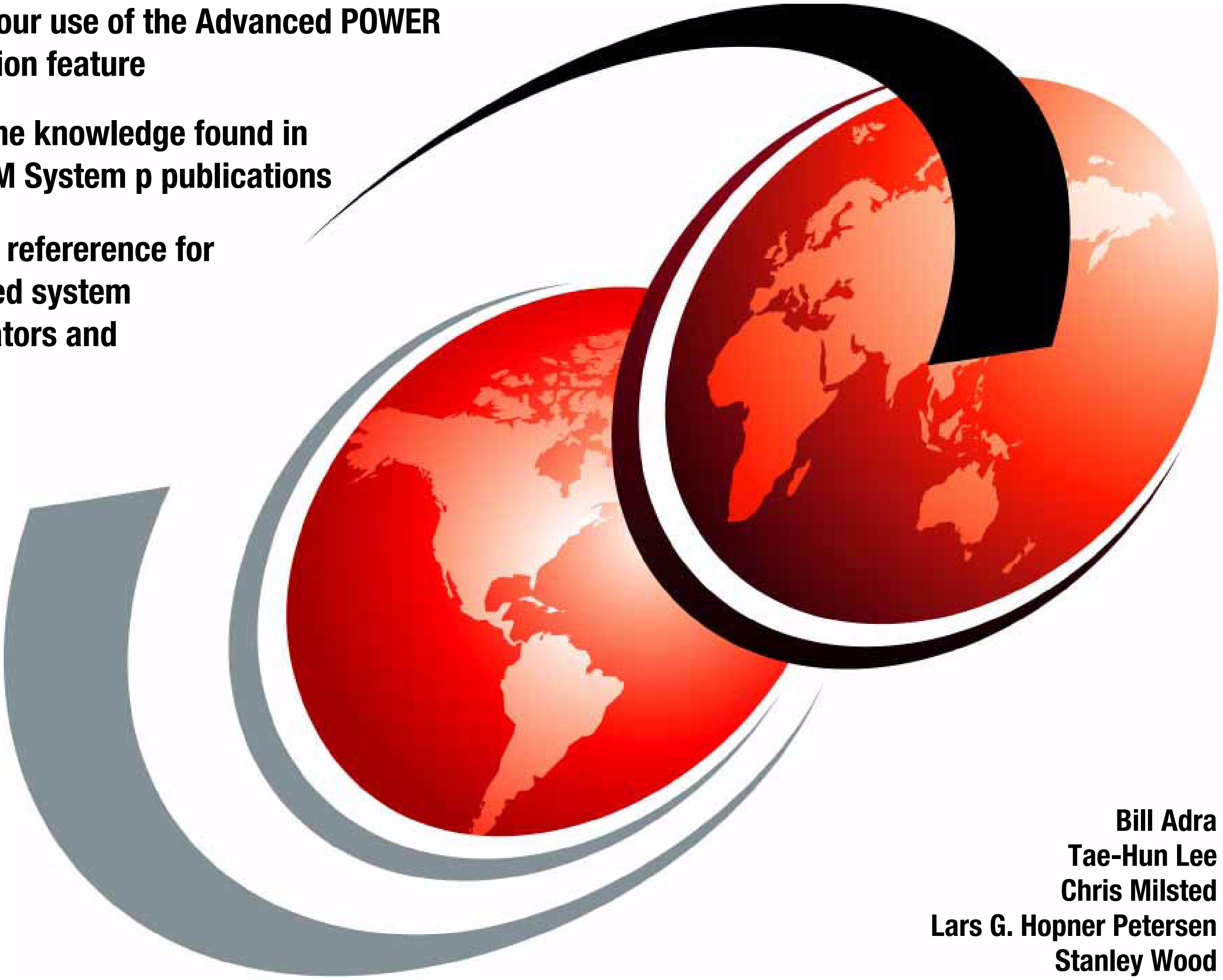
IBM System p Advanced POWER Virtualization Best Practices

- A collection of recommended practices created to enhance your use of the Advanced POWER Virtualization feature
- Builds on the knowledge found in existing IBM System p publications
- A valuable refererence for experienced system administrators and architects



Bill Adra
Tae-Hun Lee
Chris Milsted
Lars G. Hopner Petersen
Stanley Wood

Redpaper

ibm.com/redbooks

International Technical Support Organization

IBM System p Advanced POWER Virtualization Best Practices

October 2006

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

First Edition (October 2006)

This edition applies to:
Version 1, Release 3 of IBM Virtual I/O Server (product number 5765-G34)
Version 5, Release 3, technology level 5 of IBM AIX 5L for POWER (product number 5765-G03)
Version 240, Release 219, Modification 201 of the POWER5 system firmware
Version 5, Release 2, Modification 1, with specific fixes MH00688 and MH00695 of Hardware Management Console

Contents

Notices

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

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

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

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

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

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

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

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

Trademarks

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

AIX 5L™
AIX®
BladeCenter®
DS4000™
DS6000™
DS8000™
eServer™
HACMP™
ibm.com®
IBM®
POWER Hypervisor™
POWER5+™
POWER5™
POWER™
pSeries®
PTX®
Redbooks (logo) ™
Redbooks™
System i™
System p5™
System p™
System Storage™
Tivoli®
TotalStorage®

The following terms are trademarks of other companies:

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® Redpaper provides best practices for planning, installing, maintaining, and operating the functions available using the Advanced POWER™ Virtualization feature 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 IBM POWER5™ and POWER5+™ processor-based systems.

Of those, this publication focuses on the aspects of:

- General best practices
- Administration, backup, and recovery
- Performance and planning
- Storage and networking

This Redpaper can be read from start to finish, but it is meant to be read as a notebook where you access the topics that pertain best to you. This paper begins where *Advanced POWER Virtualization on IBM System p5*, SG24-7940, ends by adding additional samples and scenarios harvested by a select team that works at client and outsourcing sites, running both small and large installations. The experiences contained within are select best practices from real-life experience.

A working understanding of the Advanced POWER Virtualization feature and logical partitioning and IBM AIX® 5L™ is required, as well as a basic understanding of network and VLAN tagging.

The team that wrote this Redpaper

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

Bill Adra is an IT specialist in the IBM Systems and Technology Group in Sydney, Australia with more than seven years of experience in IBM System p™ and IBM TotalStorage® solutions. Bill specializes in System p Advanced POWER Virtualization capabilities. He provides pre-sales technical support and post-sales implementation to IBM Business Partners and clients across Australia and New Zealand. Bill is a Certified Specialist for pSeries® Solutions Sales and an IBM Certified System p AIX 5L Systems Administrator.

Tae-Hun Lee has six years of experience in the System p field and three years of experience in Manufacturing Execute System (MES). He has worked at IBM for six years. His areas of expertise include IBM General Parallel File System, Oracle on the System p server, and virtualization on the System p server.

Chris Milsted is an IT Specialist from the U.K. He has worked for IBM for five years, four of them working with AIX 5L, High-Availability Cluster Multi-Processing (HACMP™), and now Advanced POWER Virtualization. He holds a degree in chemistry from the University of Nottingham. His areas of expertise include System p technology, Linux®, AIX 5L, and HACMP.

Lars G. Hopner Petersen is an IT Architect in Denmark. He has 15 years of experience in the AIX and AIX 5L field. He holds a degree in bachelor of electronics and computer engineering from Engineering College of Copenhagen. His areas of expertise include virtualization, clustering, and UNIX®.

Stanley Wood is a Senior IT Architect in the office of the IBM CIO where he has 11 years of experience architecting and deploying distributed systems and applications. He is the enterprise architect responsible for the virtualization and storage strategies within the IBM infrastructure. Stanley graduated from Texas A&M University with a bachelor of science degree in computer science. His areas of expertise include server and storage virtualization, Network File System (NFS), IBM General Parallel File System, and UNIX systems performance.

The project that created this publication was managed by:

Scott Vetter, PMP
IBM Austin

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

John Banchy, Ron Barker, Indulis Bernsteins, Lupe Brown, Dave Buffaline, Erin C. Burke, Bradford Cobb, Ranadip Das, Azur Gucer, Anil K, Steven Knudson, Bob Kovacs, Kiet H. Lam, David Larson, Timothy Marchini, Bob Minns, Jonathan Van Niewaal, Jorge R. Nogueras, Jim Pafumi, Jim Partridge, Viraf Patel, Amartey S. Pearson, Thomas Prokop, Pacelli Pujals, Vani D. Ramagiri, Rakesh Sharma, Limei Shaw, Jaya Srikrishnan, Tony Steel, John Tesch, Kate Tinklenberg, Vasu Vallabhaneni, Venkat Venkatsubra
IBM U.S. and IBM India

Andrew Goodall, Philip Dotchon, Andy Slim, Nigel Griffiths, Dai Williams
IBM U.K.

