The current iteration values may be inaccurate.
 0  0 93201 15678   0  31   0   0    0   0   2  656 328  1 17 77  5  0.13  25.2

System configuration: lcpu=4 mem=512MB ent=0.50

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
 0  0 93329 15550   0   0   0   0    0   0   2   10 152  0  1 99  0  0.01   1.2
 0  0 93329 15550   0   0   0   0    0   0   1    5 150  0  0 99  0  0.00   1.0
```

5.6 Sizing considerations

The virtualization technologies of the servers discussed in this redbook add flexibility to the computing infrastructure. But the intrinsic computing power of the platform does not change because of virtualization. However, application performance and business responsiveness are improved by virtualization because virtualization allows you to assign resources to applications in line with the business objectives. The resource and workload management tools constantly monitor the system load and rapidly readjust resource assignments in response to any change. Herein lies the real power of IBMs virtualization technologies.

This section gives some guidelines on configuring partitions on IBM System p5 servers.

The POWER Hypervisor is part of all IBM System p5 servers; you cannot configure a server without the POWER Hypervisor. All the performance benchmarks published by IBM on System p5 servers are done with the POWER Hypervisor. When you select an server with a given rPerf, this is the performance potential that is available for your applications.

5.6.1 Partition configuration considerations

This section describes some of the points to consider when configuring partitions on an IBM System p5 server.

Partition count

As a general rule, the number of partitions should be kept as low as possible. It is preferable to consolidate several applications on to a single AIX 5L partition rather than creating one partition for each application. This is sometimes not feasible for technical or organizational reasons, for example, tuning AIX 5L for transactional database applications may degrade performance in other types of applications.

Each partition must be managed just like any stand-alone server, requiring configuration, backups, and software licenses. Keeping the partition count down has a direct impact on the administration costs and thus on the total cost of ownership of any server.

Further, each partition has associated resources. These resources have a context information that must be managed by the virtualization software. This management of state information consumes resources that might otherwise be deployed in the partitions.

Resource maximums

One parameter of partition definition is the maximum number of any given resource, be it memory, processors, or virtual I/O slots. It may be tempting to position the maximum values of these figures to be sure that it will always be possible to increase a partition's resources using dynamic reconfiguration. However, the POWER Hypervisor, just like an operating system, must maintain data structures that will allow it to manage the eventuality that each partition receives the maximum possible resources. The Hypervisor will reserve the necessary memory making it unavailable to the partitions. You should therefore specify only *realistic* values for the maximum number of resources, using a planned margin.

Virtual CPU count

Related to the resource maximums is the virtual CPU count for any partition and the sum of all virtual processors in all the shared-processor partitions.

The following rules of thumb should be considered when configuring the number of virtual processors in shared-processor partitions:

- The number of virtual processors in a shared-processor partition should not exceed the number of physical processors in the shared-processor pool.
- The number of virtual processors in a capped shared-processor partition should not exceed the entitlement rounded up to the nearest whole number.
- For versions of AIX 5L prior to V5.3 Maintenance Level 3 or with later versions of AIX 5L with the virtual processor folding feature disabled, the sum of all the virtual CPUs in all active shared-processor partitions should not be greater than four times the number of physical processors in the shared-processor pool.
- For versions of AIX 5L after Version 5.3 ML3 with virtual processor folding, an excessive virtual CPU count has a very low performance impact.

Capped or uncapped partitions?

Capped partitions will never exceed their entitlements, even if they are overloaded and there are unused resources in the system. Generally, the use of capped partitions is to be discouraged; use uncapped partitions and prioritize the allocation of spare capacity using partition weights.

5.6.2 Virtualization and applications

Some applications and middleware products are not able to adapt to dynamic reconfiguration changes, for example, the product starts a number of processes based on the number of configured processors. If the application requires more computing power, adding additional processors will not have any effect without shutting down the application and restarting it. Using shared processors, it is possible to change the entitlement of virtual processors to change the processing resources available to the application. Because the processor count does not change, then applications that are not DR aware do not have to be restarted.

If an application or workload environment is cache sensitive or cannot tolerate the variability introduced with resource sharing, it should be deployed in a dedicated processor partition with SMT disabled. In dedicated partitions, the entire physical processor is assigned solely to the partition.

5.6.3 Resource management

PLM and WLM provide resource and workload management. Some applications and middleware also provide their own resource management and in particular databases. Care should be taken when using the resource management tools in AIX 5L and the POWER Hypervisor with such applications and middleware.

5.7 Security considerations for Virtual I/O Servers

The Virtual I/O Server is crucial to the function of the system. Security is always a concern. Although the Virtual I/O Server does not run a lot of services, it has open ports in the network for connectivity and users can be created that may have rights to alter specific system parameters. We will discuss the following topics here:

- Network security
- System parameter security
- Viewing protocol entries related to security

5.7.1 Network security

After installation of the Virtual I/O Server, there is no IP address assigned to one of the Ethernet interfaces until you configure it. If that has been done, by default there are some services active on the system and available from the network that should be carefully checked. Here is the output of a port scan that took place after an IP address had been assigned to one of the network interfaces of the Virtual I/O Server:

`ftp`	Ports 20 for data and 21 for control connection allows unencrypted connections to the system; use the **`scp`** command instead.
`ssh`	Port 22 is always encrypted.
`telnet`	Port 23 allows unencrypted connections to the system; use the **`ssh`** command instead.
`rcpbind`	Port 111 is used for NFS connections.
RMC connections	Port 657 runs encrypted.

Stopping telnet and ftp services

The **`ftp`** and **`telnet`** commands can be closed if there are means to connect to and exchange data with the Virtual I/O Server via an encrypted protocol. Since SSH comes preinstalled with the system, we recommend stopping those services and use SSH instead. To stop the services in the running system as well as prevent them from starting after a reboot, use the **`stopnetsvc`** command. Example 5-48 on page 359 shows how to use it.

Example 5-48 Stopping ftp and telnet services on the Virtual I/O Server

```
$ stopnetsvc telnet
0513-127 The telnet subserver was stopped successfully.
$ stopnetsvc ftp
0513-127 The ftp subserver was stopped successfully.
$
```

To enable data transfer to the Virtual I/O Server using **scp**, see 6.2.2, “Install and configure SSL and SSH” on page 382.

Using viosecure for finer grained control of network security

The **viosecure** command, available with Virtual I/O Server Version 1.3, allows for fine grained control of network settings in terms of allowed and denied ports as well as for specifying which remote machine can connect to which port. The usage message shows which flags are supported (Example 5-49).

Example 5-49 Usage message of the viosecure command

```
$ viosecure
Too few parameters.

Usage: viosecure  -level LEVEL [-apply] | [-nonint] -view
       viosecure  -firewall on [[-force] -reload] | off
        viosecure  -firewall allow | deny -port number [-interface
ifname]
                  [-address IPaddress] [-timeout Timeout] [-remote]
       viosecure  -firewall view [-fmt delimiter]


The viosecure command failed.
$
```

The firewall can be switched on and off. If enabled with the -on switch, connections to the ports ftp-data (20), ftp (21), ssh (22), www (80), https (443), rmc (657), and cimon (32768) will be allowed, all other traffic will be denied. Example 5-50 shows this procedure.

Example 5-50 Activating the firewall with the viosecure command

```
$ viosecure -firewall on
$ viosecure -firewall view
Firewall     ON

                      ALLOWED   PORTS
          Local   Remote
Interface  Port    Port    Service      IPAddress       Expiration
Time(seconds)
---------  ----   ----   -------     ---------     ---------------
$
```

Further configuration can allow access to specific ports generally or just for specified remote machine(s). We show as an example here how to restrict the **rsh** command connections (Port 112) to a single machine with an IP address of 9.3.5.111. This could well be an administrative machine from which connections to the HMC can be allowed, as the HMC is also able to restrict connections to specified machines (Example 5-51).

Example 5-51 Allowing rsh connections from a specified IP address

```
$ viosecure -firewall allow -port exec -address 9.3.5.111
$ viosecure -firewall view
Firewall     ON

                      ALLOWED   PORTS
          Local   Remote
Interface  Port    Port    Service      IPAddress       Expiration
Time(seconds)
---------  ----   ----   -------     ---------     ---------------
all        512    any    exec        9.3.5.111      0
$
```

The login will now succeed from the IP address 9.3.5.111, while other machines will not be able to connect. To deny connections to a specified port as well as to remove the opened port from the firewall list, use the following command (Example 5-52 on page 361).

Example 5-52 Denying rsh connections from a specified IP address

```
$ viosecure -firewall deny -port exec -address 9.3.5.111
$ viosecure -firewall view
Firewall     ON

                      ALLOWED   PORTS
          Local   Remote
Interface  Port    Port    Service      IPAddress       Expiration
Time(seconds)
---------  ----   ----   -------    ---------     ---------------
$
```

All firewall settings will be recorded to the file viosfirewall.rules in the home directory of the padmin user upon firewall shutdown with the **viosecure -firewall off** command. However, the padmin user is not allowed to see the contents. Upon startup of the firewall, the settings of the file viosfirewall.rules will be applied to the firewall again so that no modifications are lost.

System parameter security

Some system parameters require careful consideration if higher security levels are required. These can be user parameters as well as network parameters. The **viosecure** command as of Virtual I/O Server Version 1.3 supports changes to 37 parameters. The output of the **viosecure -nonint -view** command shows the parameters that can be set:

```
$viosecure -nonint -view
Enable telnet (telnetdls):Uncomments the entry for telnetd daemon in
/etc/inetd.conf and starts telnetd daemon.
Disable UDP chargen service in /etc/inetd.conf (udpchargendls):comments
the entry for UDP Chargen service in /etc/inetd.conf and kills all
instances of chargen.
Disable sprayd in /etc/inetd.conf (sprayddls):comments the entry for
sprayd daemon in /etc/inetd.conf and kills all instances of sprayd.
Minimum number of chars (mindiffdls):Removes the constraint on the
minimum number of characters required in a new password that were not
in the old password.
Set core file size (coredls):Remove the core attribute for root.
Password expiration warning time (pwdwarntimedls):Removes the
constraint on the number of days before the system issues a warning
that a password change is required.
Disable dtspc in /etc/inetd.conf (dtspcdls):comments the entry for
dtspc daemon in /etc/inetd.conf when LFT is not configured and CDE is
disabled in /etc/inittab, also kills all the instances of dtspc daemon.
```

```
Enable mail client (dismaildmndls):Uncomments the entry for Sendmail
daemon in /etc/rc.tcpip.
Disable rstatd in /etc/inetd.conf (rstatddls):comments the entry for
rstatd daemon in /etc/inetd.conf and kills all instances of rstatd.
Object creation permissions (umaskdls):Specifies default object
creation permissions to 022.
Disable sysstat in /etc/inetd.conf (systatdls):comments the entry for
systat daemon in /etc/inetd.conf and kills all instances of systat.
Disable NTP daemon (disntpdmndls):Stops NTP daemon and comments it's
entry in /etc/rc.tcpip.
Disable DPID2 daemon (disdpid2dmndls):Stops DPID2 daemon and comments
it's entry in /etc/rc.tcpip.
Network option nonlocsrcroute (nonlocsrcroutedls):Set network option
nonlocsrcroute to default value.
Enable UDP time service in /etc/inetd.conf (udptimedls):Uncomments the
entry for UDP Time service in /etc/inetd.conf and kills all instances
of time service(udp).
Enable ttdbserver service in /etc/inetd.conf (ttdbserverdls):Uncomments
the entry for ttdbserver service in /etc/inetd.conf and kills all
instances of ttdbserver service.
Network option ipsrcroutesend (ipsrcroutesenddls):Set network option
ipsrcroutesend to default value.
Network option sb_max (sb_maxdls):Set network option sb_max's value to
default value.
Disable TCP echo service in /etc/inetd.conf (tcpechodls):comments the
entry for TCP Echo service in /etc/inetd.confand kills all instances of
echo(tcp).
Stop DHCP Client (disdhcpclientdls):Stops DHCP Client and comments it's
entry in /etc/rc.tcpip.
Disable TCP Discard service in /etc/inetd.conf (tcpdiscarddls):comments
the entry for TCP Discard service in /etc/inetd.conf and kills all
instances of discard(tcp).
Local login (rootlogindls):Enables root to login locally.
Disable gated daemon (disgateddmndls):Stops gated daemons and comments
the entry for gated daemon in /etc/rc.tcpip.
Enable TCP daytime service in /etc/inetd.conf
(tcpdaytimedls):Uncomments the entry for TCP Daytime service in
/etc/inetd.conf.
Network option udp_pmtu_discover (udp_pmtu_discoverdls):Set network
option udp_pmtu_discover to default value.
Disable krlogind in /etc/inetd.conf (krlogindls):comments the entry for
krlogind daemon in /etc/inetd.conf and kills all instances of krlogind.
Enable TCP time service in /etc/inetd.conf (tcptimedls):Uncomments the
entry for TCP Time service in /etc/inetd.confand kills all instances of
timed(tcp).
```

```
Network option icmpaddressmask (icmpaddressmaskdls):Set network option
icmpaddressmask to default value.
Delay between unsuccessful logins (logindelaydls):Removes any login
delays between two unsuccessful login attempts.
Set SUID of remote Commands (rmsuidfrmrcmdsdls):Sets SUID of remote
commands rcp, rdist, rexec, remsh, rlogin and rsh.
Disable rquotad in /etc/inetd.conf (rquotaddls):comments the entry for
rquotad daemon in /etc/inetd.conf and kills all instances of rquotad.
Enable rlogin in /etc/inetd.conf (rlogindls):Uncomments the entry for
rlogind daemon in /etc/inetd.conf.
Network option tcp_recvspace (tcp_recvspacedls):Set network option
tcp_recvspace's value to 262144.
Stop autoconf6 (disautoconf6dls):Stops autoconf6, if it is running and
comments the entry for autoconf6 in /etc/rc.tcpip.
Minimum number of non-alphabetic chars (minotherdls):Removes the
minimum number of non-alphabetic characters constraint, in a password.
Disable comsat in /etc/inetd.conf (comsatdls):comments the entry for
comsat daemon in /etc/inetd.conf and kills all instances of comsat.
Login timeout (logintimeoutdls):Specifies the time interval(60 seconds)
to type in a password.
Disable IMAPD (imapddls):comments the entry for imapd daemon in
/etc/inetd.conf and kills all instances of imapd.
Enable NFS daemon (disablenfsdls):Enables NFS mounts, starts NFS
daemons and enables NFS from startup.
Time to change password after the expiration (maxexpireddls):Removes
any minimum number of weeks requirements, after maxage that an expired
password can be changed by the user.
Reenable login after locking (loginreenabledls):Removes any time
interval after which a port is unlocked after being disabled by
logindisable.
Disable fingerd in /etc/inetd.conf (fingerddls):comments the entry for
fingerd daemon in /etc/inetd.conf and kills all instances of fingerd.
Remote root login (rootrlogindls):Enables remote root login.
Stop DHCP Server (disdhcpservdls):Stops DHCP Server daemon and comments
it's entry in /etc/rc.tcpip.
Enable uucpd in /etc/inetd.conf (uucpdls):Uncomments the entry for
uucpd daemon in /etc/inetd.conf.
Enable talk in /etc/inetd.conf (talkdls):Uncomments the entry for talk
daemon in /etc/inetd.conf.
Network option tcp_pmtu_discover (tcp_pmtu_discoverdls):Set network
option tcp_pmtu_discover to default value.
Stop DHCP Agent (disdhcpagentdls):Stops DHCP relay agent and comments
it's entry in /etc/rc.tcpip.
Network option directed_broadcast (directed_broadcastdls):Set network
option directed_broadcast to default value.
```

```
Network option extendednetstats (extendednetstatsdls):Set network
option extendednetstat's value to default value.
Disable TCP chargen service in /etc/inetd.conf (tcpchargendls):comments
the entry for TCP Chargen service in /etc/inetd.confand kills all
instances of chargen(tcp).
Enable rshd daemon (shelldls):Uncomments the entry for rshd daemon in
/etc/inetd.conf.
Disable rexd in /etc/inetd.conf (rexddls):comments the entry for rexd
daemon in /etc/inetd.conf and kills all instances of rexecd.
Network option ip6srcrouteforward (ip6srcrouteforwarddls):Set network
option ip6srcrouteforward to default value.
Enable qdaemon (disqdaemondls):Starts qdaemon and uncomments the
qdaemon entry in /etc/inittab.
Password reuse time (histsizedls):Removes the constraint on the number
of previous passwords a user cannot reuse.
Enable rexecd in /etc/inetd.conf (rexecddls):Uncomments the entry for
rexecd daemon in /etc/inetd.conf.
Network option ipsendredirects (ipsendredirectsdls):Set network option
ipsendredirects to default value.
Network option tcp_mssdflt (tcp_mssdfltdls):Set network option
tcp_massdflt's value to default value.
Network option ipforwarding (ipforwardingdls):Set network option
ipforwarding to default value.
Remove root user in /etc/ftpusers file (chetcftpusersdls):Removes
/etc/ftpusers file.
Remove the unsuccessful login constraint (logindisabledls):Removes the
constraint on the number of unsuccessful login attempts on a port,
before the port can be locked.
Network option ipsrcrouterecv (ipsrcrouterecvdls):Set network option
ipsrcrouterecv to default value.
Disable binaudit (binauditdls):Disables bin auditing.
Set login herald (loginheralddls):Remove login herald from default
stanza.
Number of login attempts before locking the account (loginretri:Removes
the constraint on the number of consecutive unsuccessful login attempts
per non-root user account before the account is disabled.
Enable piobe daemon (dispiobedls):Starts piobe daemon and uncomments
the piobe entry in /etc/inittab.
Enable lpd daemon (dislpddls):Stops lpd daemon and comments the lpd
entry in /etc/inittab.
Network option clean_partial_conns (clean_partial_connsdls):Set network
option clean_partial_conns to default value.
Network option ipignoreredirects (ipignoreredirectsdls):Set network
option ipignoreredirects to default value.
```

```
Network option tcp_sendspace (tcp_sendspacedls):Set network option
tcp_sendspace's value to default value.
Disable SNMP daemon (dissnmpdmndls):Uncomments the entry for SNMP
daemon in /etc/rc.tcpip and starts the daemon as well.
Disable print daemon (disprintdmndls):Stops the print daemon and
comments it's entry in /etc/rc.tcpip.
Enable UDP daytime service in /etc/inetd.conf
(udpdaytimedls):Uncomments the entry for UDP Daytime service in
/etc/inetd.conf and kills all instances of daytime.
Remove dot from non-root path (rmdotfrmpathnrootdls):Removes dot from
PATH environment variable from files .profile, .kshrc, .cshrc and
.login in user's home directory.
Disable netstat in /etc/inetd.conf (netstatdls):comments the entry for
netstat daemon in /etc/inetd.conf and kills all instances of netstat.
Interval between unsuccessful logins (loginintervaldls):Removes any
time interval for a port in which the unsuccessful login attempts must
occur before the port is disabled.
Remove entries from /etc/hosts.equiv file (rmetchostsequivdls):Removes
entries from /etc/hosts.equiv file.
Remove dot from /etc/environment (rmdotfrmpathetcenvdls):Removes dot
from PATH environment variable from /etc/environment.
Maximum times a char can appear in a password (maxrepeatsdls):Specifies
the maximum number of times a character can appear in a password to 8.
Disable UDP discard service in /etc/inetd.conf (udpdiscarddls):comments
the entry for UDP Discard service in /etc/inetd.conf and kills all
instances of UDP discard.
Remove rhosts and netrc services (rmrhostsnetrcdls):Removes /.rhosts
and /.netrc files.
Disable rwhod daemon (disrwhoddmndls):Stops rwhod daemon and comments
it's entry in /etc/rc.tcpip.
Disable rwalld in /etc/inetd.conf (rwallddls):comments the entry for
rwalld daemon in /etc/inetd.conf and kills all instances of rwalld.
Network option ipsrcrouteforward (ipsrcrouteforwarddls):Set network
option ipsrcrouteforward to default value.
Network option rfc1323 (rfc1323dls):Set network option rfc1323's value
to default value.
Disable print daemon (disdnsdmndls):Stops DNS daemon and comments it's
entry in /etc/rc.tcpip.
Limit system access (limitsysaccdls):Removes the file cron.allow and
removes all entries in cron.deny file.
Disable bootpd in /etc/inetd.conf (bootpsdls):comments the entry for
bootpd daemon in /etc/inetd.conf and kills all instances of bootpsd.
Disable pcnfsd in /etc/inetd.conf (pcnfsddls):comments the entry for
pcnfsd daemon in /etc/inetd.conf and kills all instances of pcnfsd.
```