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 have the opportunity to team with IBM technical professionals, Business Partners, and clients.

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper 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:

  redbooks@us.ibm.com

- Mail your comments to:

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

1

Introduction

The *Advanced POWER Virtualization on IBM System p5*, SG24-7940, publication provides an introduction to virtualization on the IBM System p5™ platform. This publication builds on that foundation by introducing best practices. The Advanced POWER Virtualization feature is available on the IBM BladeCenter® and IBM System i™ systems in addition to the System p platform, and therefore, many of the topics within apply to those platforms as well.

We recommend that you are familiar with the content of the introduction book and have some practical experience with virtualization before using the material contained here.

Advanced POWER Virtualization is production tested.

The Advanced POWER Virtualization feature from IBM is designed to meet all your enterprise production workload virtualization requirements. The development and test process for virtual I/O is as rigorous as it is for the physical counterparts. We recommend that you deploy it in production, development, test, or any environment you would commonly use separate servers or dedicated partitions.

This paper provides recommendations for the architecture, configuration, and documentation of the Advanced POWER Virtualization feature. In areas where there are multiple configuration options, this publication provides recommendations based on the environment and the experience of the authors.

Although this publication can be read from start to finish, it is written to allow you to select individual topics of interest and jump directly to them. The content is organized into five major headings:

- This chapter, Chapter 1, “Introduction” on page 1, describes introductory topics and best practices.
- Chapter 2, “Administration, backup, and restore” on page 15 describes the administration, backup, and recovery of the Virtual I/O Server.
- Chapter 3, “Networking” on page 51 describes network architecture and configuration within the virtual environment.
- Chapter 4, “Storage” on page 89 describes storage architecture and configuration.
- Chapter 5, “Performance and planning” on page 121 describes performance and CPU management and monitoring.

1.1 Virtual I/O Server enhancements and updates

In this section, we provide information about how to learn more about the Virtual I/O Server (VIOS) and the available service updates.

To learn how to use the VIOS in detail, see the following Web page for additional information:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1kickoff.htm

1.1.1 Tracking the latest virtualization enhancements

Keep up-to-date regarding service and new features

The development of new functions for virtualization is an ongoing process. Therefore, it is best to visit the Web where you can find more information about the new features and other features:

http://techsupport.services.ibm.com/server/vios/documentation/home.html

This section provides a short review of functions noteworthy at the time of writing.

In VIOS Version 1.2, there are several new features for system management and availability:

- The Integrated Virtualization Manager (IVM) can support Virtual I/O Server and virtual I/O client management through a Web browser without needing a Hardware Management Console (HMC). The IVM can be used on IBM System p5 platforms, especially low-end systems, instead of an HMC. The IVM is not available on IBM System p5 Models 570, 575, 590, and 595.
- Virtual optical media can be supported through virtual SCSI between VIOS and virtual I/O clients. Support for virtual optical was first introduced with VIOS Version 1.2. CD-ROM, DVD-RAM, or DVD-ROM can back a virtual optical device.
- With the previous version, network high availability with dual Virtual I/O Servers could only be configured with the Network Interface Backup (NIB) function of AIX 5L. Now you can configure network failover between Virtual I/O Servers using Shared Ethernet Adapter (SEA) Failover.
- Storage pools are a new feature. Although these are similar to volume groups, storage pools can make device management simpler for the novice user.

In VIOS Version 1.3, there are several additional features, including:

- Improvements to monitoring, such as additional **topas** and **viostat** performance metrics, and the enablement of the Performance PTX® agent. (PTX is a licensed program that can be purchased separately.)
- Virtual SCSI and virtual Ethernet performance improvements, command line enhancements, and enablement of additional storage solutions are also included.
- The Integrated Virtualization Manager (IVM) adds leadership function in this release: support for dynamic logical partitioning for memory and processors in managed partitions. Additionally, a number of usability enhancements include support through the browser-based interface for IP configuration of the VIOS.
- IBM System Planning Tool (previously named LVT) enhancements.

  To support your virtualization planning needs, the System Planning Tool (SPT) is available at no charge for download from:

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/

  You can use the System Planning Tool for designing System p and System i partitions. The resulting design, represented by a System Plan, can then be imported onto your

Hardware Management Console (HMC) Version 5.2, where, using the new System Plans feature, you can automate configuration of the partitions designed using the System Planning Tool. The System Plans feature of the HMC also enables the generation of system plans using the **mksysplan** command.

1.1.2 Updating the Virtual I/O Server using fix packs

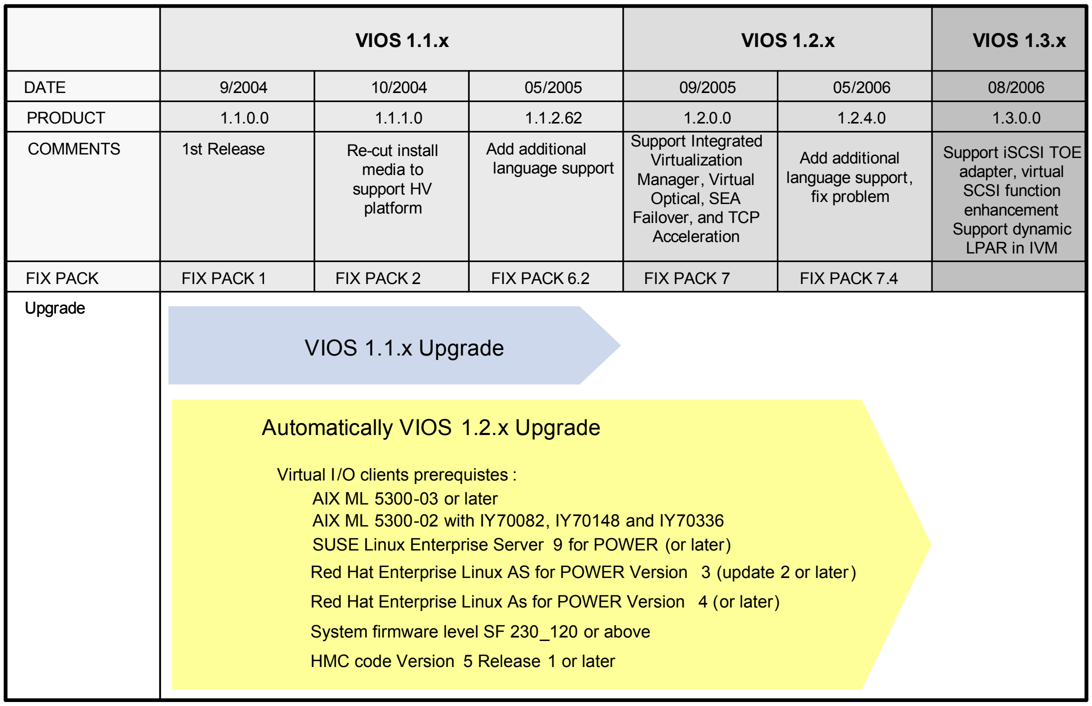
Existing Virtual I/O Server (VIOS) installations can move to the latest VIOS level by applying the latest fix pack.

Fix packs provide a migration path for existing VIOS installations. Applying the latest fix pack will upgrade the VIOS to the latest level. All VIOS fix packs are cumulative and contain all fixes from previous fix packs. The VIOS download page maintains a list of all fixes included with each fix pack.

Service is available to update a Virtual I/O Server to a new version.

Fix packs are typically a general update intended for all VIOS installations. Fix packs can be applied to either HMC-managed or IVM-managed Virtual I/O Servers. All interim fixes applied to the VIOS must be manually removed before applying any fix pack. virtual I/O clients that applied interim fixes to the VIOS should use the procedure described on the VIOS Web site to remove them prior to applying a fix pack.

Figure 1-1 shows the migration from VIOS Version 1.1 to Version 1.2.1.x and beyond using a fix pack.