```
Enable unsecure daemons (disrmtdmnsdls):Enables unsecure daemons rshd,
rlogind and tftpd
Disable POP3D (pop3ddls):comments the entry for pop3d daemon in
/etc/inetd.conf and kills all instances of pop3d.
Network option bcastping (bcastpingdls):Set network option bcastping to
default value.
Disable ruserd in /etc/inetd.conf (rusersddls):comments the entry for
rusersd daemon in /etc/inetd.conf and kills all instances of rusersd.
Enable FTP (ftpdls):Uncomments the entry for ftpd daemon in
/etc/inetd.conf and starts ftpd daemon.
Password reset time (histexpiredls):Removes any minimum number of weeks
requirements before a password can be reused.
Enable CDE (discdedls):Enables CDE.
Disable timed daemon (distimedmndls):Stops timed daemon and comments
it's entry in /etc/rc.tcpip.
Enable cmsd service in /etc/inetd.conf (cmsddls):Uncomments the entry
for cmsd service in /etc/inetd.conf and starts cmsd service.
Remove dot from path root (rmdotfrmpathrootdls):Remove dot from PATH
environment variable from files .profile, .kshrc, .cshrc and .login in
root's home directory.
Disable mrouted daemon (dismrouteddmndls):Stops mrouted daemon and
comments it's entry in /etc/rc.tcpip.
Disable krshd daemon (kshelldls):comments the entry for krshd daemon in
/etc/inetd.conf and kills all instances of krshd.
Disable tftp in /etc/inetd.conf (tftpdls):comments the entry for tftp
daemon in /etc/inetd.conf and kills all instances of tftpd.
Minimum number of alphabetic chars (minalphadls):Removes the minimum
number of alphabetic characters constraint, in a password.
Minimum length for password (minlendls):Removes the minimum length
constraint on password.
Maximum age for password (maxagedls):Removes any minimum number of
weeks requirement for a password validity.
Enable unsecure commands (disrmtcmdsdls):Enables unsecure commands
rlogin, rsh, rcp and tftp.
Minimum age for password (minagedls):Removes any minimum number of
weeks requirements for a password change.
Disable routing daemon (disrtngdmndls):Stops routed daemon and comments
it's entry in /etc/rc.tcpip.
Disable UDP Echo service in /etc/inetd.conf (udpechodls):comments the
entry for UDP Echo service in /etc/inetd.conf and kills all instances
of UDP echo.
$
```

The level of the settings is appended to the name of the setting that is mentioned in round brackets. For example, the security level of the last line of output, the UDP Echo service, is `udpechodls` which is UDP Echo default level security. The description what will be applied in the security level is found after the colon. There are four levels of security with **`viosecure`**:

- Default level security (-dls)
- Low level security (-lls)
- Medium level security (-mls)
- High level security (-hls)

The appropriate level of security can be applied by the following command and then be viewed (Example 5-53).

Example 5-53 Applying high level security with the viosecure command

```
$viosecure -level high -apply
viosecuree -nonint -view
.
.
.
Minimum age for password (minagehls):Specifies the minimum number of
weeks to 1
week, before a password can be changed.
Disable routing daemon (disrtngdmnhls):Stops routed daemon and comments
it's entry in /etc/rc.tcpip.
Disable unsecure commands (disrmtcmdshls):Disables unsecure commands
rlogin, rsh, rcp and tftp.
Check group definitions (grpckhls):Verifies the correctness of group
definitions and fixes the errors.
Disable UDP Echo service in /etc/inetd.conf (udpechohls):comments the
entry for
UDP Echo service in /etc/inetd.conf and kills all instances of UDP
echo.
Disable mail client (dismaildmnhls):Stops Sendmail daemon and comments
it's entry in /etc/rc.tcpip.
Disable telnet (telnethls):comments the entry for telnetd daemon in
/etc/inetd.conf and kills all instances of telnetd.
Disable unsecure daemons (disrmtdmnshls):Disables unsecure daemons
rlogind, rshd, and tftpd.
Disable UDP chargen service in /etc/inetd.conf (udpchargenhls):comments
the entry for UDP Chargen service in /etc/inetd.conf and kills all
instances of chargen.
Disable sprayd in /etc/inetd.conf (spraydhls):comments the entry for
sprayd daemon in /etc/inetd.conf and kills all instances of sprayd.
```

```
Minimum number of chars (mindiffhls):Specifies the minimum number of
characters
required in a new password to 4, that were not in the old password.
```

Viewing protocol entries related to security

There are two commands that can be used to view the relevant security information besides the entries made to the error log (which can be viewed by the **errorlog** command): The commands are **lsgcl**, for listing the global command log, and **lsfailedlogin**, for listing failed login attempts that may give a clue to whether the security of the Virtual I/O Server might be at risk.

The **lsgcl** command takes no options and lists the commands executed in historical order. Example 5-54 provides sample output of this command.

Example 5-54 Sample output of the lsgcl command

```
$lsgcl
Oct 12 2006, 16:03:30 padmin   stopnetsvc  telnet
Oct 12 2006, 16:04:07 padmin   starttrace
Oct 12 2006, 16:04:54 padmin   stoptrace
Oct 12 2006, 16:05:08 padmin   cattracerpt
Oct 12 2006, 16:07:59 padmin   lsnetsvc  telnet
Oct 12 2006, 16:08:07 padmin   lsnetsvc  ftp
Oct 12 2006, 16:08:16 padmin   lsnetsvc  ssh
Oct 19 2006, 18:36:35 padmin   stopnetsvc  telnet
Oct 19 2006, 18:36:42 padmin   stopnetsvc  ftp
```

The **lsfailedlogin** command takes no arguments, too, and lists unsuccessful connection attempts. The output of the command is provided in Example 5-55.

Example 5-55 Sample output of the lsfailedogin command

```
$lsfailedlogin
.
.
.
UNKNOWN_                ssh           7 241672 0000 0000 1156518419
prov009.itsc
.austin.ibm.com      Fri Aug 25 10:06:59 CDT 2006
root                    ssh           7 344080 0000 0000 1157127973
cairo.itsc.a
ustin.ibm.com        Fri Sep  1 11:26:13 CDT 2006
padmin                  vty0          7 258210 0000 0000 1160430832
                     Mon Oct  9 16:53:52 CDT 2006
padmin                  vty0          7 258214 0000 0000 1160683787
                     Thu Oct 12 15:09:47 CDT 2006
```

```
padmin                 vty0          7 258214 0000 0000 1160683796
                     Thu Oct 12 15:09:56 CDT 2006
padmin                 vty0          7 258214 0000 0000 1160683821
                     Thu Oct 12 15:10:21 CDT 2006
padmin                 vty0          7 258216 0000 0000 1160683832
                     Thu Oct 12 15:10:32 CDT 2006
padmin                 vty0          7 204974 0000 0000 1161299525
                     Thu Oct 19 18:12:05 CDT 2006
padmin                 pts/1          7 270494 0000 0000 1161302664
lpar01.itsc.
austin.ibm.com       Thu Oct 19 19:04:24 CDT 2006
UNKNOWN_               pts/1          7 270494 0000 0000 1161302670
lpar01.itsc.
austin.ibm.com       Thu Oct 19 19:04:30 CDT 2006
$
```

6

Partition Load Manager

This chapter describes the Partition Load Manager (PLM). It shows you how to install and configure it for managing both CPUs and memory.

This chapter is structured as follows:

- Partition Load Manager introduction
- Resource Monitoring and Control (RMC)
- Resource management
- Installing and configuring Partition Load Manager
- Point-in-time and recurring reconfiguration
- Tips and troubleshooting PLM
- PLM considerations

6.1 Partition Load Manager introduction

Partition Load Manager (PLM) for AIX 5L is designed to automate the administration of memory and CPU resources across logical partitions within a single central electronics complex (CEC). To improve system resource usage, PLM automates the migration of resources between partitions based on partition load and priorities; partitions with a high demand will receive resources donated by or taken from partitions with a lower demand. A user-defined policy governs how resources are moved. PLM will not contradict the partition definitions in the HMC. PLM allows administrators to monitor resource usage across all the managed partitions.

PLM is part of the Advanced POWER Virtualization feature. It is supported on both dedicated and shared-processor partitions running AIX 5L V5.2 (ML4) or AIX 5L V5.3.0 or later on IBM System p5 servers.

6.1.1 PLM operating modes

PLM can be started in one of two modes:

- Monitoring mode
- Management mode

In monitoring mode, PLM reports provide a number of statistics on resource usage in the managed partitions. This is discussed in 5.5.7, “Monitoring with PLM” on page 348.

In management mode, PLM will initiate dynamic reconfiguration operations in order to match system resources with partition workload in accordance with the defined policy.

6.1.2 Management model

PLM uses a client/server model, shown in Figure 6-1 on page 373, to monitor and manage partition resources. The clients act as agents on each of the managed partitions. The PLM server configures each of the agents (clients), setting the thresholds at which the server should be notified. The agents monitor the partition’s resource usage and notify the PLM server whenever PLM-set thresholds are passed (under or over-utilized). Based on a user-defined resource management policy, the PLM server invokes dynamic reconfiguration (DR) operations through the HMC to move resources from a spare-pool to a partition or between partitions.

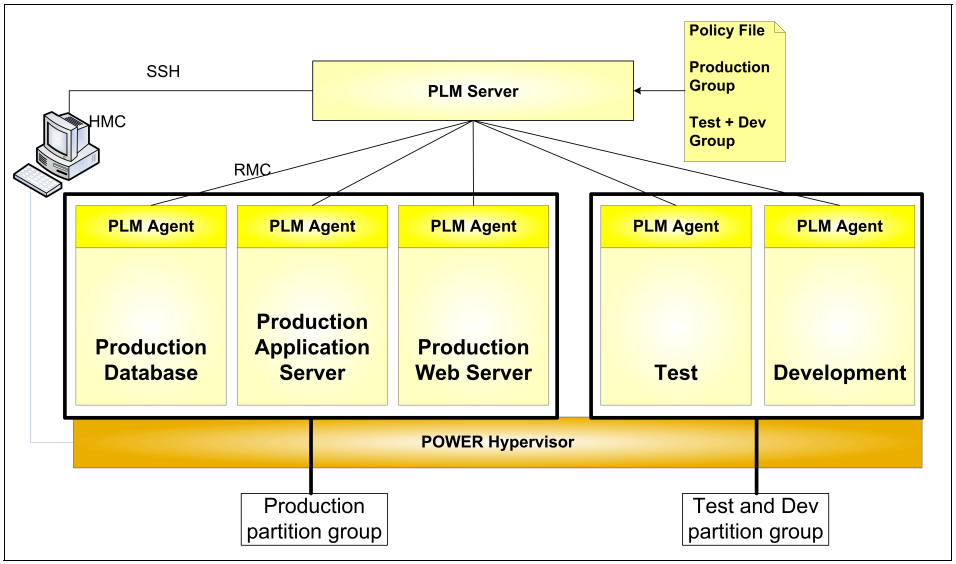


Figure 6-1 PLM architecture

PLM allows for groups of partitions. Resources within a group are managed independently. In Figure 6-1, two partition groups are shown, one for the production partitions, and the other for test and development.

Notes:

- The PLM server may reside either in a partition on the same server as the partitions being managed or on a different machine. When the PLM server runs in a partition, it is capable of managing its own partition.
- Multiple Partition Load Manager servers may be run on one AIX 5L system.
- Different PLM groups on a given server may be managed by different PLM servers.
- A partition can have, at most, one PLM manager.
- It is not required that all partitions in a system be managed.
- One Partition Load Manager server can manage partitions within only one managed CEC.
- It is not possible to have shared-processor and dedicated-processor partitions in the same PLM partition group.
- Resources are constrained to a group: A partition in one PLM group will never be given resources from another partition in another group.
- There should be at least two active partitions in a partition group.

Since each partition is monitored locally and the agents only communicate with the PLM server when an event occurs, PLM consumes a negligible amount of system and network resources.

6.1.3 Resource management policies

The resource management policy for each partition group is specified in a policy file that defines both the managed environment and the parameters of the resource management policy. The details of PLM policies are discussed in 6.2.5, “Define partition groups and policies” on page 389.

The different partition states and load thresholds are shown in Figure 6-2 on page 375. For each resource, there is an upper and lower load threshold. Every time a threshold is crossed, PLM receives a Resource Management and Control (RMC) event. When the load on the resource is above the upper threshold, the partition PLM considers the partition in need of additional resources; the partition is said to be a requestor. When the load on the resource is below the lower threshold, the partition becomes a potential donor. Normally, resources are only removed from donors when another partition enters the requestor state for the same resource. When the load on the resource is between the two thresholds, PLM considers that the resources available are adequate.

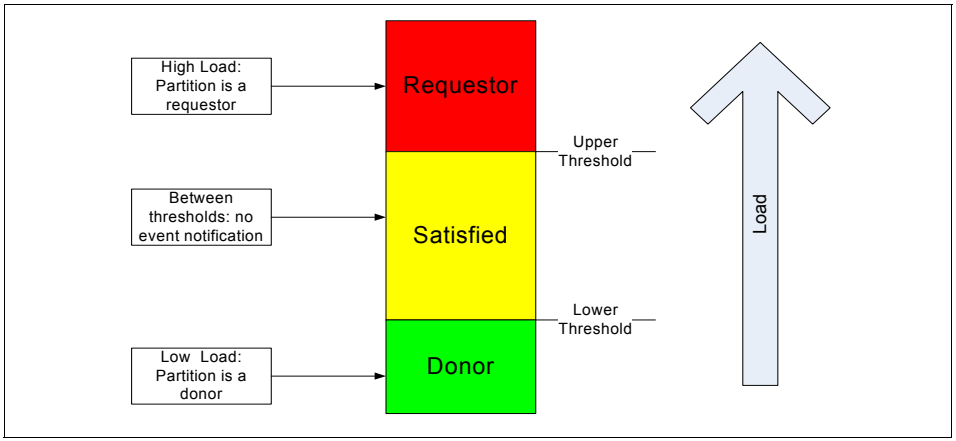


Figure 6-2 Resource utilization thresholds

There are four places where the partition group management policy can be specified. In order of increasing precedence, they are:

- The PLM defaults
- The PLM server instance defaults
- The group policy specification
- The partition policy specification

PLM has in-built defaults for tunable parameters. If these parameters have not been specified elsewhere, they will be used for the policy. The user can also specify default values for all groups managed by a server (PLM instance), for all the partitions in a given group (group policy), or for individual partitions. The values of the partition policy take precedence over all the others.

The policy file, once loaded, is static; a partition's priority does not change upon the arrival of high priority work. The priority of partitions can only be changed by loading a new policy. Policy files can be changed on-the-fly without stopping PLM.

Resource allocation

Part of a policy definition is the relative priority of each of the partitions in a group. This is done using a shares mechanism similar to that used in the AIX 5L Workload Manager (WLM). The greater the number of shares allocated to a partition, the higher its priority. To prevent some partitions from being starved PLM modulates the partition priority with its current resource amounts.

When PLM is notified that a partition has entered the requestor state, it will look for resources in the following order:

- Free pool of un-allocated resources.
- A resource donor.
- A partition with fewer shares for the requested resource that has more resources than specified by the value of its *guaranteed* configurable.

If there are resources available in the free pool, they will be given to the requestor. If there are no resources in the free pool, the list of resource donors is checked. If there is a resource donor, the resource is moved from a donor to the requester. The amount of resource moved is the minimum of the delta values for the two partitions, or the amount that would give them equal priority as specified by the policy. If there are no resource donors, the list of partitions with more resources than their guaranteed tunable is checked.

Determining which node is more or less deserving of resources is done by comparing how much of any resource a partition owns relative to its priority as specified by the number of shares. PLM calculates a ranking of partitions, including the requesting partition, from the list of partitions with excessive resources. A partition's priorities is defined as the following ratio:

$$\text{priority} = \frac{(\text{current\_amount} - \text{guaranteed\_amount})}{\text{shares}}$$

A lower value for this ratio represents a higher priority; partitions with lower value of priority can take resources from partitions with a higher value.

Figure 6-3 on page 377 shows an overview of the process for CPU resources in three partitions. Partition 3, under load, is a requestor. There are no free resources in the free pool, and there are not any donor partitions. PLM looks for partitions with excess resources (more resources than their guarantee). Both the other partitions in the group have excess resources. Partition 1 has the highest excess-to-shares ratio of all three partitions and resources will be moved from partition 1 to partition 3.

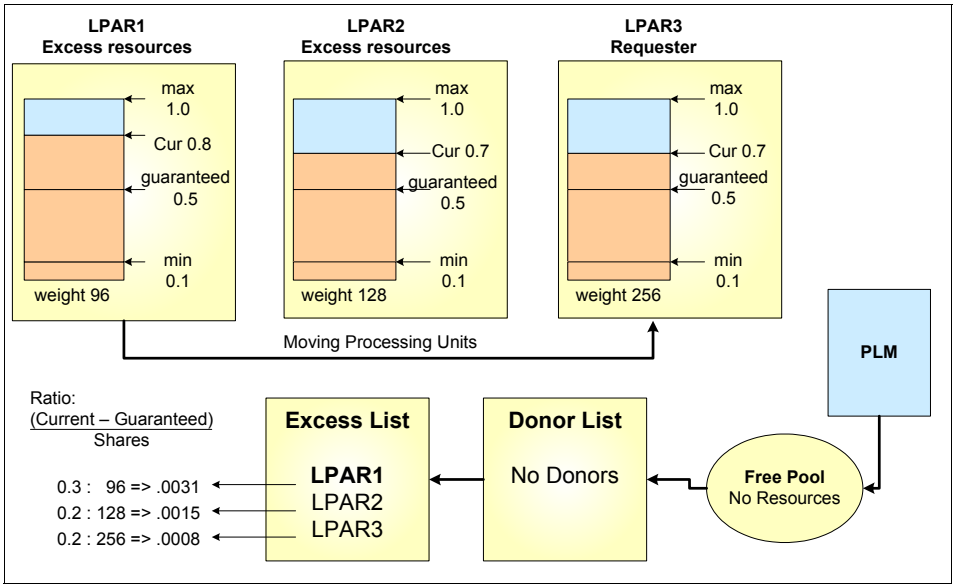


Figure 6-3 PLM resource distribution

If the request for a resource cannot be honored, it is queued and re-evaluated when resources become available.

Resource allocation constraints

There are number of considerations that must be taken into account when specifying PLM policies:

- The minimum, guaranteed, and maximum values must satisfy the following relationship: minimum <= guaranteed <= maximum.
- If the minimum, guaranteed, and maximum values all have the same value or the group maximum is set to zero, then PLM will not manage the resource.
- Independent of the priority, PLM will not let a partition fall below its minimum or rise above its maximum limit for each resource.
- The range of the PLM maximums and minimums should be a subset of the range of the maximums and minimums set on the HMC; if not, then the intersection of the PLM and HMC values is used.
- If you do not specify any values for the PLM maximums and minimums, they default to the values on the HMC.

6.1.4 Memory management

PLM manages memory by moving Logical Memory Blocks (LMBs) across partitions. The size of the LMB depends on the amount of memory installed in the CEC. It varies between 16 and 256 MB. The size of the LMB can be modified with the Advanced System Management Interface (ASMI) on the HMC.

To determine when there is demand for memory, PLM uses two metrics:

- Utilization percentage (ratio of memory in use to the amount of memory configured)
- The page replacement rate

Memory load is discussed in more detail in 6.6.2, “How load is evaluated” on page 433 and additional details of memory management are presented in 6.6.4, “Managing memory resources” on page 436.

AIX 5L will make use of all the memory made available to it. It will not move pages out of memory unless it needs to bring in other pages from disk. This means that even if there is excess memory, AIX 5L will use it, and it will be reported as used by the AIX 5L tools, even though there are no applications that are using it. Because of this, partitions will rarely become donors.

6.1.5 Processor management

For dedicated processor partitions, PLM moves physical processors, one at a time, from partitions that are not utilizing them or that have a higher excess weight, to partitions that have demand for them. This enables dedicated processor partitions running to better utilize their resources, for example, smoothing the transition from end-of-day transactions to the nightly batch jobs.

For shared processor partitions, PLM manages the entitled capacity and the number of virtual processors (VPs). When a partition has requested more processor capacity, PLM will increase the entitled capacity for the requesting partition if additional processor capacity is available. PLM can increase the number of virtual processors to increase the partition's potential to consume processor resources under high load conditions for both capped and uncapped partitions. Conversely, PLM will also decrease entitled capacity and the number of virtual processors under low-load conditions, to more efficiently utilize the underlying physical processors.

Note: The virtual processor folding optimization introduced in AIX 5L V5.3 ML3 renders the management of the virtual processor count by PLM unnecessary in most situations, but removing virtual processors is more efficient than VP folding, so in some circumstances management of virtual processors by PLM may be appropriate. Refer to “Virtual processor folding” on page 36.

Processor management is discussed in more detail in 6.6.3, “Managing CPU resources” on page 435.

6.1.6 Resource Monitoring and Control (RMC)

PLM uses the Resource Monitoring and Control (RMC) subsystem for network communication, which provides a robust and stable framework for monitoring and managing resources.

Resource Monitoring and Control (RMC) is the communications and event framework used by PLM. This section briefly introduces the key concepts and features of RMC necessary to understand how PLM works.

For a more thorough treatment of RMC, see *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615 and the AIX 5L product documentation.

RMC is a function that gives you the ability to monitor the state of system resources and respond when predefined thresholds are crossed, so that you can perform many routine tasks automatically. RMC is bundled with AIX 5L as a no-charge feature and is installed by default with the operating system. RMC is a subset function of Reliable Scalable Cluster Technology (RSCT).

RMC monitors *resources* (disk space, CPU usage, processor status, application processes, and so on) and performs an *action* in *response* to a defined *condition*. RMC can work in a stand-alone or a clustered (multi-machine or multi-partition) environment.

RMC enables you to configure response actions or scripts that manage general system conditions with little or no involvement from the administrator. For example, you can configure RMC to automatically expand a file system if its usage exceeds 95 percent.

When working with PLM, RMC is configured in a clustered environment as a management domain. In a management domain, nodes are managed by a management server. The management server is aware of all the nodes it manages and all managed nodes are aware of their management server.

However, the managed nodes know nothing about each other. The relationship between manager and managed is shown in Figure 6-4.

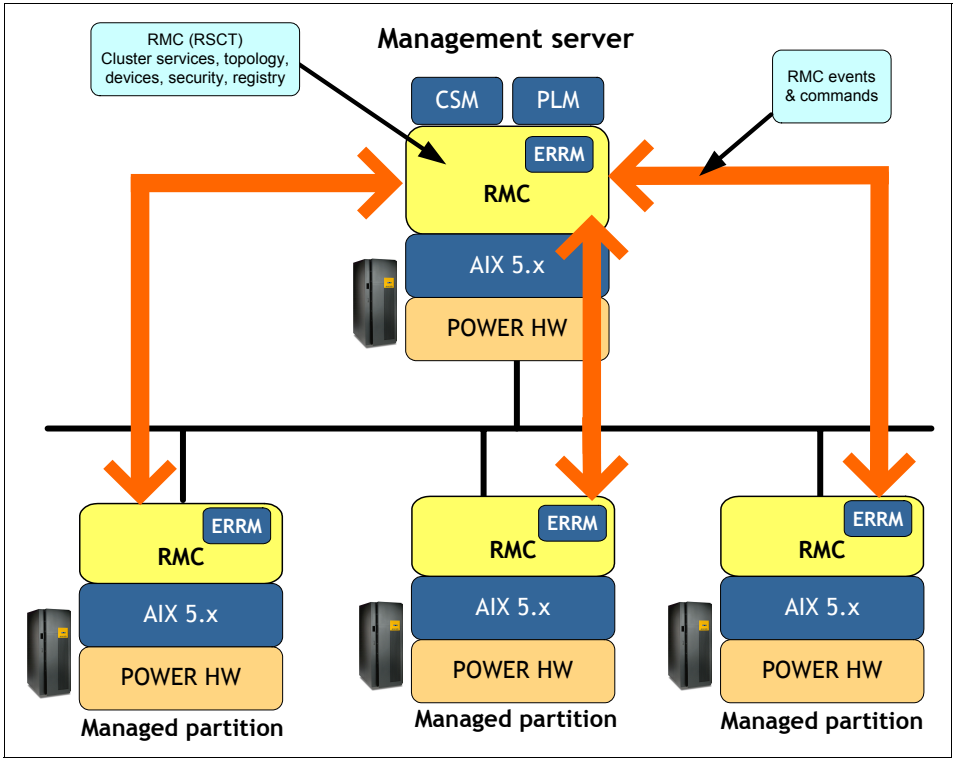


Figure 6-4 RMC management server and managed partitions

The management server may be hosted either in a partition or a remote machine. The management server can manage its own partition.

Note: CSM can run on the same machine as the PLM management server.

6.2 Installing and configuring Partition Load Manager

This section provides an overview of how to install and set up PLM, how to configure partition groups, and how to define PLM polices to manage memory and CPUs.

The following AIX 5L documentation provides additional information:

- *Introduction to pSeries Provisioning*, SG24-6389
- *AIX 5L V5.3 Partition Load Manager Guide and Reference,* SC23-4883

The following steps are necessary to install and configure PLM:

1. Prepare the AIX 5L environment.
2. Install and configure OpenSSL and OpenSSH.
3. Create the policy file.
4. Configure RMC.
5. Verify the installation.

Each of these steps is discussed in more detail in the sections that follow.

6.2.1 Preparing AIX 5L for PLM

This section describes the prerequisite preparation for PLM.

Name resolution

Before starting any of the following configuration tasks, you should decide at the outset if you are going to use fully qualified or short host names. Full-qualified names include appending the network domain to the host name, for example, my_server.my_domain.com. If you choose fully-qualified names, you should ensure that the name resolution mechanism you use returns fully-qualified names. The PLM server must be able to resolve the names of the controlling HMC and all the managed partitions.

Attention: To avoid PLM, RMC, and ssh naming difficulties, the network names should match the host names for the managed partitions, the PLM server, and the HMC.

Note: Though it is possible to create a specific AIX 5L user for running PLM, using the AIX 5L root user introduces fewer complications.

rsh and rcp

During its installation, the PLM server must be able to run remote shells on the managed partitions and copy files to them. PLM uses the AIX 5L remote shell, **rsh**, and remote copy, **rcp**, commands for this task, and these must be configured prior to installation. Once PLM is fully configured, then this remote access can be removed.

If **rsh** and **rcp** commands have been disabled for security reasons, use the following steps to enable these services:

1. Edit the .rhosts file for the root user ID on each managed partition to add the following lines:

   ```
   plmserver1 root
   plmserver1.mydomain.com root
   ```

   Some sites may prefer to edit the /etc/hosts.equiv file.

2. Enable the **rsh** and **rcp** commands on each LPAR by using the following commands:

   ```
   # chmod 4554 /usr/sbin/rshd
   # chmod 4554 /usr/bin/rcp
   ```

3. Edit the /etc/inetd.conf file, and uncomment the following line:

   ```
   shell stream tcp6 nowait root /usr/sbin/rshd rshd
   ```

4. Restart the inetd daemon by using the following command:

   ```
   # refresh -s inetd
   ```

5. Test the **rsh** command access from the Partition Load Manager server to each managed partition by using the following commands:

   ```
   # rsh root@lpar1 date
   # rsh root@lpar2 date
   ```

6.2.2 Install and configure SSL and SSH

Attention: On AIX 5L, Open Secure Shell (OpenSSH) relies on the Open Secure Sockets Layer (OpenSSL). You must install OpenSSL before installing OpenSSH. On the Virtual I/O Server, OpenSSL and OpenSSH are preinstalled as of VIOS 1.3.0.0

OpenSSL

OpenSSL provides the secure cryptographic libraries used by SSH and is available in RPM packages on the AIX Toolbox for Linux Applications CD, or you

can also download the packages from the following AIX Toolbox for Linux Applications Web site:

http://www.ibm.com/servers/aix/products/aixos/linux/download.html

OpenSSL is in the AIX Toolbox Cryptographic Content section of the Web site, in the box on the right-hand side of the page. You must have or obtain an IBM user ID to access this page. At the time of writing, the latest version available is 0.9.7l-1. Download and install the following RPM:

- openssl-0.9.7l-1.aix5.1.ppc.rpm

The two other OpenSSL packages, openssl-devel and openssl-doc, are not mandatory packages for using OpenSSH on AIX 5L. These are development tools and documentation for OpenSSH.

The following are the installation steps:

1. Use the **rpm** command to install the OpenSSL RPM package:

```
# rpm -Uvh openssl-0.9.7l-1.aix5.1.ppc.rpm
openssl
##################################################
```

2. If the package is correctly installed, you can verify the installation status using either of the following commands:

```
# lslpp -L | grep openssl
  openssl                  0.9.7l-1 C     R    Secure Sockets Layer
and
# rpm -q openssl
openssl-0.9.7l-1
```

OpenSSH

OpenSSH software tools provide shell functions to encrypt network traffic, authenticate hosts and network users, and ensures data integrity. PLM uses SSH to communicate with the HMC and the managed partitions.

For more information about OpenSSH on AIX 5L, see the following Web site, which has the latest **installp** format packages for AIX 5L:

http://sourceforge.net/projects/openssh-aix

At the time of writing, the latest package is version 3.8.1p1. Download the file openssh-4.3.p2_53.tar.Z for AIX 5L V5.3 into a directory of its own.

To install OpenSSH:

1. Uncompress and extract the files:

```
#zcat openssh-4.3p2_53.tar.Z | tar xvf -
```

2. Create a toc file with the **inutoc** command.
3. Install the package using **installp** or **smitty install_latest**:

   ```
   # installp -acv -Y -d . all
   ```

   The -d . means that you use the command in the directory you extracted the files to; if this is not the case, use the appropriate path. The -Y flag indicates that you accept the license.
4. Check the installation with the following command:

   ```
   # lslpp -L | grep ssh
     openssh.base.client     4.3.0.5300    C     F    Open Secure Shell Commands
     openssh.base.server     4.3.0.5300    C     F    Open Secure Shell Server
     openssh.license         4.3.0.5300    C     F    Open Secure Shell License
     openssh.man.en_US       4.3.0.5300    C     F    Open Secure Shell
     openssh.msg.CA_ES       4.3.0.5300    C     F    Open Secure Shell Messages -
     ...
   ```

The sshd daemon is under AIX SRC control. You can start, stop, and view the status of the daemon by issuing the following commands:

- **startsrc -s sshd** or **startsrc -g ssh** (group)
- **stopsrc -s sshd** or **stopsrc -g ssh**
- **lssrc -s sshd** or **lssrc -g ssh**

The IBM Redbook *Managing AIX Server Farms*, SG24-6606, provides information about configuring OpenSSH in AIX 5L and is available at the following Web site:

http://www.redbooks.ibm.com

Exchange SSH keys with the HMC

PLM uses SSH to securely run remote commands on the HMC without being prompted for a password. SSH must be configured to allow access from the PLM manager partition by the PLM administrator, the root user in this example. This is done by creating an SSH key-pair on the PLM server and exporting the public key to the hscroot user on the HMC. This is done as follows:

1. Log on to the PLM server partition with the root user ID.
2. Generate SSH keys on the Partition Load Manager server by using the following command:

   ```
   $ ssh-keygen -t rsa
   ```

   When prompted, leave the passphrase empty. The following output should appear:

   ```
   Generating public/private rsa key pair.
   Enter file in which to save the key (/.ssh/id_rsa):
   Created directory '/.ssh'.
   ```

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /.ssh/id_rsa.
Your public key has been saved in /.ssh/id_rsa.pub.
The key fingerprint is:
20:f5:d9:49:13:d7:2d:df:14:8c:a3:f6:ac:5e:d7:17 root@plmserver
```

The **ssh-keygen** command creates a .ssh directory in the home directory. The contents of the directory are the following:

```
# ls -l .ssh
total 40
-rw-------   1 root     system          883 Aug  7 21:59 id_rsa
-rw-r--r--   1 root     system          227 Aug  7 21:59 id_rsa.pub
```

The file id_rsa.pub contains the SSH public key.

Note: For stronger security, we recommend that you use DSA as the encryption algorithm. This can be accomplished by using the switch -t dsa instead of -t rsa with the **ssh-keygen command**. The resulting private and public key files will be called **id_dsa** and **id_dsa.pub** respectively.

3. Add the SSH public key to the HMC.

Important: Because the HMC may already have public keys from other partitions we must append our key to the HMC's existing public key list rather then just copy it.

a. Copy the HMC public key list to the PLM server; answer yes to the prompt:

```
$ scp hscroot@590hmc:.ssh/authorized_keys2 ~/.ssh/hmc_authorized_keys2
The authenticity of host '590hmc (192.168.255.69)' can't be established.
 RSA key fingerprint is 29:4b:1b:eb:1e:30:b6:da:ed:26:c7:0d:f6:2e:19:9a.
 Are you sure you want to continue connecting (yes/no)?yes
Warning: Permanently added '590hmc,192.168.255.69' (RSA) to the list of known
hosts.
hscroot@590hmc's password:
authorized_keys2                              100%    0     0.0KB/s   00:00
```

This command creates a known_hosts file in the .ssh directory:

```
$ ls -l .ssh
total 4
-rw-r--r--   1 root     system            0 Aug 09 22:07 hmc_authorized_keys2
-rw-------   1 root     system          883 Aug 09 21:59 id_rsa
-rw-r--r--   1 root     system          227 Aug 09 21:59 id_rsa.pub
-rw-r--r--   1 root     system          225 Aug 09 22:07 known_hosts
```

In the previous example, the hmc_authorized_keys2 file is empty, indicating that this is the first key exchange for the HMC.

b. Append the PLM server public key to the list of HMC public keys:

```
$ cat ~/.ssh/id_rsa.pub >>~/.ssh/hmc_authorized_keys2
$ ls -l .ssh
total 5
-rw-r--r--   1 root     system          227 Aug 09 22:16 hmc_authorized_keys2
-rw-------   1 root     system          883 Aug 09 21:59 id_rsa
-rw-r--r--   1 root     system          227 Aug 09 21:59 id_rsa.pub
-rw-r--r--   1 root     system          225 Aug 09 22:07 known_hosts
```

Copy the updated public key list back to the HMC:

```
$ scp .ssh/hmc_authorized_keys2
hscroot@590hmc:.ssh/authorized_keys2
hscroot@590hmc's password:
hmc_authorized_keys2                    100%  227     0.2KB/s
00:00
```

c. Complete the key exchange and verify the SSH configuration.

The initial SSH exchange between two servers performs the SSH key exchange. Run the **ls** command remotely on the HMC by entering the following command on the PLM server (you should not be prompted for a password):

```
# ssh hscroot@590hmc ls
WebSM.pref
websm.script
```

This **ssh** command should be repeated with the fully qualified name and the IP address of the HMC to have a complete key exchange. This will reduce the likelihood of problems during the PLM and RMC configuration:

```
$ ssh hscroot@590hmc.mydomain.com ls
$ ssh hscroot@192.164.10.10 ls
```

d. Obtain the name of the managed system.

Use the following command on the PLM server:

```
# ssh hscroot@p5hmc1 lssyscfg -r sys -F name
```

Unless the name of the managed system has been changed on the HMC using the **Properties** tab on the managed system, the default managed system name is similar to the following:

```
Server-9119-590-SN02Cxxxx
```

Tip: The very same procedure can also be used for key exchange with the Virtual I/O Server. This enables automated logins to the Virtual I/O Server, particularly useful for scripting purposes.

Since the default SSH installation on the Virtual I/O Server does not allow the padmin user to use **ssh** itself, which includes the use of **scp** from or to remote servers, it may prove useful to install links to the **ssh** and **scp** commands in a directory in the PATH of the padmin user.

To do so, use the following steps:

1. Change to the root shell:

```
$oem_setup_env
#
```

2. Establish links from the installed commands to a directory in the path of the padmin user:

```
# ln -s /usr/bin/ssh /usr/ios/oem/ssh
# ln -s /usr/bin/scp /usr/ios/oem/scp
```

3. Exit the root shell:

```
#exit
$
```

From now on, it is possible to use **ssh** and **scp** to copy files to and from the Virtual I/O Server and switch off the **telnet** and **ftp** services completely.

6.2.3 Configure RMC for PLM

The Partition Load Manager server uses Resource Monitoring and Control (RMC) to communicate with the managed partitions.

The RMC ACL setup has two components:

- Host authentication
- User authorization

The host authentication involves a public key exchange between the Partition Load Manager server and the managed partitions. This allows the PLM server to connect, or create a session, to the managed system.

The user authorization involves adding an entry to the RMC ACL file and grants the root (on the PLM) access to the required resource class.

You can set up the RMC using a script located in the /etc/plm/setup directory or using the Web-based System Manager GUI, as described in step 10 on page 404 in 6.2.6, “Basic PLM configuration” on page 395.

The **plmsetup** script automates both these tasks using remote shell commands. The setup procedure takes the following as arguments:

- The user ID under which the Partition Load Manager is to run.
- The host name of the partition.
- Optionally, the name of the PLM server when the command is run on the managed partition.

To set up from the PLM server, use the following syntax:

```
/etc/plm/setup/plmsetup lpar_hostname root
```

The *lpar_hostname* is the name of the managed partition. The script should be run for all managed partitions.

If the remote shell is unavailable or not configured on the managed partitions, you can perform these tasks manually. Run the following shell script as the root user on the managing machine that will run the Partition Load Manager:

```
/etc/plm/setup/plmsetup lpar_hostname root plmserver
```

Where the lpar_hostname is the name of the managed partition and plmserver is the name of the system or partition hosting the plmserver.