	VIOS 1.1.x			VIOS 1.2.x		VIOS 1.3.x
DATE	9/2004	10/2004	05/2005	09/2005	05/2006	08/2006
PRODUCT	1.1.0.0	1.1.1.0	1.1.2.62	1.2.0.0	1.2.4.0	1.3.0.0
COMMENTS	1st Release	Re-cut install media to support HV platform	Add additional language support	Support Integrated Virtualization Manager, Virtual Optical, SEA Failover, and TCP Acceleration	Add additional language support, fix problem	Support iSCSI TOE adapter, virtual SCSI function enhancement Support dynamic LPAR in IVM
FIX PACK	FIX PACK 1	FIX PACK 2	FIX PACK 6.2	FIX PACK 7	FIX PACK 7.4	
Upgrade						

VIOS 1.1.x Upgrade

Automatically VIOS 1.2.x Upgrade

Virtual I/O clients prerequistes :
- AIX ML 5300-03 or later
- AIX ML 5300-02 with IY70082, IY70148 and IY70336
- SUSE Linux Enterprise Server 9 for POWER (or later)
- Red Hat Enterprise Linux AS for POWER Version 3 (update 2 or later)
- Red Hat Enterprise Linux As for POWER Version 4 (or later)
- System firmware level SF 230_120 or above
- HMC code Version 5 Release 1 or later

Figure 1-1 Upgrade and migration of VIOS using fix packs

To check the latest release and instructions for the installation of VIOS, visit:

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

Tip: All VIOS fix packs are cumulative and contain all fixes from previous fix packs. Applying the latest VIOS fix pack upgrades an existing VIOS to the latest supported level.

1.2 Working with the Virtual I/O Server

Examine workloads in test before entering production.

When sizing a system, it is a best practice to plan your environment within the design limits of an infrastructure. With that said, the use of theoretical maximum values quoted as a foundation for your production system design may cause a trade off between capacity and expected performance. It is always a best practice to examine your most critical workloads in a test configuration before implementing them into production.

1.2.1 Virtual Ethernet

You can define up to 256 virtual Ethernet adapters per Virtual I/O Server. The number of virtual I/O clients that can be connected to a virtual Ethernet adapter has a design value that is limited to the available memory. We recommend that you, before you reach the maximum numbers of client connections, test your configuration. The following list gives some of the architectural limitations of virtual Ethernet:

- Each virtual Ethernet adapter is capable of being associated with up to 21 virtual local area networks (VLANs) (20 VIDs and 1 PVID).
- A system can support up to 4096 different VLANs, as defined in the IEEE 802.1Q standard.
- Each Shared Ethernet Adapter (SEA) can have up to 16 virtual Ethernet adapters, each with up to 21 VLANs (20 VID and 1 PVID) associated with it. With SEA, these VLANs can share a single physical network adapter.

1.2.2 Virtual SCSI

You can define up to 256 virtual SCSI adapters per Virtual I/O Server.

Split the queue depth between the number of devices you have.

In AIX 5L Version 5.3 (with the latest service including PTF IY81715), you can have a queue depth per SCSI device on the client from one to 256. This value determines how many requests the disk controller will queue to the virtual SCSI client driver at any one time. The value can be modified using the command:

```
chdev -l hdiskN -a'queue_depth= value'
```

Where `hdiskN` is the name of a physical volume and *`value`* is a number from 1 to 256. The current value of the `queue_depth` attribute can be viewed using the command:

```
lsattr -El hdiskN
```

Where `hdiskN` is the name of a physical volume.

For a detailed explanation about queue depth, see 4.8, “SCSI queue depth” on page 118.

1.3 Introduction to Virtual I/O Server resilience

The Virtual I/O Server is part of the Advanced POWER Virtualization hardware feature, which enables sharing of physical resources between logical partitions including virtual SCSI and virtual networking.

Dual Virtual I/O Servers configurations can provide concurrent software maintenance.

Clustering software, such as HACMP or Veritas Cluster Server are designed to support system node failover. Configuring a highly available path and device resiliency for virtualized environment does not require the implementation of HACMP on the Virtual I/O Server itself. Two or more Virtual I/O Servers and redundant devices can provide improved software maintenance and hardware replacement strategies. Multiple Virtual I/O Servers must be managed by the HMC.

The implementation of HACMP is supported using virtual I/O resources on virtual I/O clients. The installation of HACMP is a component of the client operating system.

Ways to provide additional Virtual I/O Server resilience

To further harden the availability of a Virtual I/O Server, use a combination of the following items:

- Redundant physical hardware
- Network Interface Backup Ethernet configuration
- Shared Ethernet Adapter (SEA) Failover configuration
- Storage Logical Volume Manager (LVM) mirroring and RAID configurations
- Storage area network (SAN) multipath I/O (MPIO)
- RAID protected storage (RAID provided either by the storage subsystem or by a RAID adapter)
- Effective and detailed preinstallation planning

Another method to enhance seviceability is to use hot-pluggable network adapters for the Virtual I/O Server instead of the built-in integrated network adapters. It is easier to replace the PCI-slot adapters during an upgrade or service strategy.

The following section describes the various options available with considerations when implementing a Virtual I/O environment.

1.3.1 Single Virtual I/O Server configurations

The Virtual I/O Server is a robust, resilient, and secure environment designed to support the most demanding production workloads.

The Virtual I/O Server is running a only few extremely reliable device drivers, tested and packaged with the server. Instead of using local physical device drivers, the client partition uses the virtual resource device drivers to communicate with the Virtual I/O Server, which does the physical I/O. Other than the virtual (Virtual I/O Server) device drivers and the physical resource device drivers, there are a limited number of processes running on the Virtual I/O Server and, therefore, a greatly reduced risk of server-side software problems.

We recommend the use of two Virtual I/O Servers as part of a scheduled maintenance policy, where concurrent online software updates are required, and to extend the number of configurations possible. It also enables you to have a fail-over network connection to a different switch when using link aggregation. When these functions are not required, client configurations might be ideally suited to a single Virtual I/O Server.

See 1.3.2, “Dual Virtual I/O Servers configurations” on page 9 for a discussion of the implementation of dual Virtual I/O Servers.

The following topics describe the recommended steps and procedures when implementing a virtualized environment using a single Virtual I/O Server.

Link aggregation can improve network availability and throughput.

Shared Ethernet Adapter availability

The first resource to configure on the Virtual I/O Server is the Shared Ethernet Adapter. The Shared Ethernet Adapter needs at least one physical port but can use multiple physical Ethernet ports as an aggregated link to the physical network. The use of network adapters can provide either additional bandwidth over and above a single network connection or provide increased redundancy, or both.

When implementing an aggregated link, all primary ports must be connected to the same switch and the switch ports used for aggregation must be configured for either 802.3ad link aggregation or Cisco EtherChannel. You must create the link aggregation switch settings before creating the Shared Ethernet Adapter.

The main benefit of an aggregated link is that the network bandwidth of all of its adapters appears as a single link. If an adapter becomes unavailable, the packets are automatically sent on the next available adapter without disruption to existing connections. However, link aggregation is not a complete high-availability networking solution because all the aggregated links must connect to the same switch. This requirement can be overcome with the use of a backup adapter. You can add a single additional link to the link aggregation that is connected to a different Ethernet switch with the same VLAN. This single link will only be used as a backup. Figure 1-2 shows a recommended Shared Ethernet Adapter configuration for increased availability using a single Virtual I/O Server and an aggregated link.