If the plmserver is managing its own partition, you will see a message similar to the following:

```
rcp: /tmp/exec_script.335922 and /tmp/exec_script.335922 refer to the
same file (not copied).
```

This is a normal message.

After the script runs successfully, the RMC ACL (Access Control) file, found in /var/ct/cfg/ctrmc.acls, on the remote machine will have an entry similar to the following:

```
# tail -1 /var/ct/cfg/ctrmc.acls
        root@nimmaster      *       rw
```

This user ID is used to set up the RMC ACL files on the managed partitions.

Configure RMC for the Partition Load Manager by doing the following steps:

1. Select **Set up Management of Logical Partitions**. The authenticated user name is root.
2. Select **Automatically setup with each partition in the policy file**. The policy file name is /etc/plm/policies/plm_example.
3. Click **OK**.

This can also be done using the command line if you are the root user on the Partition Load Manager server. In order for this to run, you must have **rsh** and **rcp** access. After the setup has been run, you can delete the .rhosts file.

6.2.4 Installing the Partition Load Manager

To install the Partition Load Manager server, complete the following steps:

1. Mount the Partition Load Manager CD on your system.
2. Using either the **installp** command or the **smitty install_latest** fast path, install the following filesets:
   - plm.license
   - plm.server.rte
   - plm.sysmgt.websm

   You should set the field Accept Licence to yes in **smitty** or use the -Y flag with the **installp** command.

6.2.5 Define partition groups and policies

PLM uses a policy file that defines the partitions that are to be managed, their guaranteed entitlements, and their minimum and maximum entitlements.

The policy file is a standard AIX 5L flat file that may be edited by any text editor; however, the Web-based System Manager interface to PLM provides a wizard for defining the policy and filling out the policy file. This document only covers the use of the PLM wizard for policy definition.

Attention: If you edit the PLM policy by hand, be advised that the file has a strict, stanza-based structure. If this structure is not respected, then PLM will not be able to use it. You should make a copy of a known-good file and only edit the copy. You can use the **xlplm -C -p *policy_file*** command, where policy_file is the manually edited file to check the syntax of the policy file.

Define a PLM policy as follows:

1. Create a new policy file.
2. Define the global environment and, optionally, the global tunables.
3. Define the partition groups and, optionally, the group tunables.
4. Add partitions to the groups and, optionally, define the partition tunables.

This section provides a overview of the configuration parameters and tunable. For detailed configuration steps, refer to 6.2.6, “Basic PLM configuration” on page 395.

Configuration parameters and tunables

Before detailing the process of creating a policy file, an understanding of PLM tunables is required. As indicated above, tunables may specified in three different places in the PLM policy wizard, and in addition to these three, PLM has a set of default values for a certain number of them. Any tunable specified for a partition takes precedent over tunables specified for a group, which in turn takes precedence over the globals, which takes precedence over the PLM defaults.

The global, groups, and partitions each have their own tabs in the PLM policy wizard.

Note: The following sections describe each of the configuration and tunable parameters for PLM policy files. Though these lists may be difficult to learn, all of the cited parameters have default values. The only mandatory parts of a PLM configuration are:

- The PLM group name
- The type of the PLM group: dedicated or shared
- The list of partitions in the group

Configuration parameters

PLM has four parameters for each managed resource that control how PLM adds and removes memory and processors to and from a partition. These are shown in Table 6-1.

Table 6-1 PLM configuration parameters

Parameter	Min	Default	Max	Description
Memory minimum	-	-	-	The minimum memory capacity of the partition. PLM will never leave a partition with less than this amount. Defaults to HCM value.
Memory guaranteed	-	-	-	The guaranteed amount of memory. Defaults to HCM desired value.
Memory maximum	-	-	-	The maximum amount of memory resources PLM will allow a partition to have. Defaults to HCM value.
Memory shares	0	1	255	A factor used to specify how memory capacity in excess of the memory guaranteed is distributed to partitions in the group. Specifying a value of 0 indicates that a partition will never receive more than the guaranteed memory.
CPU minimum	-	-	-	The minimum CPU capacity of the partition. PLM will never leave a partition with less than this amount. Defaults to HCM value.
CPU guaranteed	-	-	-	The guaranteed amount of CPU whenever a partition's load is greater than the CPU load average low threshold. Defaults to HCM desired value.
CPU maximum	-	-	-	The maximum amount of CPU resources PLM will allow a partition to have. Defaults to HCM value.
CPU Shares	0	1	255	A factor used to specify how CPU capacity in excess of the CPU guaranteed is distributed to partitions in the group. Specifying a value of 0 indicates that a partition will never receive more than the guaranteed CPU.

What is the difference between guaranteed and minimum?
PLM will ensure that a partition has the *guaranteed* resources whenever the partition load is greater than the low resource usage threshold. When the resource utilization is less than this limit, the partition becomes a donor and PLM will remove resources until the *minimum*, hard-limit, is reached.

CPU tunables

Table 6-2 shows the tunables common to all processor resources, while Table 6-3 on page 393 shows the tunables specific to the virtual processors in shared-processors partitions. These tunables are optional, and they are used to customize the PLM policy. Not all of these tunables are applicable in the globals tab of the PLM configuration wizard.

Table 6-2 CPU-related tunables

Tunable	Min	Default	Max	Description
CPU notify intervals	1	6	100	The number of contiguous 10 second sample periods that a CPU-related sample must cross a threshold before PLM will initiate any action.
CPU load average high threshold	0.1	1.0	10.0	The processor load average high threshold value. A partition with a load average above this value is considered to need more processor capacity (requestor).
CPU load average low threshold	0.1	0.5	1.0	The CPU load average low threshold value. A partition with a load average below this value is considered to have unneeded CPU capacity (donor).
Immediate release of free CPU	-	no	-	Indicates whether or not unused excess processor capacity will be removed from the partition and placed in the shared processor pool. A value of no indicates unneeded CPU capacity remains in the partition until another partition has a need for it. A value of yes indicates unneeded CPU capacity is removed from the partition when the partition no longer has a need for it.

Table 6-3 Virtual-processor related tunables

Tunable	Min	Default	Max	Description
Entitled capacity delta	1	10	100	The percentage increase of CPU entitled capacity to add or remove from a shared processor partition. The value specifies the percent of the partition's current entitled capacity to add or remove.
Minimum entitlement per VP	0.1	0.5	1.0	The minimum amount of entitled capacity per virtual processor. This attribute prevents a partition from having degraded performance by having too many virtual processors relative to its entitled capacity. When entitled capacity is removed from a partition, virtual processors will also be removed if the amount of entitled capacity for each virtual processor falls below this number. The default value is 0.5. The minimum value is 0.1. The maximum value is 1.0.
Maximum entitlement per VP	0.1	0.8	1.0	The maximum amount of entitled capacity per virtual processor. This attribute controls the amount of available capacity that may be used by an uncapped shared processor partition. When entitled capacity is added to a partition, virtual processors will be added if the amount of the entitled capacity for each virtual processor goes above this number. Increasing the number of virtual processors in an uncapped partition allows the partition to use more of the available processor capacity.

Memory tunables

Table 6-4 shows the memory related tunables. All these tunables are applicable for the partition and group policies, but only a subset are used for the global definition.

Table 6-4 Memory related tunables

Tunable	Min	Default	Max	Description
Memory notify intervals	1	6	100	The number of contiguous 10 second sample periods that a memory-related sample must cross a threshold before PLM will initiate any action.
Memory utilization low	1	50	90	The memory utilization low threshold below which a partition is considered to have excess memory and will become a memory donor. The units are in percent. The minimum delta between memory_util_low and memory_util_high is 10 percent.
Memory utilization high	1	90	100	The memory utilization high threshold at which a partition is considered to need more memory. Units are in percent. The minimum delta between memory_util_low and memory_util_high is 10 percent.
Memory page steal high	0	0	2^{31}-1	The page steal rate threshold at which a partition is considered to need more memory. Units are page steals per second. The result of checking this threshold is logically ANDed with the result of the memory utilization high threshold when determining if additional memory is needed by a partition.
Memory free unused	-	No	-	Indicates whether or not excess memory capacity will be removed from the partition and placed in the spare memory pool. A value of no indicates unneeded memory capacity remains in the partition until another partition has a need for it. A value of yes indicates unneeded memory capacity is removed from the partition when the partition is no longer using it.
Memory delta	1	1 LMB	256	The number of megabytes of memory to be removed or added to a partition in any single DR operation. If the value is less than the system's logical memory block (LMB) size, the value is rounded up to the system's LMB size. If the value is greater than the system's LMB size but not a multiple of it, the value is rounded down to the nearest multiple of LMB size.

6.2.6 Basic PLM configuration

This section describes the steps required to set up an initial PLM configuration using the PLM configuration wizard in Web-based System Manager. The overall procedure is shown in Figure 6-5.

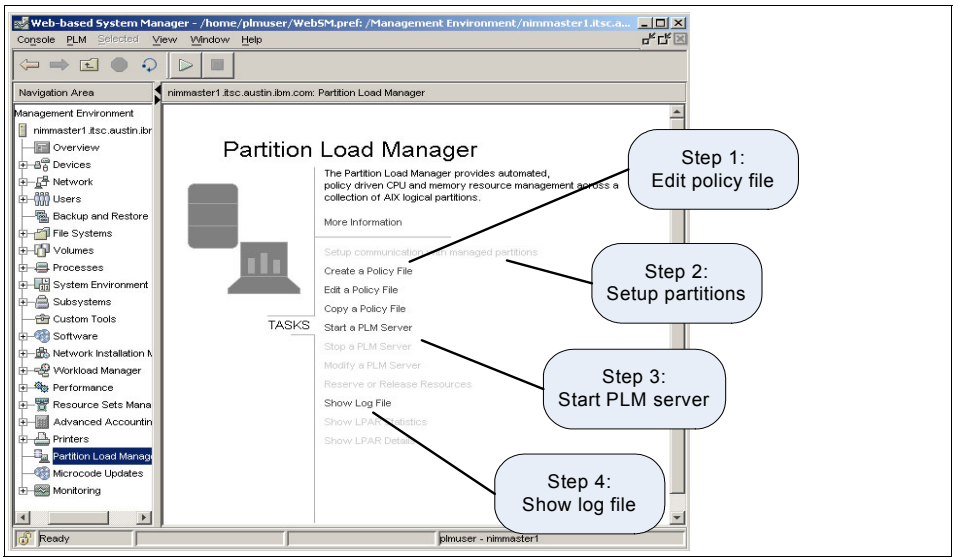


Figure 6-5 Steps required to set up PLM

Managing processor entitled capacity

In the first example, we configure PLM to manage the entitled capacity of two capped shared-processor partitions: plmserver and vio_client2. These two partitions have the definition on the HMC, as shown in Table 6-5.

Table 6-5 Shared processor partition initial configuration

Resource	Min	Desired	Max
Virtual CPU	1	1	30
Entitlement	0.1	3	5
Memory (MB)	256	512	1024

1. Start the Web-based System Manager.

   Use one of the following procedures.

   – Log on to the PLM server as root and run the **wsm** command. When you use the **wsm** command, you must position the DISPLAY environment variable to the name or IP address of your X11 terminal:

     ```
     $ export DISPLAY=my_xterm:0
     ```

     Where my_xterm is the name of your X11 terminal. You must include the `:0` at the end; this indicates the X11 window number.

   – Use the Java-based Windows® Web-based System Manager client (Downloadable from the HMC at `http://my_hmc/remote_client.html`, where my_hmc is the name or IP address of your HMC).

2. Start the PLM configuration wizard.

   Double-click the Partition Load Manager icon either in the list in the navigation area on the left-hand side or in the main window. This brings up the PLM window, as shown in Figure 6-6.

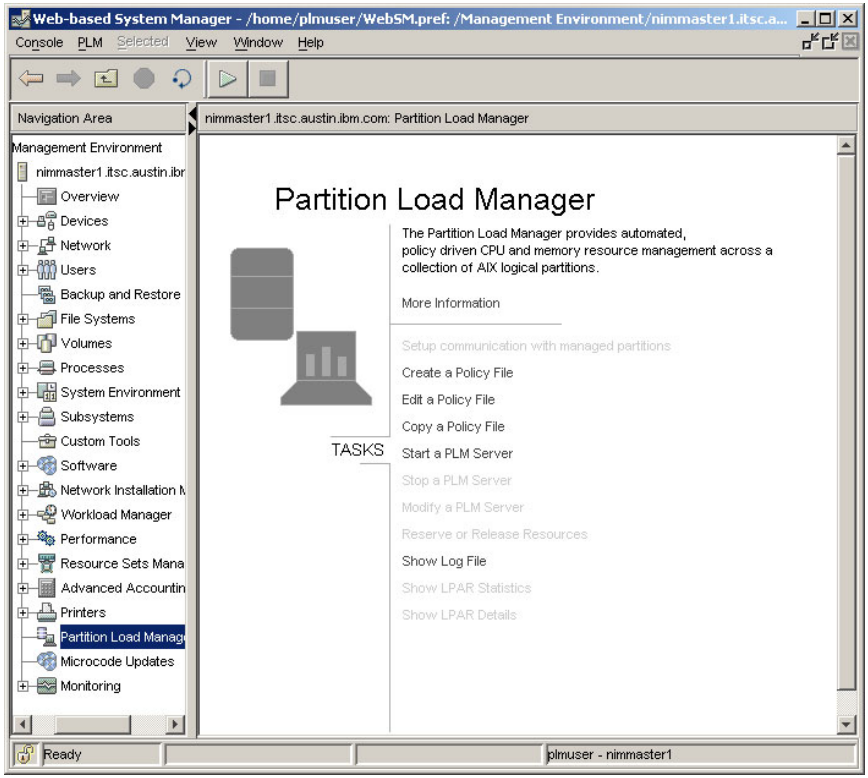


Figure 6-6 Partition Load Manager start-up window

3. Create a policy file using the wizard.

   Click on the **Create policy file** line, which will pop up the window shown in Figure 6-7.

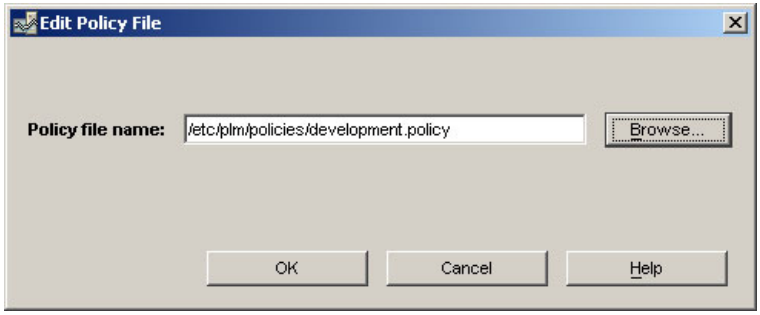


Figure 6-7 PLM wizard General tab of the Create Policy File window

The default location for policy file is /etc/plm/policies. You may create a policy file in any directory in which you have write access, but the file name in the create window should be an absolute path. In this example, the policy file is called development. The comment section is optional; in this example, we show which partitions the policy is for.

4. Complete the information in the Globals tab, as shown in Figure 6-8. The HMC name is the name or IP address of your HMC. Check again that the name you choose here can use the **ssh** command to the HMC without prompting for a key exchange or for a password.

   The HMC user name is the user that was used when setting up the ssh connection; this is typically hscroot.

   The CEC name is the name you received when testing the ssh connection between the PLM server and the HMC. Use the following command from the PLM server:

   ```
   # ssh hscroot@p5hmc1 lssyscfg -r sys -F name
   Server590
   ```

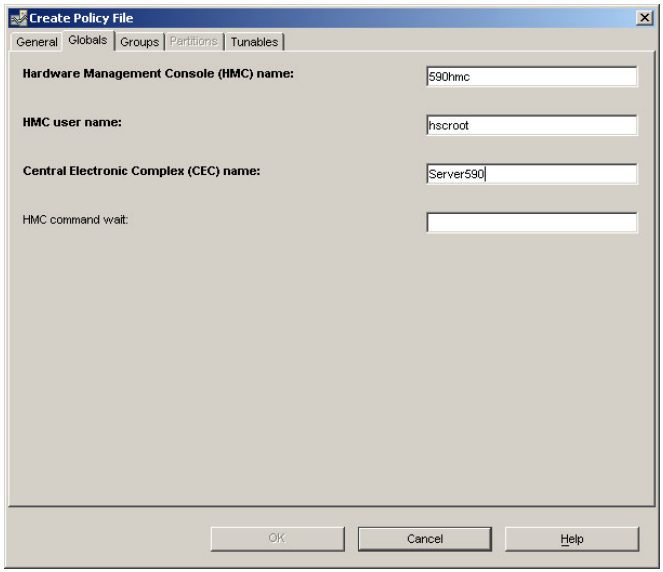


Figure 6-8 PLM wizard Globals tab of the Create Policy File window

   The HMC command wait field is used to tell PLM how long to wait, in minutes, for an HMC command before it times out. The default value is five minutes. The minimum value is 1, and the maximum value is 60.

5. Define the partition groups. All partitions belong to a partition group. All the partitions in any group are of the same type, dedicated or shared.

   Click the **Group** tab, then the **Add** button. This pops up the Group Definition window, as shown in Figure 6-9 on page 399. In this example, the group has been called development, and the maximum amount of physical CPU for the group has been set at 4, that is, the sum of the CPU entitlements of all the partitions in the group will not exceed four CPUs. We have turned off the management of memory resources by deselecting the check box.

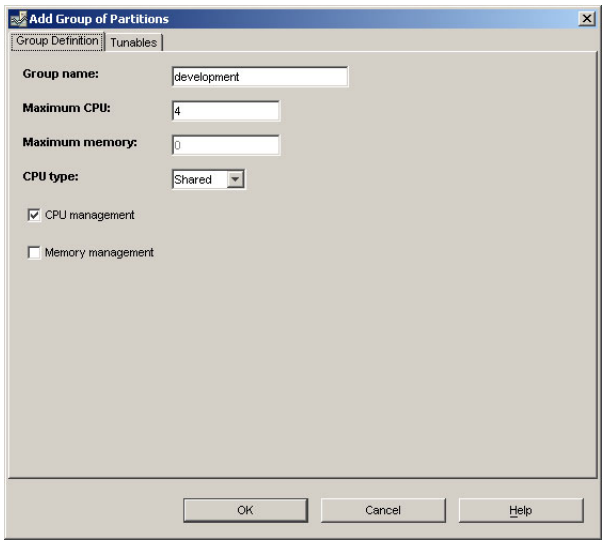


Figure 6-9 PLM wizard Group Definitions window

6. In our example, there are just two partitions, and we are going to set the same tunables for each partition, so we use group tunables. Click the **Tunables** tab and set the default policy values for all partitions in the group, which brings up the window shown in Figure 6-10.

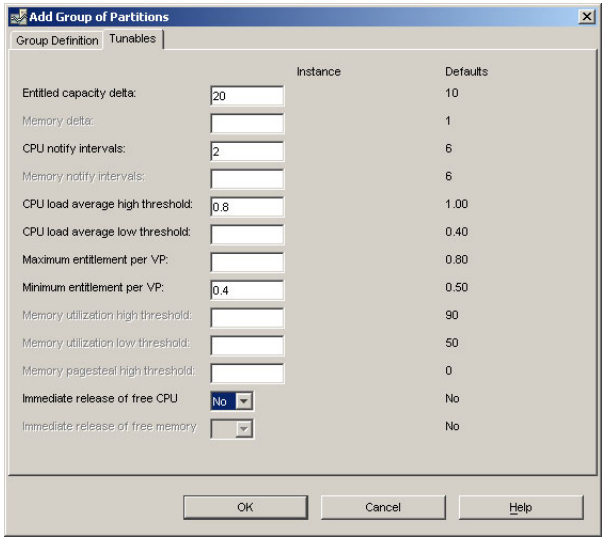


Figure 6-10 PLM Wizard Add Group of Partitions tunables window

Fill out the values for the tunable parameters. When you leave a value blank, it takes a default value. In this example, there are no Instance defaults, so all the PLM defaults, displayed on the right-hand side of the window, are used.