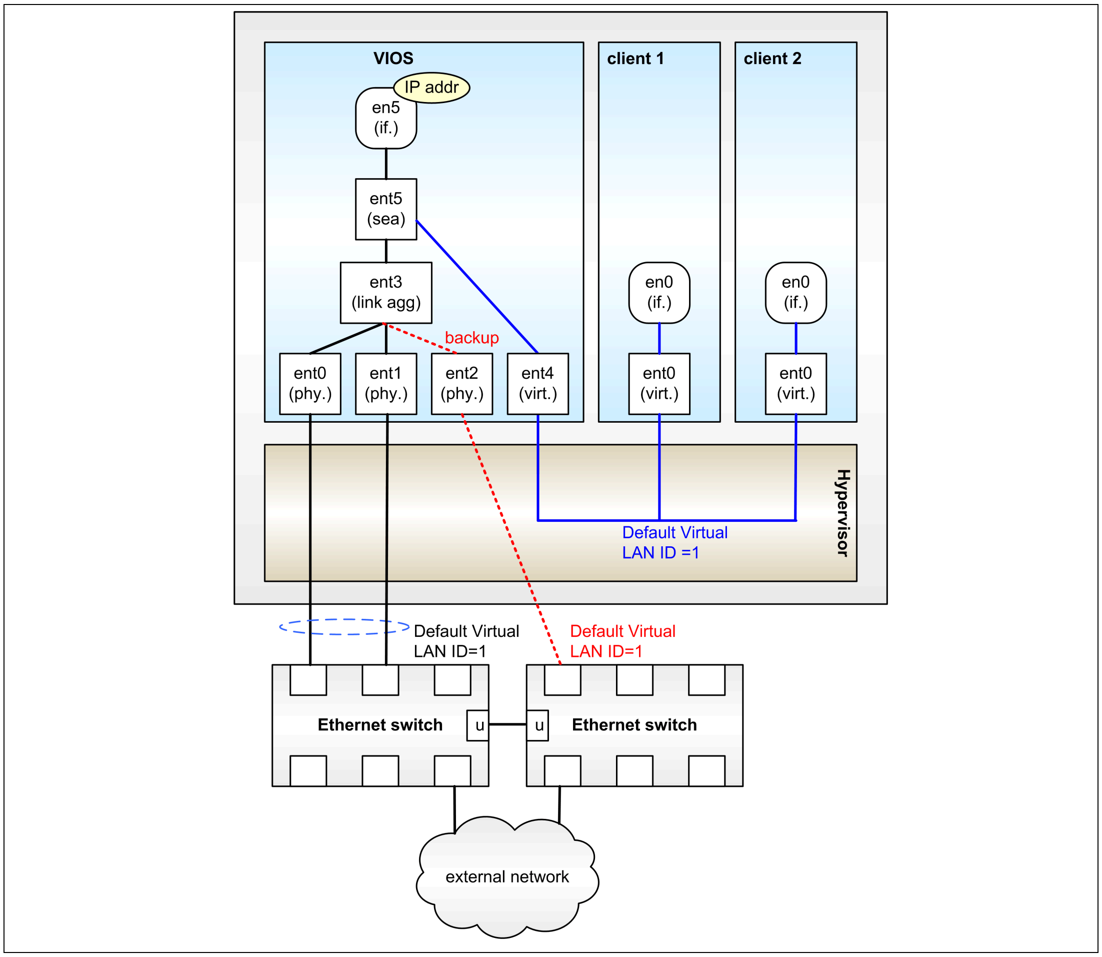


Figure 1-2 Recommended single Virtual I/O Server network configuration

Redundant boot disks provide availability during disk replacement.

Boot disk redundancy in the Virtual I/O Server

The Virtual I/O Server software resides in a logical partition. As such, the Virtual I/O Server requires its own resources including processor, memory, and dedicated disk. The Virtual I/O Server software is installed on a dedicated disk that is the Virtual I/O Server's root volume group (rootvg). When booting the Virtual I/O Server from locally attached SCSI disks, the root volume group (for example, hdisk0) should be extended to include a second SCSI disk (hdisk1) and the data mirrored on that disk. When performing this operation, the Virtual I/O Server is rebooted. Therefore, we recommend that you perform this at installation time, or when all virtual I/O clients using resources through this Virtual I/O Server do not require the defined virtual resources. Use the following commands:

```
$ extendvg –f rootvg hdisk1
$ mirrorios hdisk1
This command causes a reboot. Continue [y|n]?
```

After the mirroring completes, all the remaining physical storage on the Virtual I/O Server can then be exported to the client partitions and the clients started.

In addition to booting from locally attached SCSI disks, the Virtual I/O Server can be attached to SAN storage. You can install the Virtual I/O Servers root volume group on an assigned SAN disk. We recommend that you boot the Virtual I/O Server from SAN storage when the storage subsystem supports multipathing during boot time to provide additional paths. Check with your storage vendor to determine considerations regarding booting from SAN.

It is a common practice not to mix user data with system data; therefore, we also do not recommend placing the boot disks of clients on the same disk as the boot disk of a Virtual I/O Server.

Configure SAN storage for virtual I/O clients

The Virtual I/O Server supports SAN-backed storage devices to be used for the virtual I/O clients. Figure 1-3 shows the recommended configuration for virtual I/O clients using SAN-based backing devices through a single Virtual I/O Server.

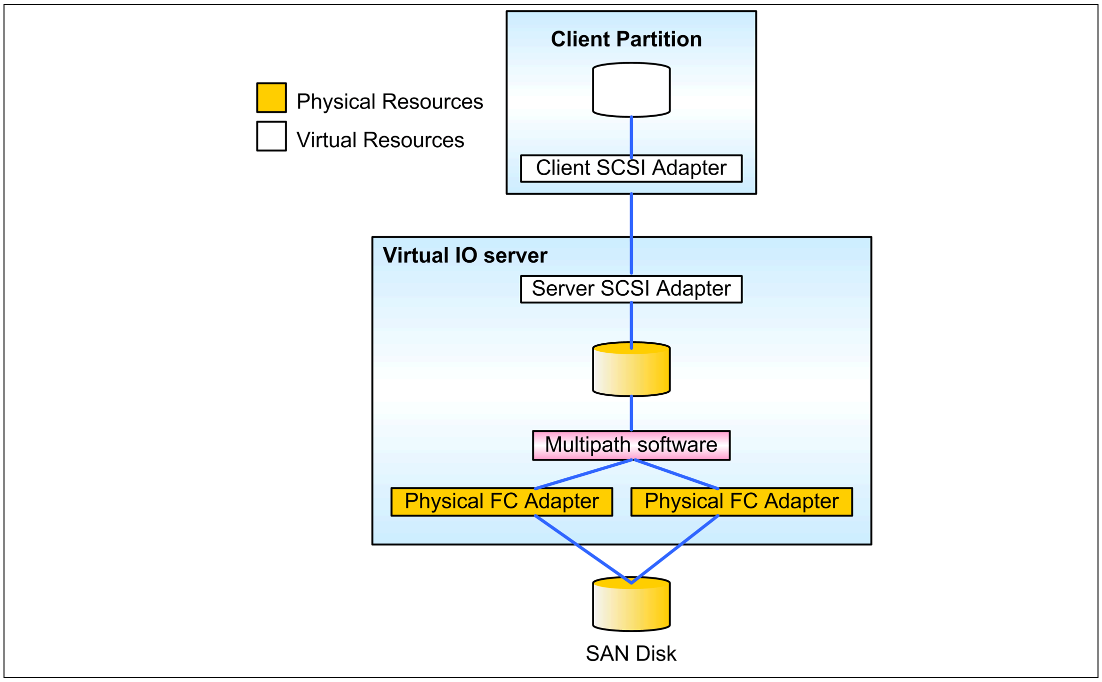


Figure 1-3 Single Virtual I/O Server connected to SAN storage

MPIO, or multipathing, provides multiple paths to critical data.

Figure 1-3 shows a single Virtual I/O Server configuration that includes two fibre channel adapters connected to the SAN storage. The storage vendor multipathing software is installed on the Virtual I/O Server using the **oem_setup_env** command. The multipathing software can provide load balancing and fail-over capabilities in the event that an adapter becomes unavailable. When presenting a SAN LUN to a client partition, we recommend that the entire LUN is passed through rather than configuring a Virtual I/O Server storage pool or volume group. This is because logical volumes or storage pools are not supported in a dual Virtual I/O Servers configuration. The disk size required for the virtual I/O client partition should also be configured using the SAN storage subsystem.

You can obtain the effect of dual VIOS lvvscsi disks (for example, a large LUN presented from a storage subsystem that is subdivided) by using dual Virtual I/O Servers and SAN Volume Controller (SVC).

Disk mirroring for virtual I/O clients

Disk mirroring protects client data.

The Virtual I/O Server supports SCSI-backed storage devices to be used for the virtual I/O clients. Figure 1-4 shows the recommended configuration for client partitions using SCSI-based backing devices through a single Virtual I/O Server.

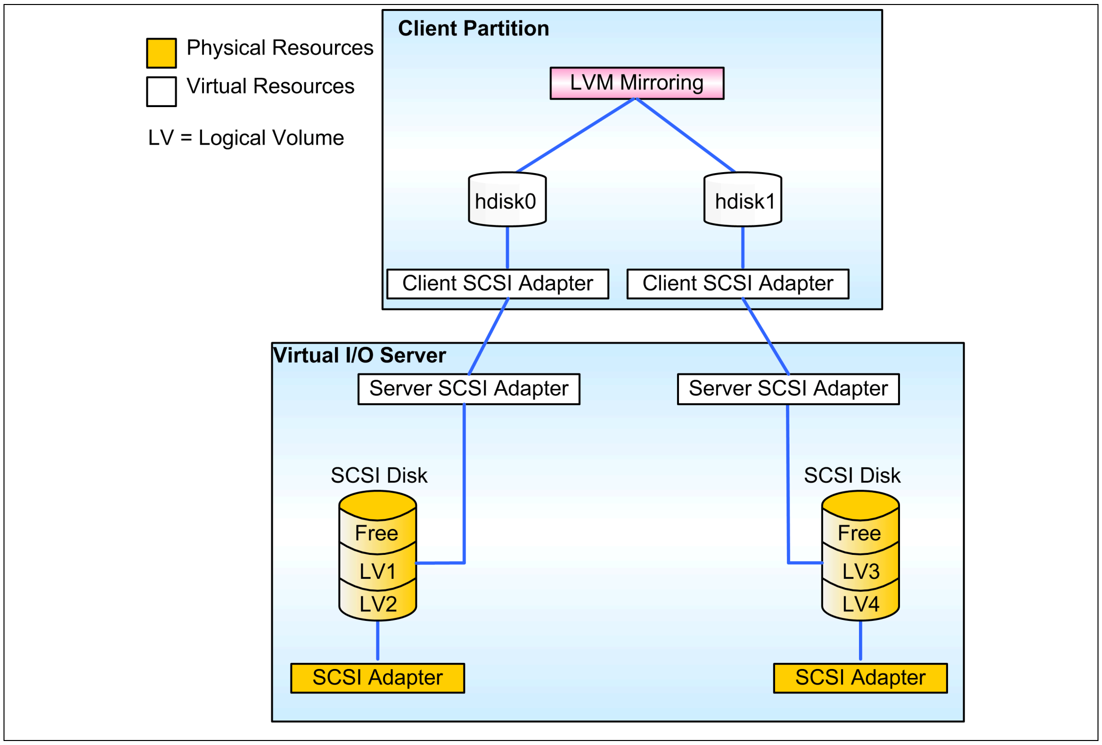


Figure 1-4 SIngle Virtual I/O Server using LVM mirroring at the client partition

Figure 1-4 shows a single Virtual I/O Server configuration that includes two SCSI adapters, each with a set of SCSI disk drives attached. When presenting a disk to a virtual I/O client using SCSI disks, we recommend that either two physical disks or two logical volumes each from a different disk (with each disk on a separate controller) are passed to the virtual I/O client partition. The virtual I/O client partition will then detect two virtual SCSI disks that are then mirrored at the virtual I/O client using client operating system LVM mirroring.

Important: Logical volumes used as virtual disks can be as large as 1 TB in size. In addition, logical volumes on the Virtual I/O Server that are going to be used as virtual disks cannot be mirrored, striped, or have bad block relocation enabled.

Logical volumes exported to clients as virtual disks should not span more than one physical volume. If more space is required, export a second logical volume on another disk as a separate virtual disk.

1.3.2 Dual Virtual I/O Servers configurations

With multiple virtual I/O client partitions dependent on the Virtual I/O Server for resources, you ban implement dual Virtual I/O Servers and duplicate paths and devices to provide additional system service and configuration options.

Dual Virtual I/O Servers offer improved serviceability.

Fundamentally, the primary reasons for recommending a dual Virtual I/O Servers configuration include:

- Future hardware expansion and new function
- Unscheduled outages due to human intervention
- Unscheduled outages due to physical device failure or natural events
- Scheduled outages required for Virtual I/O Server maintenance
- Isolation of network and storage workload to provide increased virtual I/O client partition performance
- Multiple multipath codes such as MPIO and device redundancy

The following sections describe the general recommendations and best practices when implementing a dual Virtual I/O Servers configuration.

Enhanced network availability

The two common methods available to provide virtual I/O client partition network redundancy in dual Virtual I/O Servers configurations are:

- Network Interface Backup (NIB)
- Shared Ethernet Adapter (SEA) Failover

The following sections introduce both methods. Section 3.8.2, “Dual Virtual I/O Servers enhanced availability options” on page 75 discusses the comparison of both the Network Interface Backup (NIB) and Shared Ethernet Adapter (SEA) Failover options.

Network Interface Backup

Figure 1-5 on page 10 shows a highly-available network configuration using dual Virtual I/O Servers. This configuration uses virtual Ethernet adapters created using default virtual LAN IDs with the physical Ethernet switch using untagged ports only. In this example, VIOS 1 has a Shared Ethernet Adapter that provides external connectivity to client partition 1 through the virtual Ethernet adapter using virtual LAN ID 2. VIOS 2 also has a Shared Ethernet Adapter that provides external connectivity to the client partition through the virtual Ethernet adapter using virtual LAN ID 3. Client partition 2 has a similar set up except that the virtual Ethernet adapter using virtual LAN ID 2 is the primary and virtual LAN ID 3 is the backup. This enables client partition 2 to get its primary connectivity through VIOS 1 and backup connectivity through VIOS 2.

Client partition 1 has the virtual Ethernet adapters configured using Network Interface Backup such that the virtual LAN ID 3 network is the primary and virtual LAN ID 2 network is the backup, with the IP address of the default gateway to be used for heartbeats. This enables client partition 1 to get its primary connectivity through VIOS 2 and backup connectivity through VIOS 1.

If the primary Virtual I/O Server for an adapter becomes unavailable, the Network Interface Backup mechanism will detect this because the path to the gateway will be broken. The Network Interface Backup setup will fail over to the backup adapter that has connectivity through the backup Virtual I/O Server.

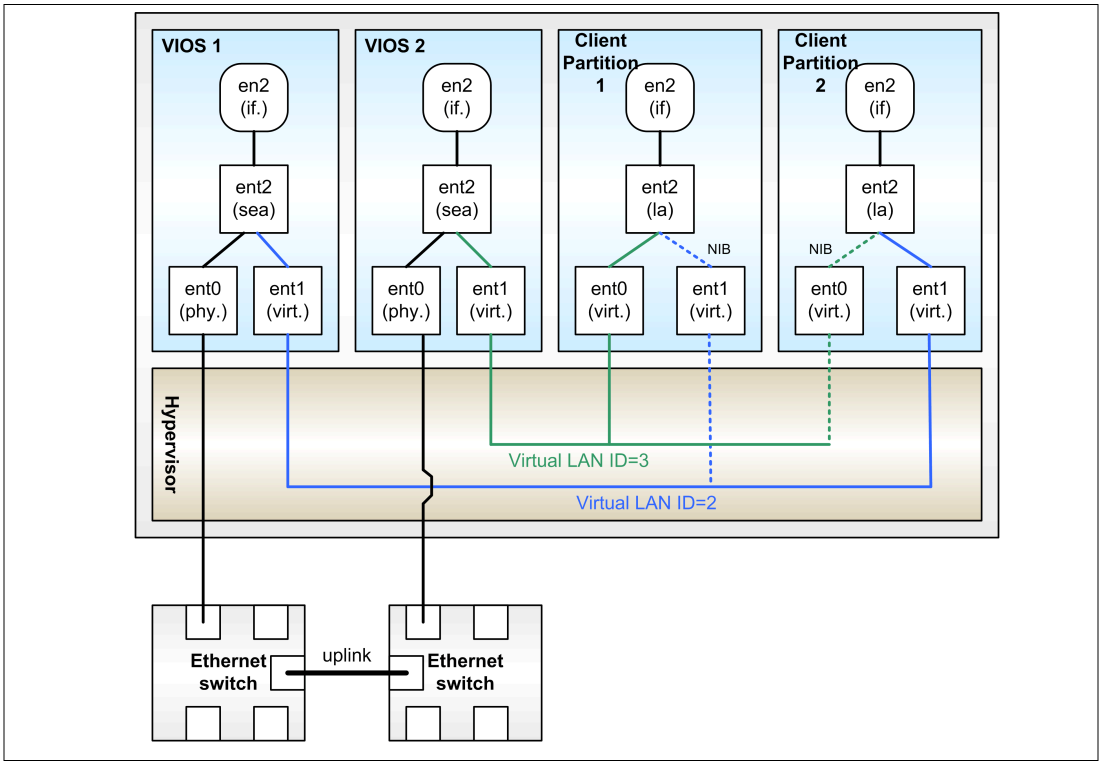


Figure 1-5 Network Interface Backup using dual Virtual I/O Servers

Shared Ethernet Adapter Failover

Shared Ethernet Adapter (SEA) Failover is implemented on the Virtual I/O Server using a bridging (layer-2) approach to access external networks. SEA Failover supports IEEE 802.1Q VLAN-tagging, unlike Network Interface Backup.

With SEA Failover, two Virtual I/O Servers have the bridging function of the Shared Ethernet Adapter to automatically fail over if one Virtual I/O Server is unavailable or the Shared Ethernet Adapter is unable to access the external network through its physical Ethernet adapter. A manual failover can also be triggered.

As shown in Figure 1-6 on page 11, both Virtual I/O Servers attach to the same virtual and physical Ethernet networks and VLANs, and both virtual Ethernet adapters of both Shared Ethernet Adapters have the access external network flag enabled. An additional virtual Ethernet connection must 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. This VLAN serves as a channel for the exchange of keep-alive or heartbeat messages between the two Virtual I/O Servers that 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 any other purpose.