In this example, we have changed the entitled capacity delta from 10 percent to 20 percent, so PLM will add or remove 20 percent of the current resources when performing a DR operation. We have reduced the number of notify intervals to just two; this means that after two notifications for low CPU, PLM will try to find resources to help the partition. We have reduced the CPU load maximum to 80 percent, which means that when CPU load average attains 80 percent, the agent will send a notification to the PLM server. We have also reduced the minimum entitlement per virtual processor to 0.4.

Click on the **OK** button when you have finished and a summary of the group definition is shown (see Figure 6-11).

.

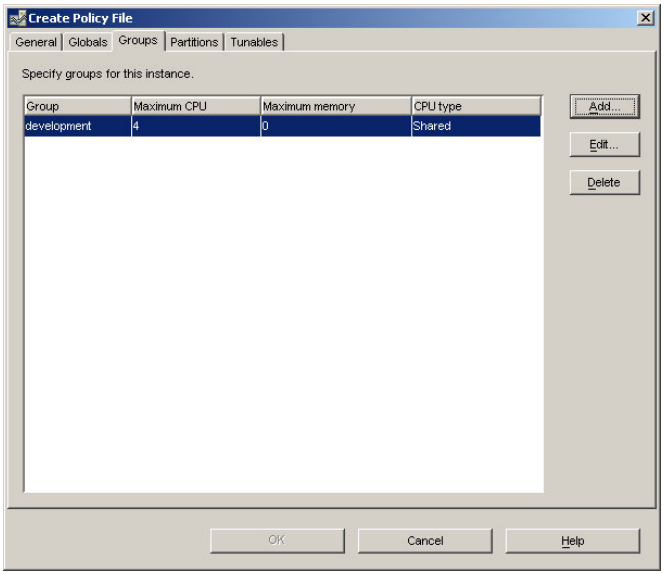


Figure 6-11 PLM wizard group definition summary window

7. Having defined the group, we must now specify which partitions are included in the group. To do this task, click the **Partitions** tab in the Create Policy File window and then the **Add** button of the pop-up window, as shown in Figure 6-12 on page 401.

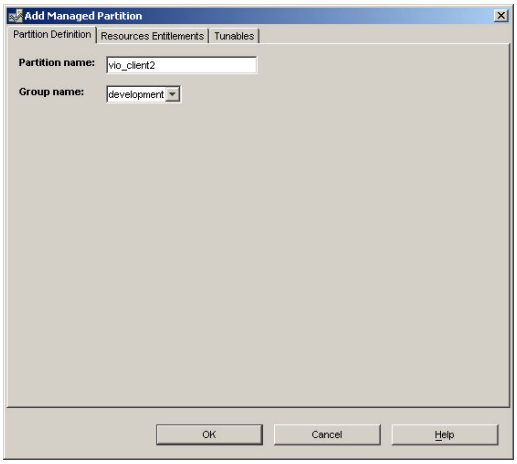


Figure 6-12 PLM wizard add managed partition window

There is a pull-down menu for the group name, which is useful when you have created several groups.

In the Partition name field, you type the host network name. This must be the same host name used when configuring RMC. If you used fully-qualified host names, this must fully qualified too.

Attention: The Partition name field in the user interface is *not* for the name of the partition defined in the HMC but for the network name.

8. Specify the CPU resource policy for this partition by clicking the **Resource Entitlements** tab of the Add Managed Partition window. This brings up the window shown in Figure 6-13. The default values for the minimum, guaranteed, and maximum entitlements are taken from the minimum, desired, and maximum values of the HMC partition definition. The default value for the CPU variable shares is one.

   In this example, we have two partitions in the group. We will set the variable shares so that one partition has twice the shares of the other. We will have one partition with 128 shares, and the other with 64. We will see how PLM uses this weighting when it is distributing resources between partitions.

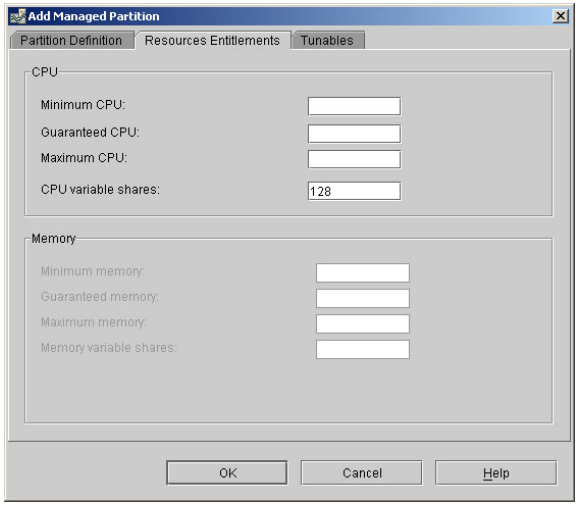


Figure 6-13 PLM wizard partition resource entitlement window

Note: The PLM tunable values used in the preceding examples have been chosen for the purposes of this demonstration only. They are not appropriate for most production machines.

9. Repeat the operation starting from step 7 on page 400 for the other partitions in the group. When you have finished the **Partitions** tab of the Create Policy File window, it should look similar to Figure 6-14 on page 403.

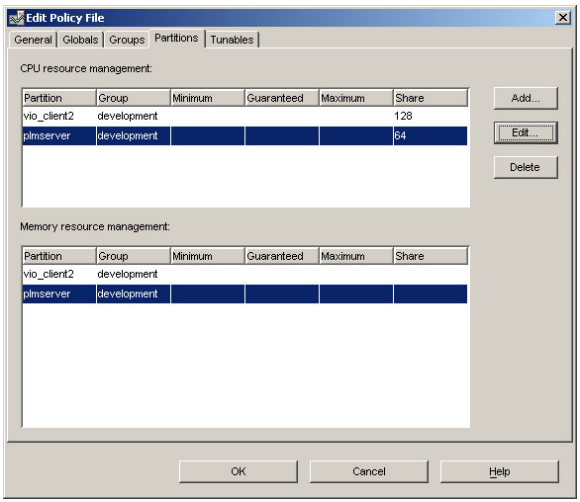


Figure 6-14 PLM wizard completed partitions window

You will notice that the memory resource management list is completed, even though there is no memory management for this group.

Clicking the **OK** button will create the policy file. The policy file is kept in the directory you specified at step 3 on page 397. The default directory is /etc/plm/policies. If you look at this file, you should find something similar to Example 6-1.

Example 6-1 PLM policy file for managing CPU resources

```
# cat /etc/plm/policies/development
#Policy for the development and admin partitions

globals:
        hmc_host_name = 590hmc
        hmc_user_name = hscroot
        hmc_cec_name = Server590

development:
        type = group
        cpu_type = shared
        cpu_maximum = 4
        mem_maximum = 0

vio_client2.mydomain.com:
        type = partition
        group = development
        cpu_shares = 128
        ec_delta = 20
        cpu_intervals = 2
        cpu_load_high = 0.8
        cpu_load_low = 0.3
        cpu_free_unused = yes

plmserver.mydomain.com:
        type = partition
        group = development
        cpu_shares = 64
        ec_delta = 20
        cpu_intervals = 2
        cpu_load_high = 0.8
        cpu_load_low = 0.3
        cpu_free_unused = yes
```

10. Once the policy file has been defined with the managed partitions, we can set up the inter-partition RMC communications if this was not done manually with the **plmsetup** command earlier. Click **Setup communication with managed partitions**, which will present the window shown in Figure 6-15 on page 405.

Use root as the Authenticated user name from the pull-down menu and type in or navigate with the **Browse** button to the policy file for the managed partitions. Click **OK**.

This operation can be performed manually, as described in 6.2.3, “Configure RMC for PLM” on page 387.

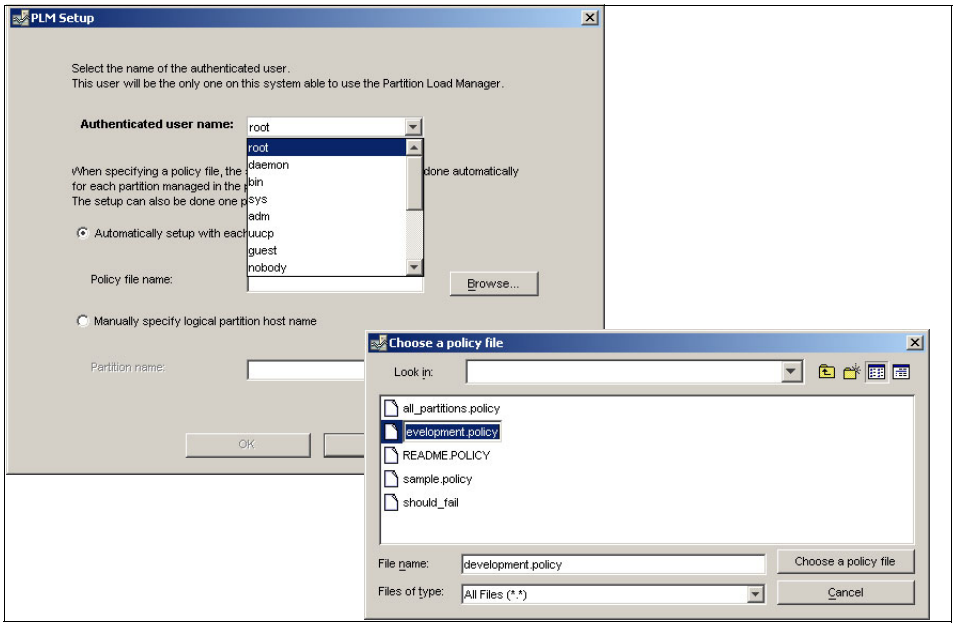


Figure 6-15 PLM setup communications with managed partitions window

11. We can now start the PLM server with our policy file. Make a note of the configuration of the managed partitions using the **`mpstat`** command

    PLM can be started from the command line or using the Web-based System Manager interface. In this example, we call the configuration RedbookServer.

    a. From the command line, run:

    ```
    # cd /etc/plm/policies
    # xlplm -S -p ./development.policy -l /var/opt/plm/plm.log
    RedbookServer
    ```

    The default operational mode is management, so we do not need to specify it on the command line. We can verify that the instance is running:

    ```
    #xlplm -Q
    RedbookServer
    ```

b. From the Web-based System Manager main PLM window, as shown in Figure 6-16.

   This performs the same operation as the command line: It creates a PLM server named RedbookServer in management mode, using the same policy and log files used in the command line. You can use the configuration with your own designated name or use the default.

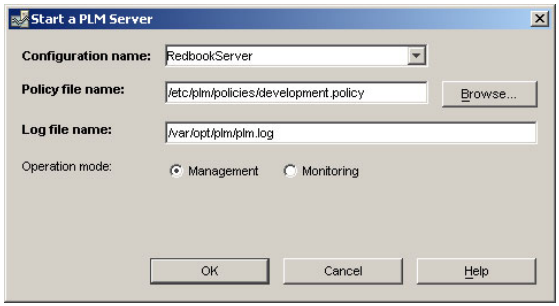


Figure 6-16 Starting a PLM server

The policy file stipulates that PLM will not start taking action before the CPU load reaches 0.8. Start a number of CPU intensive jobs in each partition and observe what action PLM takes.

You can observe PLM's actions by using the **tail** command to look at the log file:

```
# tail /var/opt/plm/plm.log
```

You can check the new partition configuration with the **mpstat** command.

12. From the PLM main Web-based System Manager window, you can show the PLM status and statistics and modify the PLM server. You can determine the names of the running instances using the following command:

```
# xlplm -Q
RedbookServer
```

13. Stop PLM from the main PLM Web-based System Manager window or use the following command:

```
# xlplm -K RedbookServer
```

Managing memory

Now we have management of processors in shared-processor partitions, we will move on to managing memory. Before continuing, you should stop the PLM server instance.

In this second example, we will configure PLM to manage the configured memory of two dedicated processor partitions: app_server and db_server. Because PLM cannot manage different processor types within the same group, we must create a new group. This new group can either be incorporated into the existing PLM policy or we could create a new PLM server instance to manage it.

In this example, we will choose the first option to show the management of two groups from a single PLM server.

The two dedicated processor partitions are configured identically on the HMC. This is shown in Table 6-6.

Table 6-6 Shared processor partition initial configuration

Resource	Min	Desired	Max
CPU	1	1	4
Memory (MB)	512	512	3076

The configuration steps are:

1. Edit the existing PLM policy file.

   Choose the **Edit a Policy File** from the main PLM window. This produces a window, as shown in Figure 6-17. Type in the file name of the previous policy file or navigate to it using the **Browse** button. Click **OK**, which pops up the **Edit Policy File** window.

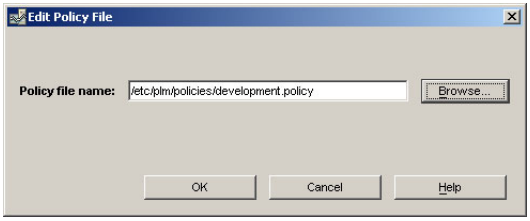


Figure 6-17 PLM wizard: Edit Policy File window

2. Click the **Groups** tab, and then the **Add** button. This brings up the window shown in Figure 6-18 (the same one used in step 5 on page 398 of the first scenario). In this example, we create a group called production. Click **OK**.

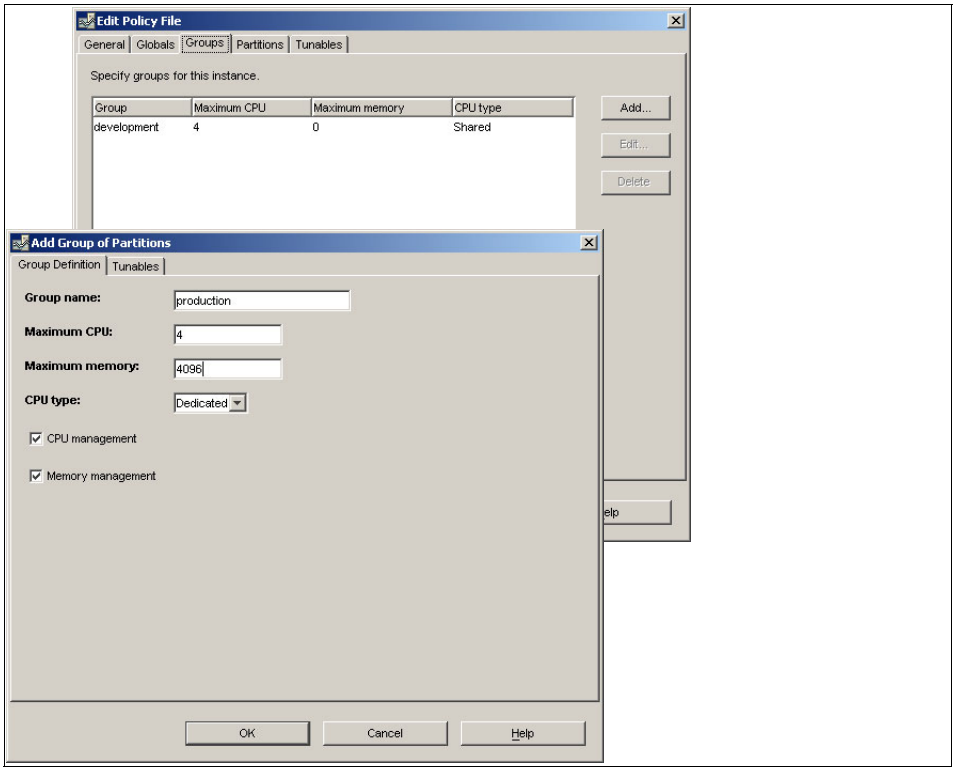


Figure 6-18 PLM dialog to add a group of partitions to an existing policy file

3. Define which partitions belong to the new group.

   Click the **Partitions** tab and then the **Add** button. Type the name of the managed partition in the first line and then click the **Group name** pull-down menu and select the group created in the previous step, as shown in Figure 6-19 on page 409.

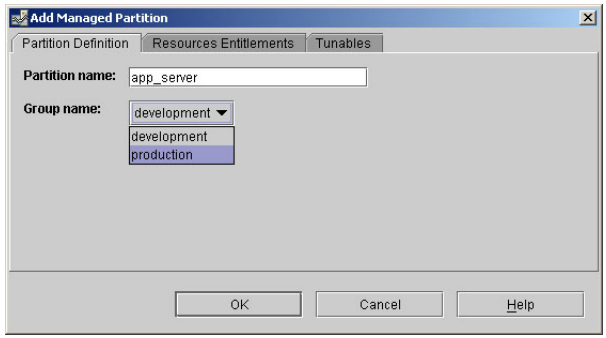


Figure 6-19 PLM wizard Add Managed Partition dialog

4. Set the PLM shares for CPU and memory. For this group, both partitions will be symmetrical, each with the same weights (128 for both CPU and memory), as shown in Figure 6-20 and Figure 6-21 on page 410. Click on **OK**.

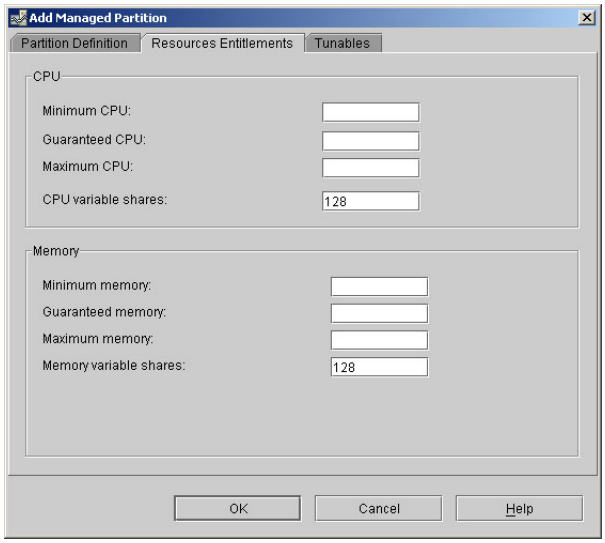


Figure 6-20 PLM wizard Resource Entitlements dialog

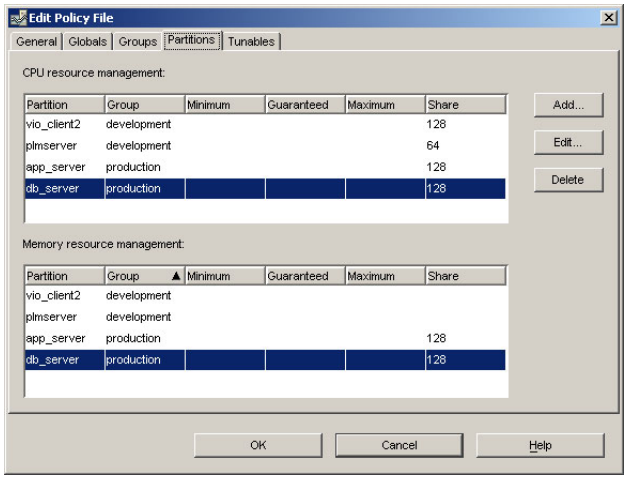


Figure 6-21 Edit Policy File partition summary window

5. Since the policy for both partitions is to be identical, we will define a group policy rather than individual partition policies. Click the **Groups** tab, which shows us a summary of all the defined groups. Click the production group to highlight it and then click the **Edit** button and in the window that pops up, click the **Tunables** tab, as shown in Figure 6-22.