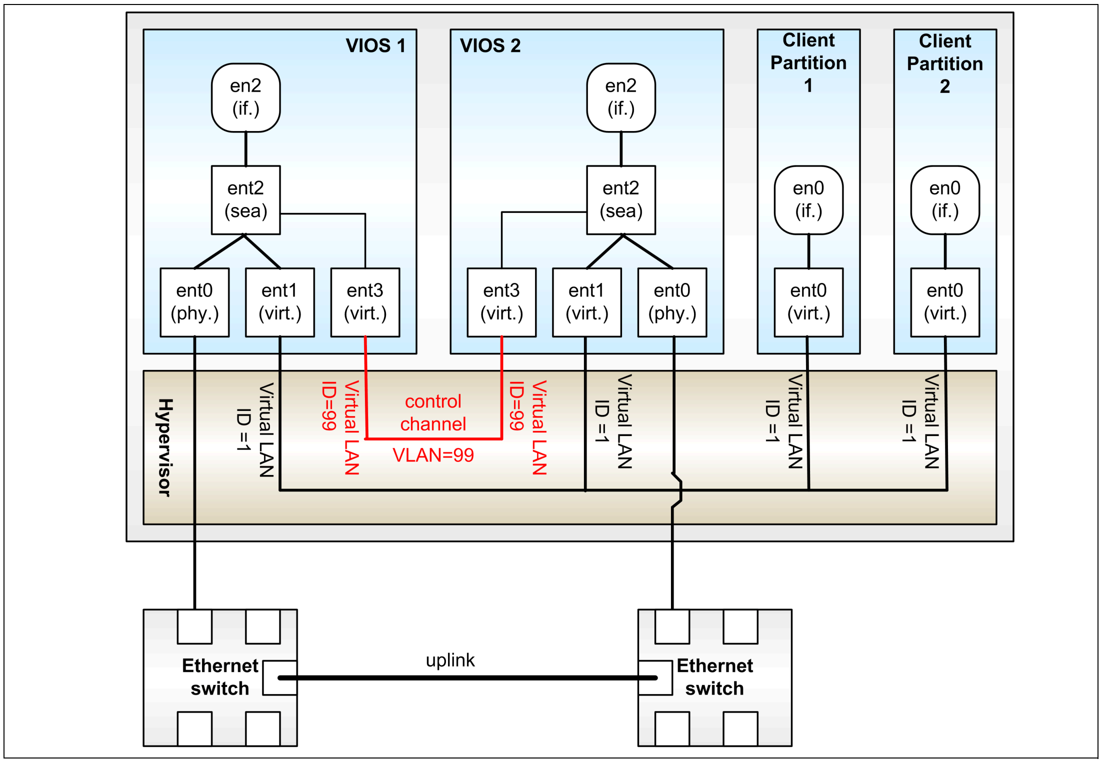


Figure 1-6 SEA Failover using dual Virtual I/O Servers

In addition, the Shared Ethernet Adapter in each Virtual I/O Server must be configured with different priority values. The priority value defines which of the two Shared Ethernet Adapters will be the primary (active) and which will be the backup (standby). The lower the priority value, the higher the priority (for example, priority=1 is the highest priority).

You can also configure the Shared Ethernet Adapter with an IP address that it will periodically ping to confirm that network connectivity is available. This is similar to the IP address to ping that can be configured with Network Interface Backup (NIB). If you use NIB, you have to configure the reachability ping on every client compared to doing it once on the SEA.

See 3.9, “Creating Shared Ethernet Adapters to support SEA Failover” on page 76 for more details.

It is possible that, during an SEA Failover, the network drops up to 15-30 packets while the network reroutes the traffic.

Book disk redundancy for dual Virtual I/O Servers configurations

Implement the mirroring of a Virtual I/O Servers root volume group as discussed in “Boot disk redundancy in the Virtual I/O Server” on page 6.

Multipathing

In the case of physical disks that are accessed by virtual I/O client partitions using SAN-attached storage on a dual Virtual I/O Servers configuration, multipathing can be used from the virtual I/O client partition to provide redundant paths to the disk. Using multipathing from the virtual I/O client partition provides the most highly available and automated availability solution. Figure 1-7 on page 12 shows this in more detail.

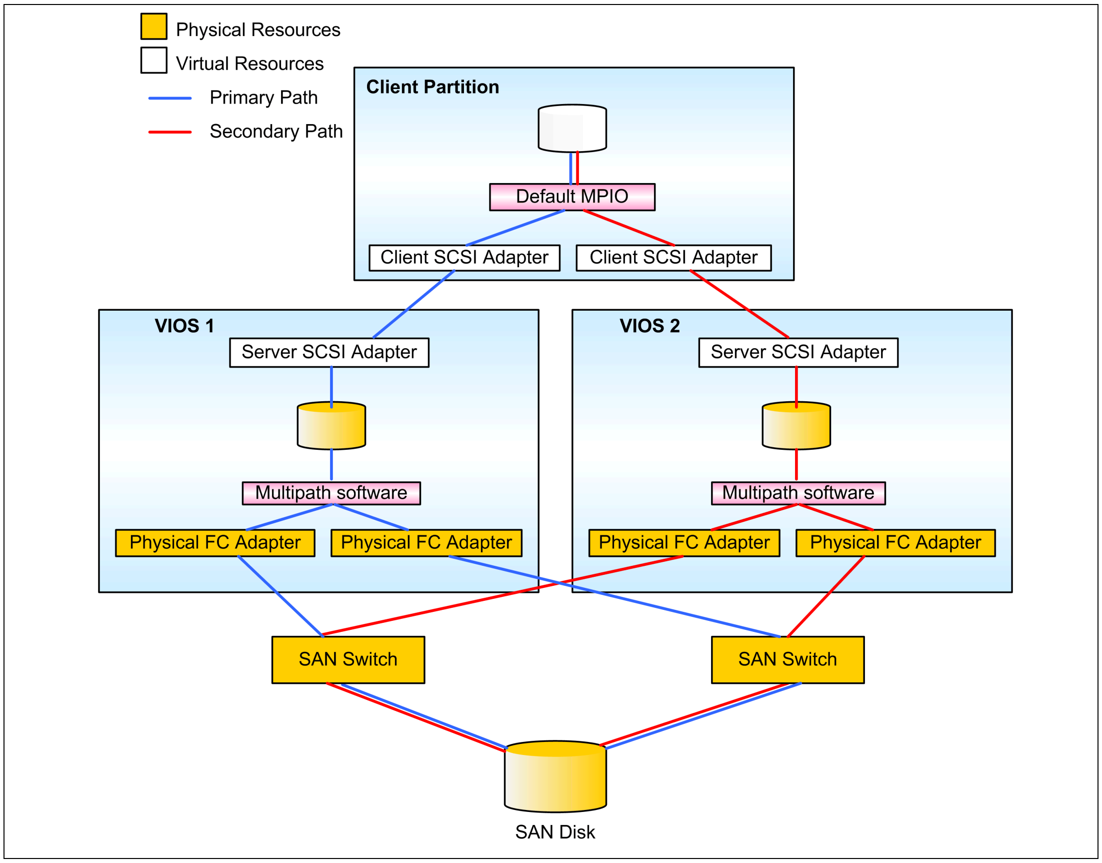


Figure 1-7 Dual Virtual I/O Servers connected to SAN storage using MPIO

The MPIO support through virtual SCSI between the virtual I/O client partition and the dual Virtual I/O Servers supports failover mode. A virtual I/O client partition LUN will use a primary path to VIOS 1 and fail over to the secondary path to use VIOS 2. Only one path is used at a given time even though both paths might be enabled. For more detailed information about configuring MPIO using dual Virtual I/O Servers, refer to 4.7, “Planning and deploying MPIO” on page 113.

Using this configuration, you can shut down a Virtual I/O Server for scheduled maintenance and all active clients will automatically access their disks through the backup Virtual I/O Server. When the Virtual I/O Server comes back online, no action on the virtual I/O clients is needed.

LVM mirroring configuration

LVM mirroring goes beyond disk failure protection.

LVM mirroring provides additional adapter, storage server, and Virtual I/O Server redundancy over using traditional RAID. It protects against more than just a disk failure.

With the use of logical volume mirroring on the virtual I/O client partition, each Virtual I/O Server can present a virtual SCSI device that is physically connected to a different disk and then use AIX 5L LVM mirroring on the client partition to provide a higher level of availability. Client volume group mirroring is also required when using a logical volume from the Virtual I/O Server as a virtual SCSI device on the virtual I/O client partition. In this case, the virtual SCSI devices are associated with different SCSI disks, each controlled by one of the two Virtual I/O Servers.

Figure 1-8 displays an advanced configuration using LVM mirroring in the client partition. Dual Virtual I/O Servers host the disks for the client partition. In this example, the client partition uses LVM mirroring to access two SCSI disks.

Using this configuration, if a Virtual I/O Server is shut down for maintenance, a mirror will be temporarily lost. When the Virtual I/O Server is restored, a re-synchronize of the mirrors on the virtual I/O clients should be performed. The **varyonvg** command on the client performs this.

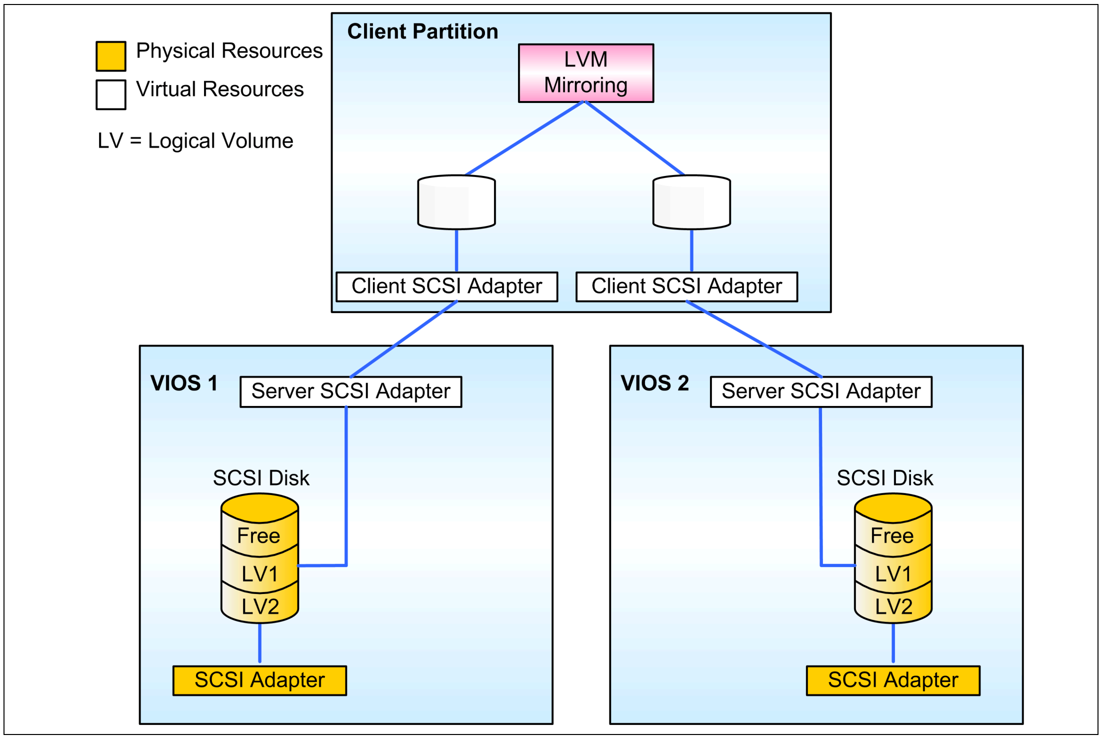


Figure 1-8 Dual Virtual I/O Servers connected to SCSI storage using LVM mirroring

1.3.3 Scheduling maintenance

A scheduled maintenance window is a best practice.

Virtualization enables clients to combine different workloads on the same system. This provides many opportunities for cost savings and efficiency. When many different workloads and client sets are running on the same system, it also presents a new set of considerations for scheduling maintenance.

Schedule maintenance windows as much as possible throughout the year to minimize the impact to operations. Twice yearly maintenance windows have been found to be a good fit for many organizations, especially when scheduled to avoid peak demands that often occur at the end of financial periods.

This publication addresses methods of configuring virtual I/O to enable concurrent maintenance of many aspects of the system, as in 2.5, “Virtual I/O Server maintenance” on page 41, and to quickly shift partitions from one system to another, found in 4.6.4, “Moving an LPAR” on page 112. However, there are some aspects of system maintenance, such as firmware updates, that might require scheduled down time.

Administrators of large shared systems should schedule periodic maintenance windows. Schedule these windows regularly, but only take them as needed. If there is no maintenence to be performed, the system can remain available during the window. If maintenance is required, only perform it during a scheduled window. This helps set expectations across all user bases and enables them to plan their schedules and operations around maintenance windows in advance.

Time management is needed in the scheduling of maintenance windows. If too few are scheduled, situations are more likely to arise when maintenance must be performed outside of a window. If too many are scheduled, clients will come to expect that maintenance windows will not be used and attempt to work through them. Both of these situations eventually make the scheduled windows ineffective.

2

Administration, backup, and restore

A comprehensive backup strategy is a best practice.

This chapter describes best practices for general administration topics of the virtual I/O environment, with a focus on backing up and restoring the Virtual I/O Server. We also address scheduling jobs, sequencing the startup and shutdown of the server, and performing maintenance on the Virtual I/O Server.

Because virtual I/O clients depend on the Virtual I/O Server for services, it is critical that the entire virtual I/O environment be backed up and that system restore and startup procedures include the Virtual I/O Server.

2.1 Backing up and restoring the Virtual I/O Server

The Virtual I/O Server, like all other servers within an IT environment, needs to be backed up as part of an enterprise's data recovery program. This section sets out a strategy for doing this and also describes how to coordinate the backups of the Virtual I/O Server with an existing or new backup strategy that are part of your IBM AIX 5L or Linux operating system environments.

In this section, we describe a complete solution that could be used to restore the Virtual I/O Server to another server, independent of machine type or model number. If you want to perform a backup of just your Virtual I/O Sever, only a subset of this section is required.

2.1.1 When to back up the Virtual I/O Server

A complete disaster recovery strategy for the Virtual I/O Server should include backing up the following components such that you can recover the virtual devices and their physical backing devices. When to back up the Virtual I/O Server, if necessary, will be followed by our server backup strategy where we rebuild the AIX 5L or Linux operating system-based logical partitions.

Backup is required if any of these items are changed.

The following components make up a virtualized environment; a change to any one of these requires a new backup:

- External device configuration, for example, SAN and storage subsystem configuration.
- Memory, CPU, virtual and physical devices.
- The Virtual I/O Server operating system.
- User-defined virtual devices that couple the virtual and physical environments. This can be considered virtual device mappings, or metadata.

You might notice that there is no mention of the operating system, installed applications, or application data of the clients listed. This is because the Virtual I/O Server manages only the devices and the linking of these devices along with the Virtual I/O operating system itself. The AIX 5L or Linux operating system-based clients of the Virtual I/O Server should have a backup strategy independently defined as part of your existing server backup strategy.

For example, if you have an AIX 5L server made up of virtual disk and virtual network, you would still have a **mksysb**, **savevg**, or equivalent strategy in place to back up the system. This backup strategy can rely on the virtual infrastructure. For example, backing up to an IBM Tivoli Storage Manager server over a virtual network interface through a physical Shared Ethernet Adapter.

If you just want to back up the Virtual I/O Server, the third point will be the one you are most interested in, the Virtual I/O Server operating system.

2.1.2 Virtual I/O Server backup strategy

In this section, we define how and when to perform the backup operations.

Determine disaster recovery requirements.

External device configuration

In the event that a natural or man-made disaster destroys a complete site, planning for that occurrence should be included into the end-to-end backup strategy. This is probably part of your disaster recovery (DR) strategy, but consider it in the complete backup strategy. The backup strategy for this depends on the hardware specifics of the storage, networking equipment, and SAN devices to name but a few. Examples of the type of information you will need to record include the network virtual local area network (VLAN) or logical unit number (LUN) information from a storage subsystem.

This information is beyond the scope of this document, but we mention it here to make you aware that a complete DR solution for a physical or virtual server environment will have a dependency on this information. The method to collect and record the information will depend not only on the vendor and model of the infrastructure systems at the primary site, but also what is present at the DR site.

Resources defined on the Hardware Management Console

Back up the HMC configuration.

The definition of the Virtual I/O Server logical partition on the HMC includes, for example, how much CPU and memory and what physical adapters are to be used. In addition to this, you have the virtual device configuration (for example, virtual Ethernet adapters and what virtual LAN ID to which they belong) that needs to be captured. The backup and restore of this data is beyond the scope of this document. For more information, see the IBM Systems Hardware Information Center under the “Backing up partition profile data” topic:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphai/backupprofdata.htm

Note that, especially if you are planning for disaster recovery, you might have to rebuild selected HMC profiles from scratch on new hardware. In this case, it is important to have detailed documentation of the configuration, such as how many Ethernet cards are needed. Using the system plans and the viewer can help record such information but you should check that this is appropriate and that it records all the information needed in every case.

The Virtual I/O Server operating system

Back up the VIOS operating system data.

The Virtual I/O Server operating system consists of the base code, fix packs, custom device drivers to support disk subsystems, and user-defined customization. An example of user-defined customization can be as simple as the changing of the Message of the Day or the security settings.