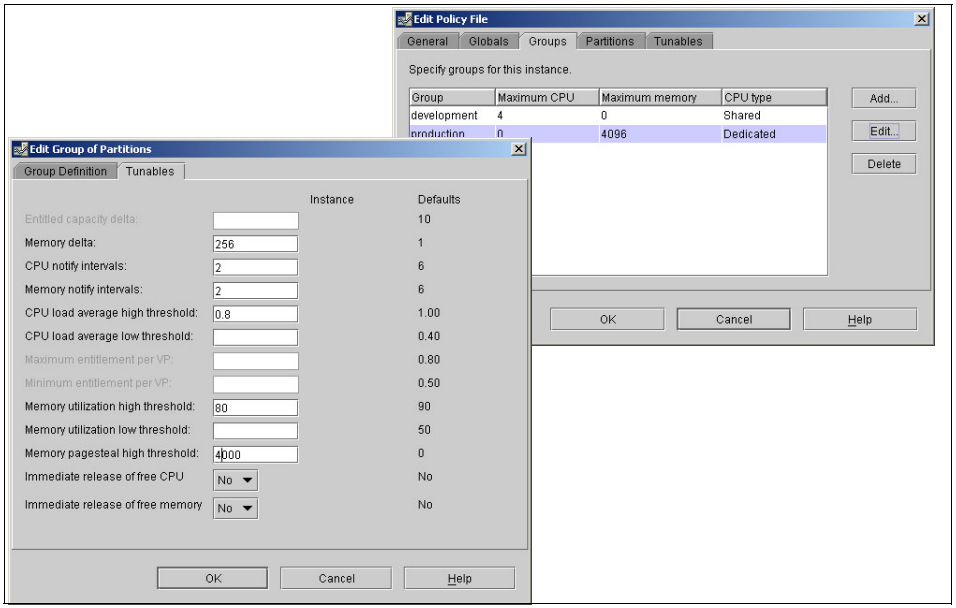


Figure 6-22 PLM wizard: setting the group tunables

In this example, the memory delta is set to 256 MB, so PLM will try to add (and remove) memory in 256 MB chunks to a partition. The notify count for both CPU and memory has been reduced from the default of six to two. The CPU load average high threshold has been reduced to 0.8, but the low threshold has been left at its default value of 0.4. The virtual processor thresholds are not available, as this partition group is for dedicated processor partitions. The memory high utilization threshold has been decreased to 0.8 and the page-steal high threshold is set at 4,000 page-steals per second.

When you have set up your policy, click **OK**.

6. Add the second partition to the group using the same procedure starting from step 3 on page 408.
7. With the partition groups defined, we can now restart the PLM server as described in step 11 on page 405 of the first example on page 405.

6.2.7 Partition Load Manager command line interface

PLM has two commands to control and monitor its operation through scripts:

xlplm	Controls and checks PLM.
xlpstat	Shows logical partition load statistics.

The **xlpstat** command is discussed in 5.5.7, “Monitoring with PLM” on page 348.

There are six forms of the **xlplm** command, depending on the operation to be performed. Possible operations are:

- Start a PLM server (-S).
- Stop a PLM server (-K).
- Modify a PLM server (-M).
- Reserve or release resources in or from a PLM group (-R).
- Query PLM status (-Q or -T).
- Verify the syntax and structure of a PLM policy file (-C).

Start a PLM server

The syntax of the **xlplm** command to start a PLM server is:

```
xlplm -S -p policy_file -l log_file [-o operational_mode]
[plm_instance]
```

The policy_file and log_file parameters are mandatory. The policy file specifies the policy to be used; it can be any valid policy file created either manually or with the PLM policy file wizard. This file must adhere to the PLM policy file stanza-based format. This format is described in detail in the README POLICY file in the /etc/plm/policies directory and is not presented in this redbook.

Using the -C switch, **xlplm** can check the syntax of the file. The syntax is:

```
xlplm -C -p policy_file
```

The log file holds the PLM activity log, and it can be any file you can write to. The PLM wizard proposes /var/opt/plm/, but this directory is only writable by the root authority by default.

The operational_mode is either monitoring, with a value N, or managing, with an M. The default operational mode is management (M).

The plm_instance parameter is the name to be given to the PLM server instance. The default value is default. A value must be supplied when multiple instances of PLM run on a single system or partition.

Stop a PLM server

Use the **xlplm -K [*plm_instance*]** command to stop a PLM server. If the PLM instance name was specified with the start command, then it must also be specified in the stop command. Use the **xlplm -Q** command to obtain a list of all running PLM server instances.

Modify a PLM server

The **xlplm -M** command can be used to:

- Change the active policy file.
- Change the log file.
- Change the operational mode: managing or monitoring.

The syntax is similar to the start PLM command:

```
xlplm -M { -p Policy } { -l Logfile } { -o {M|N} } [Instance]
```

You can change more than one aspect of the configuration in a single command.

The **xlplm -M** command can be placed in a crontab file to periodically change the PLM policy or rotate the log files.

Reserve or release resources for a PLM group

You can change the PLM group limits by reserving or releasing resources for and from a partition group. The syntax is:

```
xlplm -R { -c Amount | -m Amount } [-g Group] [Instance]
```

You can only specify a resource managed by the PLM server instance, that is, if the PLM server is not managing memory, you cannot ask to reserve or release memory resources.

List and query the running PLM server instances

There are two command forms to query PLM: -Q and -T. The former queries PLM instances; the latter shows the PLM default tunables. The syntaxes of the two command forms are:

```
xlplm -Q [ -v ] [ -r] [ -f plm_instance]
```

(The -v switch shows the current tunable values.)

and

```
xlplm -T [ -r ]
```

In both cases, the -r flag puts the output in a colon-separated raw format suitable for parsing.

In its short form, the **xlplm -Q** command lists all running PLM server instances. In its long form, the **xlplm -Q -f plm_instance** command prints key data pertaining to the given instance, as shown in Example 6-2. A value of 0 indicates that the PLM defaults are used.

Example 6-2 Querying PLM server instance status

```
# xlplm -Q -f dedicated
PLM Instance: dedicated

GROUP: group1
                   CUR       MAX      AVAIL      RESVD      MNGD
        CPU:      0.00      4.00       4.00       0.00       Yes
        MEM:         0         0          0          0        No

  app_server.mydomaincom

    RESOURCES:
                   CUR       MIN       GUAR       MAX       SHR
        CPU:      0.00      0.00       0.00       0.00        50
        MEM:         0         0          0          0         1

  db_server.mydomain.com

    RESOURCES:
                   CUR       MIN       GUAR       MAX       SHR
        CPU:      0.00      0.00       0.00       0.00       200
        MEM:         0         0          0          0         1
```

With the -r (raw) flag, the same data is presented, as in Example 6-3 on page 415.

Example 6-3 Querying PLM server instance status

```
# xlplm -Q -rf dedicated
#globals:hmc_host_name:hmc_user_name:hmc_cec_name:policy_file:log_file:
mode
globals:192.168.255.69:hscroot:Server590:/etc/plm/policies/dedicated:/e
tc/plm/policies/dedicated.log:manage
#group:group_name:cpu_type:cpu_maximum:cpu_free:cpu_reserved:mem_maximu
m:mem_free:mem_reserved
group:group1:dedicated:4.00:4.00:0.00:0:0:0
#partition:status:host_name:group_name:cpu_minimum:cpu_guaranteed:cpu_m
aximum:cpu_shares:cpu_current:mem_minumum:mem_guaranteed:mem_
maximum:mem_shares:mem_current:cpu_intervals:cpu_free_unused:cpu_load_h
igh:cpu_load_low:ec_per_vp_max:ec_per_vp_min:ec_delta:mem_int
ervals:mem_free_unused:mem_util_high:mem_util_low:mem_pgstl_high:mem_de
lta
partition:up:app_server.mydomain.com:group1:-1.00:-1.00:-1.00:50:0.00:-
1:-1:-1:1:0:2:0:1.00:0.40:0.80:0.50:10:6:0:90:50:0:1
partition:up:db_server.mydomain.com:group1:-1.00:-1.00:-1.00:200:0.00:-
1:-1:-1:1:0:2:0:1.00:0.40:0.80:0.50:10:6:0:90:50:0:1s
```

Check the PLM policy file syntax

The **xlplm -C -p *policy_file*** command checks the syntax of the policy file. It does not verify that the policy conforms to the partition definition in the HMC.

PLM demand measurement

To obtain PLM's view of system load for memory and processors, use the **lsrsrc** command, as shown in Example 6-4.

Example 6-4 PLM resource load

```
$ lsrsrc -Ad IBM.LPAR
Resource Dynamic Attributes for IBM.LPAR
resource 1:
        ConfigChanged = 0
        CPULoad       = 0.0878906
        CPUUtil       = 0.39
        MemLoad       = 92.5613
        MemPgSteal    = 149320
        CPUZone       = 2
        MemZone       = 2
```

6.3 Point-in-time and recurring reconfiguration

Point-in-time reconfiguration for partitions and PLM policies are both possible. These are described in this section.

6.3.1 Partition reconfiguration

The HMC provides a mechanism to perform point-in-time or periodic reconfiguration of the partitions it is controlling through the Scheduled Operations window of the HMC Configuration window, as shown in Figure 6-23.

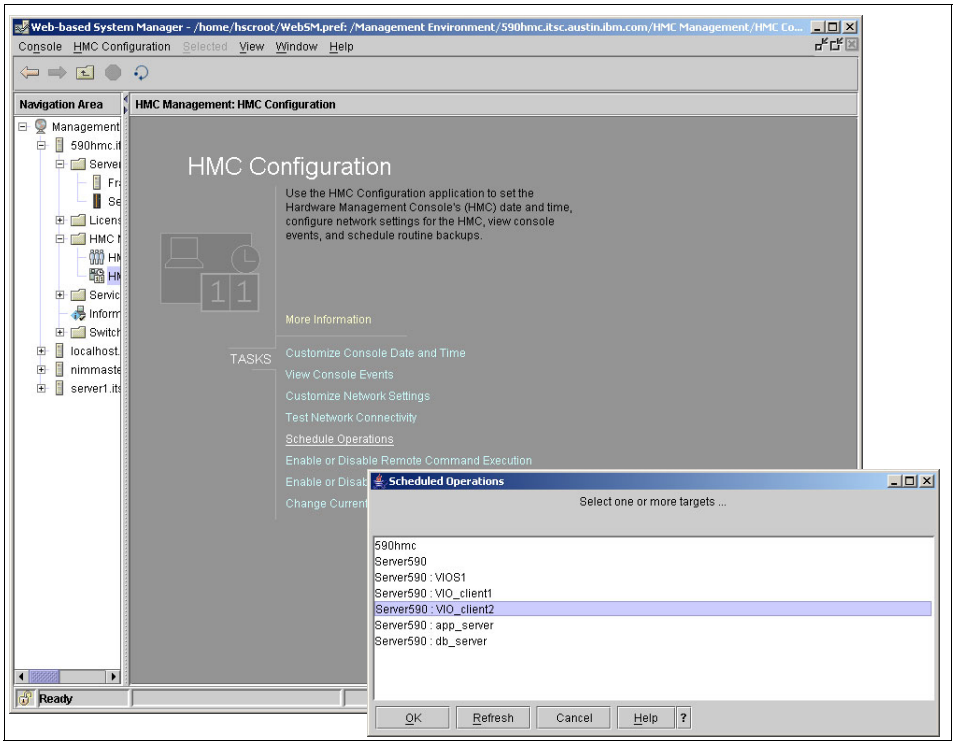


Figure 6-23 HMC Configuration and Schedule Operations windows

Select the partition for which you wish to program a dynamic reconfguration (in the above example, VIO_client2), and press **OK**, which will bring up the Customize Scheduled Operations window, as shown in Figure 6-24 on page 417, which pops up the Add a Scheduled Operation window, as shown in Figure 6-25 on page 417.

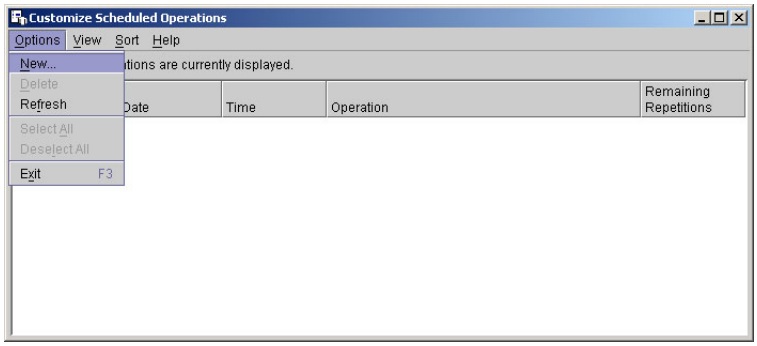


Figure 6-24 Customize Scheduled Operations window

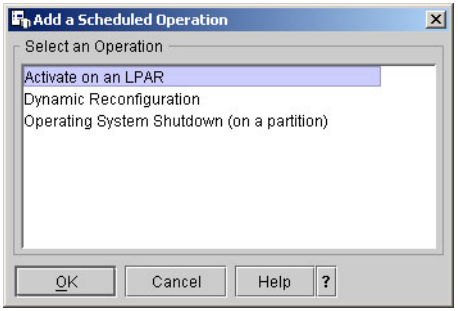


Figure 6-25 Add a Scheduled Operation window

The **Add a Scheduled Operation** window presents three different HMC operations to perform on the partition:

- Activate the partition.
- Reconfigure the system resources.
- Shut down the partition using an Operating System Shutdown.

Select the type of operation you wish to perform and click **OK**. The Setup Scheduled Operation window that is presented next has a similar form for all operations. It has two or three tabs labelled: Date and Time, Repeat, and Options (there are no options for Operating System Shutdown). The window for Dynamic Reconfiguration is shown in Figure 6-26. The date and time fields are mandatory. You also specify a time window in which the operation will start. The smallest time window is ten minutes.

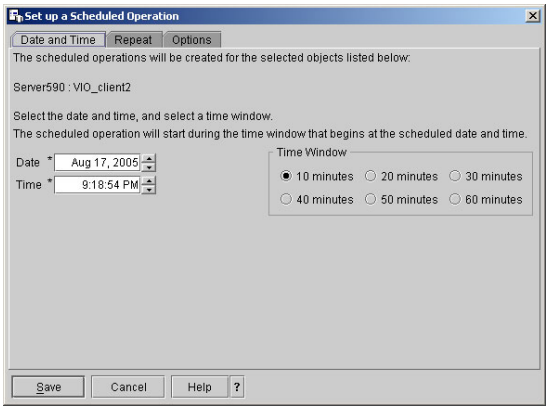


Figure 6-26 Setup a Scheduled Operation Date and Time tab

If you wish to periodically change the configuration, click the **Repeat** tab, which is shown in Figure 6-27. The repetition frequency can be daily or weekly. In this example the configuration will change every Friday for the next year.

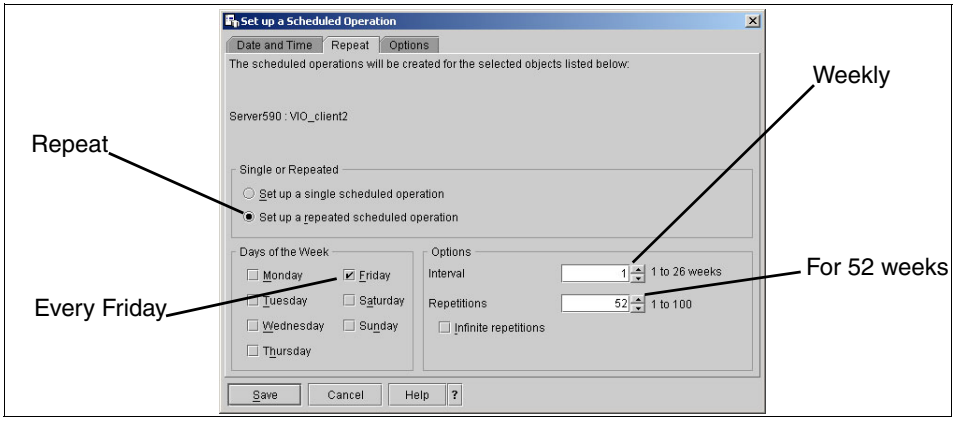


Figure 6-27 Set up a Scheduled Operation Repeat window

The options window, Figure 6-28, is used to specify what resources should be added or removed from the partition at the programmed time. In this example, 0.4 processing units will be moved from the partition VIO_client2 to the partition VIO_client1. Click the **Save** button, which adds this task to the list of programmed operations.

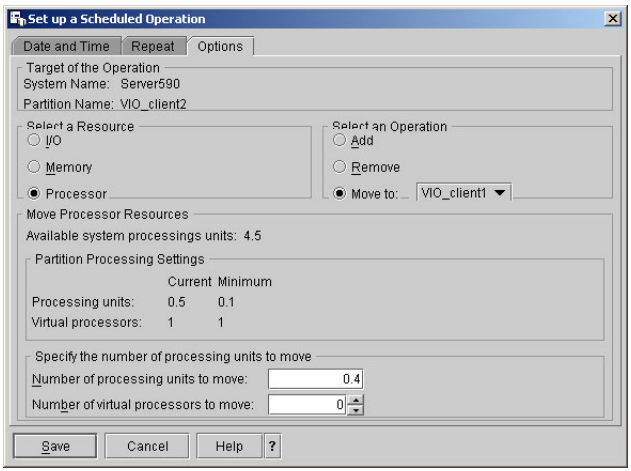


Figure 6-28 Set up a Scheduled Operation Options window

Important: If you schedule a periodic dynamic LPAR reconfiguration, as in this example, then you must program other reconfiguration operations to move the same amount of resources into or out of the partition; otherwise, there will inevitably be eventual depletion of resources.

The operation to start up or shut down a partition is similar to that for the reconfiguration.

6.3.2 PLM policy reconfiguration

To periodically change the PLM policy file, you must use the PLM command line interface with a scheduler, such as the AIX 5L **`cron`** or **`at`** commands. Use the **`xlplm -M`** command, as shown in the “Modify a PLM server” on page 412. Refer to the AIX 5L documentation for information about the **`cron`** and **`at`** commands.

6.4 Tips and troubleshooting PLM

This section provides additional hints and troubleshooting information for setting up the Partition Load Manager.

6.4.1 Troubleshooting the SSH connection

To check the SSH setup, run the **ssh** command. Make sure the HMC does not query for a password; otherwise, check the setup described earlier again.

```
# ssh hscroot@192.168.255.69 date
Fri Aug 19 01:29:19 CDT 2005
```

If your SSH connection from the PLM server to the HMC is not working, ensure that you checked **Enable remote command execution** on the HMC configuration window.

You also have to check whether the HMC firewall function allows incoming SSH traffic. On the Web-based System Manager, select **HMC Configuration** in the HMC Management menu and choose **Customize Network Settings**. Choose the LAN Adapter tap on the Customized Network Settings window shown in Figure 6-29.

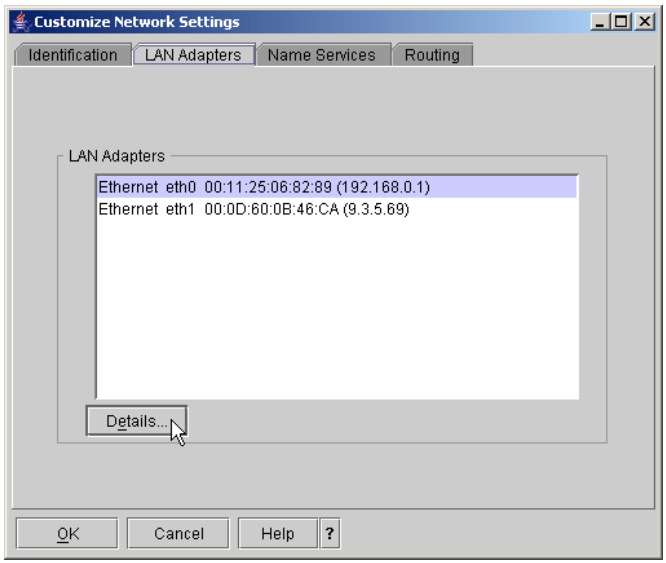


Figure 6-29 Customize Network Setting menu: selecting the LAN Adapters

Select the configured network adapter that attaches to the external network you use for your **ssh** command and click **Details**. Choose the **Firewall** tab on the LAN Adapter Details window shown in Figure 6-30.

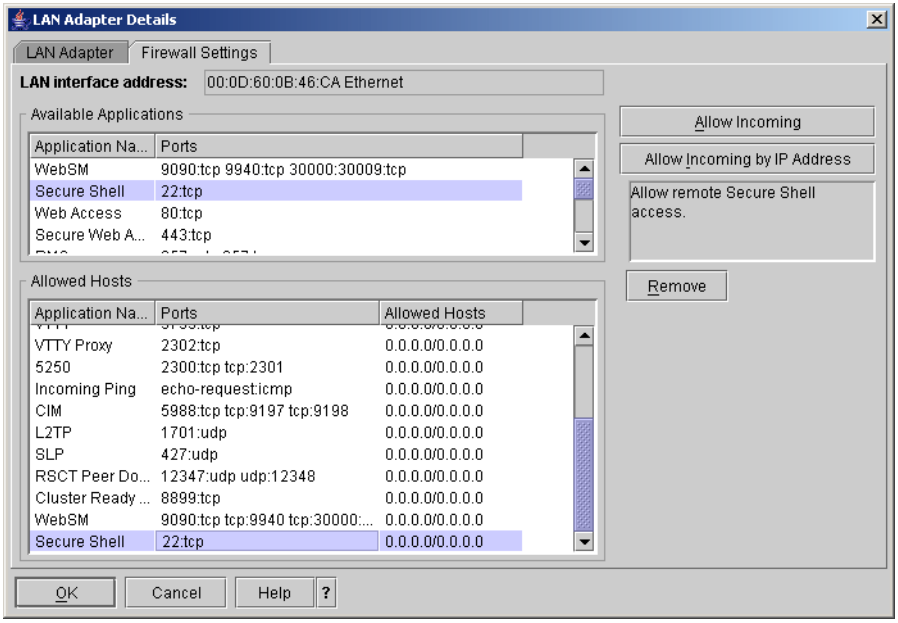


Figure 6-30 Firewall Settings window

Check if the Secure Shell application is added to the Allowed Host section on the second half of the window. If it is not added, select the **Secure Shell** from the Available Applications section above and press the **Allow Incoming** or **Allow Incoming by IP Address** button, if you want to limit the access to a specified IP address or address range.

Some changes to the Ethernet address configuration of the HMC will require a reboot of the HMC.

When the ssh connection is working without querying for a password, but PLM fails to start with the following error message, then you missed the key exchange step for the HMC name defined in your profile:

```
1498-057 Failed to execute ssh command for hscroot@192.168.255.69.
Verify the path, permissions, and user authorization for this command.
The version number 1 for is not valid.
```

Note: Exchanging keys is an important step. If you do not exchange keys, PLM will not start. The key exchange depends on the host name you use. If you use the short host name for the HMC entry in the PLM profile, exchange keys using the short host name. If you use full qualified host names for the HMC in the PLM profile, exchange keys using the full qualified host name.

For exchanging keys, use the **ssh** command.

If you want to check for which HMC name the SSH key are exchanged, look at the /.ssh/known_hosts file on the PLM server:

```
# cat /.ssh/known_hosts
590hmc,192.168.255.69 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAzLNslAT5xqQMqwPXEc9cTMiIaeO1ytHNvkH7Qf+e824
4jFemEdLQIJYlBoVihuQrgeMgsFiv1NvzpfqtW4Gxccr8J1EOT/7VKDp+2uJtUB4OEEC/9T
t6fYIKam2fSv6YWU4PtDbAWBeM3aKYZyRLLfShIzYDAk4BP56PVY1yLic=
590hmc.mydomain.com ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAzLNslAT5xqQMqwPXEc9cTMiIaeO1ytHNvkH7Qf+e824
4jFemEdLQIJYlBoVihuQrgeMgsFiv1NvzpfqtW4Gxccr8J1EOT/7VKDp+2uJtUB4OEEC/9T
t6fYIKam2fSv6YWU4PtDbAWBeM3aKYZyRLLfShIzYDAk4BP56PVY1yLic=
```

6.4.2 Troubleshooting the RMC connection

RMC requires public key exchanges when it is set up. This can be done by either Web-based System Manager or by running a shell script. In both cases, the managing machine must have **rsh** permission for the root user on all managed partitions.

Check for a .rhosts file or create one. The .rhosts file is only required for the setup. After the setup, you can delete it.

For detailed configuration steps, refer to 6.2.3, “Configure RMC for PLM” on page 387.

After setting up the RMC communication, check if the CT_CONTACT command is successful. Issue the command from the PLM server using the host name of one of the managed partitions.

To verify the RMC communication, enter the following command on the PLM server for all managed client partitions:

```
# CT_CONTACT=dbsrv lsrsrc IBM.LPAR
Resource Persistent Attributes for IBM.LPAR
resource 1:
        Name              = "DB_Server"
        LPARFlags         = 7
        MaxCPU            = 10
        MinCPU            = 1
        CurrentCPUs       = 2
        MinEntCapacity    = 0.2
        MaxEntCapacity    = 1
        CurEntCapacity    = 0.5
        MinEntPerVP       = 0.1
        SharedPoolCount   = 0
        MaxMemory         = 1024
        MinMemory         = 128
        CurrentMemory     = 512
        CapacityIncrement = 0.01
        LMBSize           = 16
        VarWeight         = 128
        CPUIntvl          = 0
        MemIntvl          = 0
        CPULoadMax        = 0
        CPULoadMin        = 0
        MemLoadMax        = 0
        MemLoadMin        = 0
        MemPgStealMax     = 0
        ActivePeerDomain  = ""
        NodeNameList      = {"dbsrv"}
```

If the output is similar to that shown here, the RMC setup was successful.

If you experience problems similar to this example, you have to check your RMC setup:

```
# CT_CONTACT=vio_client2 lsrsrc IBM.LPAR
/usr/sbin/rsct/bin/lsrsrc-api: 2612-024 Could not authenticate user.
```

You can check the setup of the RMC configuration with the **ctsvhbal** and **ctsthl** commands.

The **ctsvhbal** command shows the identity of the local system that is used for the Host Based Authentication (HBA) mechanism. PLM authentication is based on the first Identity entry shown in the output of the **ctsvhbal** command.

The Identity entry is either based on the **nslookup** result on the host name of the partition or on the host name specified in the /etc/hosts file if no DNS is defined. Be aware that the HBA mechanism picks the first name specified after the IP address in the /etc/hosts file. If you change this entry, you have to authenticate the new Identity.

In our example, we use the local /etc/hosts to resolve the host name. Although the **hostname** command returns the short host name plmserver, the **ctsvhbal** command returns the fully qualified name that is specified as the first entry in the /etc/hosts for the IP address 192.168.255.85:

```
# hostname
plmserver

# cat /etc/hosts | grep 192.168.255.85
192.168.255.85        plmserver.mydomain.com plmserver nimmaster1
vio_client1.mydomain.com vio_client1

# /usr/sbin/rsct/bin/ctsvhbal
ctsvhbal: The Host Based Authentication (HBA) mechanism identities for
the local system are:

                Identity:  plmserver.mydomain.com

                Identity:  192.168.255.85

ctsvhbal: In order for remote authentication to be successful, at least
one of the above identities for the local system must appear in the
trusted host list on the remote node where a service application
resides. Ensure that at least one host name and one network address
identity from the above list appears in the trusted host list on any
remote systems that act as servers for applications executing on this
local system.
```

For PLM order for remote authentication to be successful, the first Identity for the local system must appear in the trusted host list. Check the trusted host list using the **ctsthl** command using the -l flag the following way:

- On the PLM server, there should be an Identity entry for each managed partition you set up RMC communication for and the PLM server itself.
- On the managed partition, there should only be an entry for the PLM server and the managed partition itself.

The following example is the output of the **ctsthl** command on the PLM server:

```
# /usr/sbin/rsct/bin/ctsthl -l
ctsthl: Contents of trusted host list file:
--------------------
Host Identity:                 vio_client2.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200d1af82e0530f9e80ddd291814ccd6339b41f0731cb1c65046797875e81c1c7f83
a2d23288a10cb9a44d75727e212442ef45c789801cc50242c98bc03f7b41a9d0103
--------------------
Host Identity:                 plmserver.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
Host Identity:                 db_server.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200dc43ff154996991db8845b3d57b93cf37d674dc9bdab00f0b17923c9bc2a9f69b
f88cc5777e5af67d7d31d8e876aad2b0fac3f0f808db2ff9e286c143c8d97eb0103
--------------------
Host Identity:                 app_server.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200b04353a9afea953e25846b6fe65b5062a7c2591ea52112427b699287cae1d669f
9e27bcf126b4ab9c5e42326d1a278e1810e322fc0ec002b4febfa9f1b69c3630103
--------------------
Host Identity:                 nimmaster1
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
Host Identity:                 nimmaster1.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
Host Identity:                 192.168.255.69
Identifier Generation Method: rsa512
Identifier Value:
```

```
120200c10d3ad7a20951523b7c01bd31f421be4ec9f98b1bf2d3051edd044ad19250be8
c9db350ff6fdeb0ddb46bc1b3365d33bd194382bb71b4784d88d8e7d740d78d0103
--------------------
Host Identity:                loopback
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
Host Identity:                127.0.0.1
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
Host Identity:                192.168.255.85
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148
a64546f89a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
--------------------
```

If you want to delete wrong entries from your trusted host list, use the **ctsthl** command as follows:

```
# /usr/sbin/rsct/bin/ctsthl -d -n vio_client2.mydomain.com
ctsthl: The following host was removed from the trusted host list:
        vio_client2.mydomain.com
```

After you removed the Host Identity, you can either run the setup of your partition again to create the Host Identity or use the **ctsthl** command to add the Host Identity in the right way.

Using the **ctsthl** command, you have to provide the Host Identity, the Identifier Generation Method, and the Identifier Value.

```
# /usr/sbin/rsct/bin/ctsthl -a -n IDENTITY -m METHOD -p ID_VALUE
```

The easier way to re-add the identities to the trusted host list is to use the Web-based System Manager interface and re-use the window Management of Logical Partitions, as you did in the initial setup. Or use the plmsetup script on the PLM server as shown below:

```
# /etc/plm/setup/plmsetup vio_client2.mydomain.com root
```

Figure 6-31 on page 427 shows an overview of a working configuration. In this example, you can either use name resolution by /etc/hosts or DNS.

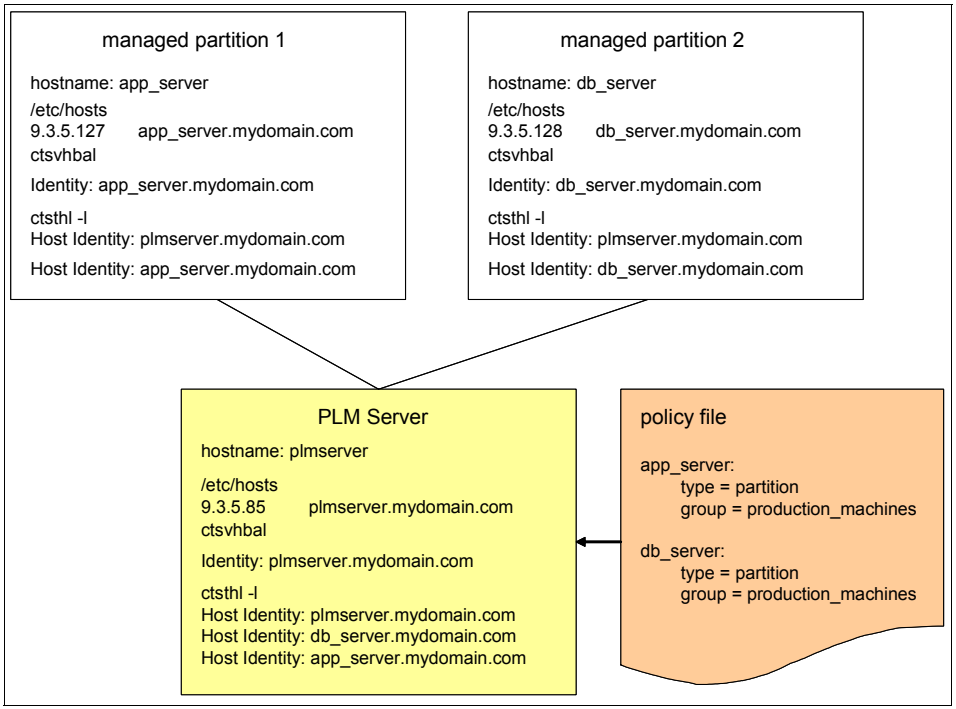


Figure 6-31 Configuration example for PLM RMC authentication

Tip: For easier configuration and debugging, keep the naming convention consistent. Use either short or full qualified host names for all managed partitions and the PLM Server. Check your /etc/hosts or DNS entry again when the problem persists.

6.4.3 Troubleshooting the PLM server

This section discusses various PLM problems when starting the PLM server and problems you may experience after the start of PLM.

- If you experience problems starting up the PLM server (you are experiencing an similar error to the following), check the managed system name on the HMC:

```
Status: Finished. Failed (5)

1498-031 Syntax error in policy, line 4.
```

PLM does not directly state a problem with the managed system name, but reports a problem in line 4. Check that the managed server name does not contain a blank or try a shorter managed system name.

- If your PLM server starts successfully, check the log file for any additional errors message. The updated version of PLM provides more information about the problems you may experienced and passes the HMC error messages. There are three types of message in the PLM log file:

  Trace Information messages showing the actions taken by PLM.

  Error Error conditions with the PLM server or the agents

  External Error conditions with the environment external to PLM, for example, error messages coming from the HMC.

- To define your managed partitions without the values for Resource Entitlement of that partition, PLM requests the values from the HMC and displays it in the Show LPAR Statistic window.

  If you experience a problem with PLM not showing the values (such as the maximum, minimum, and guaranteed values), then there may be a problem with the HMC communication. Check the log file for more information similar to the following:

  ```
  <08/18/05 12:33:21> <PLM_TRC> Cannot get the active profile for
  db_server.  lssyscfg returned 1.
  ```

  One problem that can appear is that your managed partition provides the wrong partition name (not host name) to the PLM server. The PLM server requests the partition name from the partition using the **`uname -L`** command. To check the partition name of your managed partition, use the following command:

  ```
  # uname -L
  5 db_server
  ```

  Note: This command returns the partition name defined on the HMC, not the host name of the partition. The host name of the partition and partition name on the HMC can be different.

  When you use the HMC to change the partition name, it is not updated on the partition until it is rebooted. PLM might then retrieve the former partition name and the HMC commands will fail because of the wrong partition name.

- If you see error messages similar to the following in your log file, check your RMC setup, as described in 6.2.3, “Configure RMC for PLM” on page 387 and 6.4.2, “Troubleshooting the RMC connection” on page 422:

```
<08/18/05 18:38:21> <PLM_TRC> Cannot establish an RMC session with
vio_client2. System call returned 39.
<08/18/05 18:38:21> <PLM_ERR> 2610-639 The user could not be
authenticated by the RMC subsystem.
```

6.5 PLM considerations

Consider the following when managing your system with the Partition Load Manager:

- The Partition Load Manager can be used in partitions running AIX 5L Version 5.2 ML4 or AIX 5L Version 5.3. Linux or i5/OS support is not available.
- A single instance of the Partition Load Manager can manage a single server. However, multiple instances of the Partition Load Manager can be run on a single system, each managing a different server or a different groups of partitions on the same server.
- The PLM cannot move I/O resources between partitions. Only processor and memory resources can be managed by Partition Load Manager.
- The PLM does not support dedicated and shared processor partition managed in one group. You need a group for dedicated processor partitions and a group for shared processor partitions.
- Management of the Virtual I/O Server or Partition Load Manager partitions using PLM is not provided.
- The Partition Load Manager supports a single HMC. You can create a second profile with the information of a second HMC, if one exists. In case of problems with the primary HMC, you can switch the profile manually.
- The Partition Load Manager requires the system to be managed by an HMC.

6.6 Resource management

Owners of IBM System p5 servers using AIX 5L V5.3 with the Advanced POWER Virtualization feature have the choice of three different workload management mechanisms:

Shared processor, uncapped partitions
: The POWER Hypervisor allocates unused processor cycles to those uncapped partitions that can make use of them.

Workload Manager (WLM)
: Prioritizes applications' access to system resources: CPU, memory, and I/O within a partition.

Partition Load Manager (PLM)
: Adds and moves CPU and memory resources to and between partitions using dynamic LPAR operations.

Note: AIX 5L (the scheduler with its process/thread priority mechanism and the virtual memory manager (VMM)) also manages resources, but this is not considered in this discussion.

PLM and WLM are only supported on AIX 5L; they are not supported on i5/OS or on Linux.

This section discusses the relationship between each of these mechanisms and how they may be used together to optimize resource usage and performance. An understanding of the PLM, WLM, and shared processor concepts is assumed in what follows. A comparison of these and other provisioning techniques is given in Chapter 4, "pSeries provisioning tools overview", in the *Introduction to pSeries Provisioning*, SG24-6389.

Figure 6-32 on page 431 shows the scope of each of the above resource and workload management mechanisms on IBM System p5 servers.

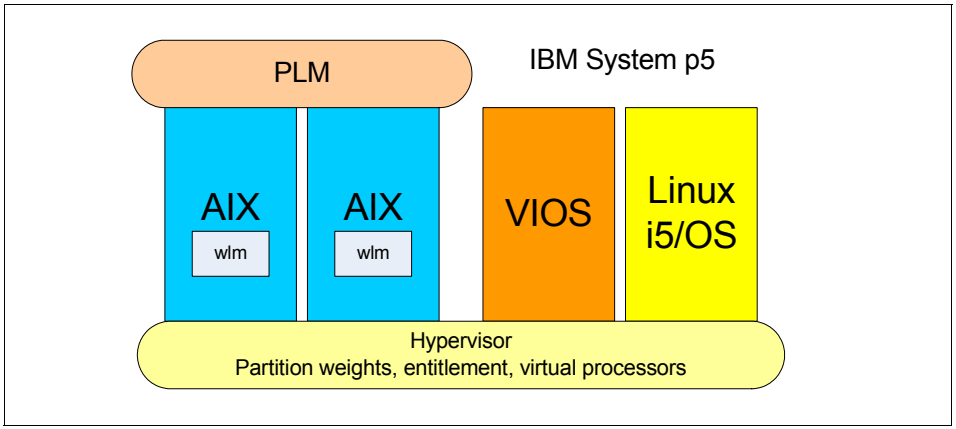


Figure 6-32 Resource and workload management mechanisms

6.6.1 Resource and workload management

There are two aspects of resource and workload management that are common to all the above mechanisms previously mentioned.

First, the workload and resource management mechanisms only take effect when there is a *shortage* of system resources and there is competition for these resources, that is, whenever the system resources are insufficient to meet the requirements of the workload and there are two or more active workloads. When the system is *normally* loaded or when there is only one resource consumer, then the resource managers do not have a significant effect even if they intervene.

Second, all workload and resource management mechanisms require an explicit policy describing the (relative) priorities of the managed components, which means identifying those components that will be penalized and those that will be given an advantage whenever there is a resource shortage.

Note: If there is a pool of free resources (unused memory or unused processors), then PLM will use these resources before removing resources from other partitions. The free pool comprises resources for which there is no competition.

Characteristics of WLM, PLM, and shared processors

Each of the available resource management mechanisms has its own characteristics that distinguishes it from the others.

WLM

WLM monitors and manages the use of CPUs, memory, and disk I/O rates within a stand-alone AIX 5L system or partition. It can limit the use of these resources by particular processes or groups of processes. WLM places each process in a class (a class may hold more than one process). Each class is allocated a number of shares. The ratio of the shares owned by a class to the sum of the shares of all active classes gives the proportion of the resources that the class will receive. The shares mechanism results in a self-adapting percentage.

In a virtualized environment, WLM is able to manage competing demands for memory, dedicated, and virtual processors, and disk I/O within a single AIX 5L instance.

In dedicated processor partitions, WLM can assign classes to specific CPUs using processor resource sets.

WLM periodically evaluates resource usage. It makes the necessary adjustments to usage targets at the end of each period by acting on the process priorities. WLM can manage resource usage conflicts as soon as they occur and for as long as they last.

Note:

- WLM only has visibility of resource usage within a partition. It has no visibility of what is happening in other partitions and cannot modify the priority of a partition relative to other partitions.
- WLM is provided as part of AIX 5L.

PLM

PLM manages memory, dedicated processor partitions, and shared processor partitions.

PLM manages partitions. It has no knowledge of the importance of any workload running in the partitions and therefore cannot readjust priority based on the changes of workload types.

PLM makes resource allocation decisions based on a policy file defined by the system administrator and will request the HMC to perform the appropriate dynamic LPAR operation.

PLM has a relatively high latency (in the order of minutes). This high latency makes PLM appropriate only for medium and long-term changes in resource usage and ineffective for managing short-lived peaks.

Shared processors

When using shared processor partitions, the POWER Hypervisor manages a pool of processors shared among a number of partitions. Unused processor cycles can be moved to uncapped virtual processors that require them, giving them more than they would normally be allowed.

The POWER Hypervisor has relatively low latencies. It has a scheduler dispatch cycle of 10 ms and can make changes to the dispatch in the middle of the cycle. Dynamic LPAR operations to increase the entitlement of virtual processors take a few seconds.

All virtual processors have to be scheduled on physical processors. Having a very high number of virtual processors (across all partitions) relative to the number of physical processors in the shared pool can degrade overall system performance. AIX 5L Version 5.3 Maintenance Level 3 significantly reduces the side effects of large numbers of virtual CPUs (refer to“Virtual processor folding” on page 36.

6.6.2 How load is evaluated

All resource and workload managers rely on a measure of resource usage. This may be an point-in-time measurement or a rolling average over a period of time. An understanding of how PLM, WLM, and the POWER Hypervisor measure resource usage is necessary to appreciate how they will interact.

Evaluating memory load

Determining precisely how much physical memory is being actively used by all applications in AIX 5L is difficult because AIX 5L’s strategy is to make the best use of all the resources put at its disposal and it will leave pages in physical memory even though they are no longer required.

As an alternative to measuring physical memory usage, as reported by the **vmstat** command, it is possible to infer the active memory occupation indirectly from the paging rates.

WLM uses the standard memory load statistics similar to those provided by the **vmstat** command. PLM uses both metrics for evaluating the memory load in a partition.

Evaluating CPU utilization

Traditionally, evaluating CPU utilization has been a simple exercise. It is sufficient to measure the amount of time a CPU is busy in a given interval. With multiple processor systems, the utilization on all CPUs is averaged to provide a single figure.

When sharing resources, a different approach is required and the POWER5 processor includes the new PURR register to measure CPU usage in a virtualized environment. Using shared processors in the PURR register and system monitoring is discussed in 5.5, “Monitoring a virtualized environment” on page 321.

Just because a CPU is close to or above 100% utilization in an uncapped, shared-processor partition does not necessarily indicate that there is shortage of CPU resources. It simply indicates that this partition is receiving more than its entitlement (guaranteed minimum). PLM and WLM use different strategies for evaluating the CPU load within a partition.

WLM measures resource utilization to determine if priorities need adjusting. PLM uses load because the resource utilization does not tell you if you need more resources.

PLM

PLM uses a figure called the *load average*. This gives an indication of the average length of the AIX 5L run queue over an interval of time (these are similar to the figures given in the first line of the **w** and **uptime** commands for 1, 5, and 15 minute intervals). PLM uses a weighted-average window, which allows it to cater for both short-term spikes and long-term trends, and normalizes the figure to the number of configured logical processors. The **lsrsrc -Ad IBM.LPAR** command displays the load average value used by PLM.

WLM

WLM periodically examines the real resources of the whole system and each of the WLM classes. If the CPU resources are not fully used, then WLM will not intervene. When the CPU occupation starts approaching 100%, WLM will start adjusting the priorities of all the processes in the classes so that the real CPU usage of each class approaches the specified target.

In the case of uncapped shared processor partitions, the class CPU consumption is calculated based on the CPU time of the partition:

$$\text{consumation(CLASS)} = \frac{\text{cpu\_time(CLASS)}}{\text{cpu\_time(PARTITION)}}$$

WLM's resource usage figure will never exceed 100%.

WLM and dynamic re-configuration

WLM is *dynamic LPAR aware*. It will automatically recalculate the resource usage targets as resources are added and removed from a partition.

6.6.3 Managing CPU resources

This section describes how PLM and WLM treat virtual processors in uncapped, shared-processor partitions. There are no special considerations for dedicated partitions, and capped, shared-processor partitions behave in much the same way as dedicated-processor partitions.

PLM and uncapped virtual processors

When the CPU load average in a partition managed by PLM rises through the PLM threshold, PLM will first attempt to increase the entitlement of the partition concerned and subsequently increase the number of virtual processors as appropriate if the condition persists, based on the active policy.

There are two tunable parameters in a policy that affect the addition and removal of virtual processors:

- Minimum entitlement per virtual processor
- Maximum entitlement per virtual processor

Whenever the entitlement is changed, PLM will add or remove the processing capacity specified by the entitled capacity delta tunable. When adding entitlement PLM, if the new entitlement per processor exceeds the maximum entitlement per virtual processor, PLM will add one or more virtual processors to the partition. When removing entitlement, if the new entitlement is lower than the minimum entitlement per virtual processor, PLM will remove one or more virtual processors to the partition.

WLM and virtual processors

Resource sets, along with the **`bindprocessor`** command and system call, allow processes and WLM classes (applications) to be bound to processors. This *hard locality* is usually done to ensure that the processor caches are kept *hot* with the applications data, thus improving performance.

Shared-processor partitions support processor binding (the commands and system calls will not return an error), but the effect on application performance will not be the same as for dedicated processor partitions:

- Though the Hypervisor tries to maintain affinity, there is not a guaranteed fixed mapping between virtual processors and physical processors.
- If the entitled capacity of a virtual processor is less than 1.0 (100 processing units), then the Hypervisor will use the same physical processor for other virtual processors (potentially from other partitions).
- Whenever a virtual processor uses less than 100% of a physical processor the spare physical-processor cycles are given to the shared pool for use by any other virtual processor.

6.6.4 Managing memory resources

PLM manages the memory of all *managed* partitions with memory management enabled. PLM will only move resources within a partition group; it will *not* move resources, memory, or CPU between partitions in two different partition groups.

6.6.5 Which resource management tool to use?

WLM manages resource usage conflicts within a partition. If the partition is known to have sufficient resources for the workload, or there is only one application running in the partition, there will be no competition for resources and WLM will have no role. WLM is most often used when consolidating several different applications on a single AIX 5L server or partition.

Shared-processor partitions and the POWER Hypervisor can move CPU resources almost instantaneously from one partition to another. Shared processor partitions are appropriate when there is marked, short-term fluctuation in the workload when consolidation in to a single AIX 5L partition is not appropriate.

Shared-processor partitions can be used when a partition requires a fractional part of a POWER5 processor, for example, two partitions each with an entitlement of 1.5 running on three POWER5 processors in a shared pool.

Partition Load Manager has a relatively long latency and cannot manage workload peaks that are of short duration. PLM manages the medium and long-term trends; it can handle the necessary migration of resources as operations move from the daily transactions to the overnight batch and back again.

PLM can be used to optimally configure servers with stable workloads. By setting the initial partition configuration with minimal resources, leaving unassigned resources in the free pool (memory and processor), PLM will move just enough resources (assuming that they are available) in to each partition to satisfy the workload. This alleviates the need to perform any precise estimation regarding the distribution of the server hardware between partitions.

For more information, refer to *Introduction to pSeries Provisioning*, SG24-6389.

Abbreviations and acronyms

ABI	Application Binary Interface
AC	Alternating Current
ACL	Access Control List
AFPA	Adaptive Fast Path Architecture
AIO	Asynchronous I/O
AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report
API	Application Programming Interface
ARP	Address Resolution Protocol
ASMI	Advanced System Management Interface
BFF	Backup File Format
BIND	Berkeley Internet Name Domain
BIST	Built-In Self-Test
BLV	Boot Logical Volume
BOOTP	Boot Protocol
BOS	Base Operating System
BSD	Berkeley Software Distribution
CA	Certificate Authority
CATE	Certified Advanced Technical Expert
CD	Compact Disk
CDE	Common Desktop Environment
CD-R	CD Recordable
CD-ROM	Compact Disk-Read Only Memory
CEC	Central Electronics Complex
CHRP	Common Hardware Reference Platform
CLI	Command Line Interface
CLVM	Concurrent LVM
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSM	Cluster Systems Management
CUoD	Capacity Upgrade on Demand
DCM	Dual Chip Module
DES	Data Encryption Standard
DGD	Dead Gateway Detection
DHCP	Dynamic Host Configuration Protocol
DLPAR	Dynamic LPAR
DMA	Direct Memory Access
DNS	Domain Naming System
DRM	Dynamic Reconfiguration Manager
DR	Dynamic Reconfiguration
DVD	Digital Versatile Disk
EC	EtherChannel
ECC	Error Checking and Correcting
EOF	End of File
EPOW	Environmental and Power Warning
ERRM	Event Response resource manager
ESS	Enterprise Storage Server
F/C	Feature Code
FC	Fibre Channel
FCAL	Fibre Channel Arbitrated Loop

FDX	Full Duplex
FLOP	Floating Point Operation
FRU	Field Replaceable Unit
FTP	File Transfer Protocol
GDPS	Geographically Dispersed Parallel Sysplex
GID	Group ID
GPFS	General Parallel File System™
GUI	Graphical User Interface
HACMP	High Availability Cluster Multiprocessing
HBA	Host Bus Adapters
HMC	Hardware Management Console
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
I/O	Input/Output
IBM	International Business Machines
ID	Identification
IDE	Integrated Device Electronics
IEEE	Institute of Electrical and Electronics Engineers
IP	Internetwork Protocol
IPAT	IP Address Takeover
IPL	Initial Program Load
IPMP	IP Multipathing
ISV	Independent Software Vendor
ITSO	International Technical Support Organization
IVM	Integrated Virtualization Manager
JFS	Journaled File System
L1	Level 1
L2	Level 2
L3	Level 3
LA	Link Aggregation
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LMB	Logical Memory Block
LPAR	Logical Partition
LPP	Licensed Program Product
LUN	Logical Unit Number
LV	Logical Volume
LVCB	Logical Volume Control Block
LVM	Logical Volume Manager
MAC	Media Access Control
Mbps	Megabits Per Second
MBps	Megabytes Per Second
MCM	Multichip Module
ML	Maintenance Level
MP	Multiprocessor
MPIO	Multipath I/O
MTU	Maximum Transmission Unit
NFS	Network File System
NIB	Network Interface Backup
NIM	Network Installation Management
NIMOL	NIM on Linux
NVRAM	Non-Volatile Random Access Memory
ODM	Object Data Manager
OSPF	Open Shortest Path First
PCI	Peripheral Component Interconnect
PIC	Pool Idle Count
PID	Process ID
PKI	Public Key Infrastructure
PLM	Partition Load Manager

POST	Power-On Self-test
POWER	Performance Optimization with Enhanced Risc (Architecture)
PPC	Physical Processor Consumption
PPFC	Physical Processor Fraction Consumed
PTF	Program Temporary Fix
PTX	Performance Toolbox
PURR	Processor Utilization Resource Register
PV	Physical Volume
PVID	Physical Volume Identifier
PVID	Port Virtual LAN Identifier
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
RCP	Remote Copy
RDAC	Redundant Disk Array Controller
RIO	Remote I/O
RIP	Routing Information Protocol
RISC	Reduced Instruction-Set Computer
RMC	Resource Monitoring and Control
RPC	Remote Procedure Call
RPL	Remote Program Loader
RPM	Red Hat Package Manager
RSA	Rivet, Shamir, Adelman
RSCT	Reliable Scalable Cluster Technology
RSH	Remote Shell
SAN	Storage Area Network
SCSI	Small Computer System Interface
SDD	Subsystem Device Driver
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor
SMS	System Management Services
SMT	Simultaneous Multithreading
SP	Service Processor
SPOT	Shared Product Object Tree
SRC	System Resource Controller
SRN	Service Request Number
SSA	Serial Storage Architecture
SSH	Secure Shell
SSL	Secure Socket Layer
SUID	Set User ID
SVC	SAN Virtualization Controller
TCP/IP	Transmission Control Protocol/Internet Protocol
TSA	Tivoli System Automation
UDF	Universal Disk Format
UDID	Universal Disk Identification
VIPA	Virtual IP Address
VG	Volume Group
VGDA	Volume Group Descriptor Area
VGSA	Volume Group Status Area
VLAN	Virtual Local Area Network
VP	Virtual Processor
VPD	Vital Product Data
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
VSD	Virtual Shared Disk
WLM	Workload Manager

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 446. Note that some of the documents referenced here may be available in softcopy only.

- *Advanced POWER Virtualization on IBM eServer p5 Servers: Architecture and Performance Considerations*, SG24-5768
- *AIX 5L Practical Performance Tools and Tuning Guide*, SG24-6478
- *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038
- *i5/OS on eServer p5 Models A Guide to Planning, Implementation, and Operation*, SG24-8001
- *IBM System p Advanced POWER Virtualization Best Practices*, REDP-4194
- *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, SG24-6769
- *Introduction to pSeries Provisioning*, SG24-6389
- *Linux Applications on pSeries*, SG24-6033
- *Managing AIX Server Farms*, SG24-6606
- *NIM: From A to Z in AIX 4.3*, SG24-5524
- *NIM from A to Z in AIX 5L*, SG24-7296
- *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615
- *Integrated Virtualization Manager on IBM System p5*, REDP-4061

Other publications

These publications are also relevant as further information sources:

- The following types of documentation are located through the Internet at the following URL:

  http://www.ibm.com/servers/eserver/pseries/library

  - User guides
  - System management guides
  - Application programmer guides
  - All commands reference volumes
  - Files reference
  - Technical reference volumes used by application programmers
- Detailed documentation about the Advanced POWER Virtualization feature and the Virtual I/O Server

  https://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/home.html
- *AIX 5L V5.3 Partition Load Manager Guide and Reference,* SC23-4883
- *Linux for pSeries installation and administration (SLES 9)*, found at:

  http://www-128.ibm.com/developerworks/linux/library/l-pow-pinstall/
- *Linux virtualization on POWER5: A hands-on setup guide*, found at:

  http://www-128.ibm.com/developerworks/edu/l-dw-linux-pow-virutal.html
- *POWER5 Virtualization: How to set up the SUSE Linux Virtual I/O Server*, found at:

  http://www-128.ibm.com/developerworks/eserver/library/es-susevio/

Online resources

These Web sites and URLs are also relevant as further information sources:

- AIX 5L and Linux on POWER community

  http://www-03.ibm.com/systems/p/community/
- Capacity on Demand

  http://www.ibm.com/systems/p/cod/

- IBM Advanced POWER Virtualization on IBM System p Web page

  http://www.ibm.com/systems/p/apv/
- IBM eServer pSeries and AIX Information Center

  http://publib16.boulder.ibm.com/pseries/index.htm
- IBM System Planning Tool

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/
- IBM Systems Hardware Information Center

  http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp
- IBM Systems Workload Estimator

  http://www-912.ibm.com/supporthome.nsf/document/16533356
- Latest *Multipath Subsystem Device Driver User's Guide*

  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303
- Novell SUSE LINUX Enterprise Server information

  http://www.novell.com/products/server/index.html
- SCSI T10 Technical Committee

  http://www.t10.org
- SDDPCM software download page

  http://www.ibm.com/support/docview.wss?uid=ssg1S4000201
- SDD software download page

  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&dc=D430&uid=ssg1S4000065&loc=en_US&cs=utf-8&lang=en
- Service and productivity tools for Linux on POWER

  http://techsupport.services.ibm.com/server/lopdiags
- Silicon-on-insulator (SOI) technology

  http://www.ibm.com/chips/technology/technologies/soi/
- VIOS supported environment

  http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html
- Virtual I/O Server documentation

  http://techsupport.services.ibm.com/server/vios/documentation/home.html

- Virtual I/O Server home page

  http://techsupport.services.ibm.com/server/vios/home.html
- Virtual I/O Server home page (alternate)

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html
- Virtual I/O Server supported hardware

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html
- Virtual I/O Server Support Page

  http://techsupport.services.ibm.com/server/vios/download/home.html

How to get IBM Redbooks

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

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

D

E

N

O

P

Q

R

S

T

U

V

W

X

Z

Advanced POWER Virtualization on IBM System p5:
Introduction and Configuration

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Advanced POWER Virtualization on IBM System p5:
Introduction and Configuration

Basic and advanced configuration of the Virtual I/O Server and its clients

New features and benefits of virtualization explained

High availability, system administration, and PLM

This IBM Redbook provides an introduction to Advanced POWER Virtualization on IBM System p5 servers.

The Advanced POWER Virtualization feature is a combination of hardware and software that supports and manages the virtual I/O environment on POWER5 and POWER5+ systems. The main technologies are:

- Virtual Ethernet
- Shared Ethernet Adapter
- Virtual SCSI Server
- Micro-Partitioning technology
- Partition Load Manager

In addition, this IBM Redbook is also designed to be used as a reference tool for System Administrators who manage the IBM System p platform.

Though the discussion in this IBM Redbook is focused on System p5 hardware and the AIX 5L operating system, the basic concepts extend themselves to the i5/OS and Linux operating systems, as well as the IBM System i5 platform.

A basic understanding of logical partitioning is required.

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

SG24-7940-02 ISBN 0738489956