These settings, after an initial set up, will probably not change apart from the application of fix packs, so a sensible backup strategy for the Virtual I/O Server is after fix packs have been applied or configuration changes made. Although we discuss the user-defined virtual devices in the next section, it is worth noting that the backup of the Virtual I/O Server will capture some of this data. With this fact in mind, you can define the schedule for the Virtual I/O operating system backups to occur more frequently to cover both the Virtual I/O operating system and the user-defined devices in one single step.

With the release of Virtual I/O Server Version 1.3, you can schedule jobs through the **crontab** command (2.3, “Scheduling jobs on the Virtual I/O Server” on page 39). You can schedule the following backup steps to take place at regular intervals using this command.

The **backupios** command performs a backup of the Virtual I/O Server to a tape device, an optical device, or a file system (local or a remotely mounted Network File System, NFS, one).

Note: Consider the following information:

- Virtual device mappings (that is, customized metadata) is backed up by default. Nothing special needs to happen.
- Client data is not backed up.

Note: The following backups were taken on a system with a storage pool storage01 and volume group volgrp01 defined to illustrate that the **savevgstruct** command is called for all online volume groups by default. The output of the **lssp** command on the system is shown:

```
$ lssp
Pool            Size(mb)   Free(mb)   Alloc Size(mb)  BDs
rootvg          139776     107136     128             0
storage01       69888      69760      64              1
volgrp01        69888      69760      64              1
```

Backing up the Virtual I/O Server operating system to CD or DVD

The command to back up the Virtual I/O Server to a DVD-RAM is similar to the command shown in Example 2-1 (depending on whether your DVD drive is at /dev/cd0).

Example 2-1 Backing up the Virtual I/O Server to DVD-RAM

```
$ backupios -cd /dev/cd0 -udf -accept

Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
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 44933 files.........
44933 of 44933 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...
```

If you want to write the backup to multiple CDs instead (such as when the backup is too large to fit on a single CD), use the command shown in Example 2-2.

Example 2-2 Backing up the Virtual I/O Server to CD

```
$ backupios -cd /dev/cd0 -accept

Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
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 44941 files...........
44941 of 44941 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Creating temporary file system: /mkcd/cd_fs...
Populating the CD or DVD file system...
Copying backup to the CD or DVD file system...
.
Building chrp boot image...
Creating temporary file system: /mkcd/cd_images...
Creating Rock Ridge format image: /mkcd/cd_images/cd_image_315528
Running mkisofs ...
..
mkrr_fs was successful.

Making the CD or DVD image bootable...
Writing the CD or DVD image to device: /dev/cd0...
Running cdrecord ...
Cdrecord 1.9 (rs6000-ibm-aix) Copyright (C) 1995-2000 Jörg Schilling
scsidev: '0,0'
scsibus: 0 target: 0 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : WBUS16 SYNC
Vendor_info    : 'IBM     '
Identifikation : 'RMBO0020501     '
Revision       : 'H106'
Device seems to be: Generic mmc2 DVD.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 31 seconds.
............................................
.............................................
.....................Track 01: Total bytes read/written: 673855488/673855488
(329031 sectors).
.......
burn_cd was successful.

The backup will require an additional CD or DVD.
Remove the current writable CD or DVD (volume 1) from the
CD or DVD device and place a new writable CD or DVD (volume 2),
in device /dev/cd0.
Press the <enter> key when ready...

Copying the remainder of the backup to the CD or DVD file system...
```

```
Creating Rock Ridge format image: /mkcd/cd_images/cd_image_315528
Running mkisofs ...
.
mkrr_fs was successful.

Writing the CD or DVD image to device: /dev/cd0...
Running cdrecord ...
Cdrecord 1.9 (rs6000-ibm-aix) Copyright (C) 1995-2000 Jörg Schilling
scsidev: '0,0'
scsibus: 0 target: 0 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : WBUS16 SYNC
Vendor_info    : 'IBM     '
Identifikation : 'RMBO0020501     '
Revision       : 'H106'
Device seems to be: Generic mmc2 DVD.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 1 seconds.
.............................................
.....................................Track 01: Total bytes read/written:
496412672/496412672 (242389 sectors).
.......
burn_cd was successful.


Removing temporary file system: /mkcd/cd_images...
Removing temporary file system: /mkcd/cd_fs...
Removing temporary file system: /mkcd/mksysb_image...
```

Tip: Consult your drive vendor to determine exactly what media is supported. In our testing, we recorded DVD+R format media on our DVD multi-recorder device. To do this, the **-cdformat** flag was added to the above command to burn a DVD+R in place of a CD such that the command will read:

```
$ backupios -cd /dev/cd0 -accept -cdformat
```

Note that the CD process prompts the user to place additional CDs (in our case, only one additional CD) into the drive if the backup is too large and spans multiple discs. Externally label the media to ensure proper identification and order.

Both of the methods for the CD or DVD backup produce bootable media that can be used to restore the Virtual I/O Server, as shown in “Restoring the Virtual I/O Server operating system” on page 26.

Backing up the Virtual I/O Server operating system to tape

With tape, you use the **backupios** command, as shown in Example 2-3 on page 21.

Example 2-3 Backing up the Virtual I/O Server to tape

```
$ backupios -tape /dev/rmt0

Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
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 tape boot image.............

Creating list of files to back up.
Backing up 44950 files..........................
44950 of 44950 files (100%)
0512-038 mksysb: Backup Completed Successfully.
```

Backing up the Virtual I/O Server operating system to file

To back up to a file, use the **backupios** command. The big difference here compared to tape or optical media is that all of the previous commands resulted in a form of bootable media that can be used to directly recover the Virtual I/O Server.

Backing up to a file will result in either:

- A tar file that contains all of the information needed for a restore
- A **mksysb** image

Both methods depend on an installation server for restoration.

The restoration server can be:

- An HMC using the Network Installation Manager on Linux facility and the **installios** command
- An AIX 5L Network Installation Management (NIM) server and a standard **mksysb** system installation

We discuss both of these methods later in 2.1.3, “Restoring the Virtual I/O Server” on page 26.

Important: If you are using a NIM server for the installation, it must be running a level of AIX 5L that can support the Virtual I/O Server installation. For this reason, the NIM server should be running the very latest technology level and service packs at all times.

You can use the **backupios** command to write to a local file on the Virtual I/O Server, but the more common scenario will be to perform a backup to remote NFS-based storage. The ideal situation might be to use the NIM server as the destination because this server can be used to restore these backups. In the following example, a NIM server has a host name of SERVER5 and the Virtual I/O Server is LPAR01.

The first step is to set up the NFS-based storage export on the NIM server. Here, we export a file system named /export/ios_backup, and in this case, /etc/exports looks similar to the following:

```
#more /etc/exports
/export/ios_backup -sec=sys:krb5p:krb5i:krb5:dh,rw=lpar01.itsc.austin.ibm.com,ro
ot=lpar01.itsc.austin.ibm.com
```

Important: The NFS server must have the root access NFS attribute set on the file system exported to the Virtual I/O Server logical partition for the backup to succeed.

In addition, make sure that the name resolution is functioning from the NIM server to the Virtual I/O Server and back again (reverse resolution) for both the IP and host name. To edit the name resolution on the Virtual I/O Server, use the **hostmap** command to manipulate the /etc/hosts file or the **cfgnamesrv** command to change the DNS parameters.

The backup of the Virtual I/O Server can be large, so ensure that the system **ulimits** on the NIM server will allow the creation of large files.

With the NFS export and name resolution set up, the file system needs to be mounted on the Virtual I/O Server. Using an appropriate user ID, you can use the **mount** command, as shown in Example 2-4.

Example 2-4 Mounting remote NFS-based storage and performing a backup

```
$ mount server5:/export/ios_backup /mnt
$ mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd1         /home            jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /proc            /proc            procfs Jun 27 10:48 rw
         /dev/hd10opt     /opt             jfs2   Jun 27 10:48 rw,log=/dev/hd8
server5.itsc.austin.ibm.com /export/ios_backup /mnt             nfs3   Jun 27
10:57
$ backupios -file /mnt

Creating information file for volume group storage01.

Creating information file for volume group volgrp01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...
$
```

The command creates a full backup tar file package including all of the resources that the **installios** command will need to install a Virtual I/O Server (**mksysb**, bosinst.data, network bootimage, and SPOT) from an HMC using the **installios** command.

Note that in this example the command argument is a directory; the file name will be nim_resources.tar. We describe the restoration methods in 2.1.3, “Restoring the Virtual I/O Server” on page 26.

Depending on the restore method chosen, it is possible to create the **mksysb** backup of the Virtual I/O Server as in the following examples. Note that the command argument is a fully qualified file name. At the time of writing, the NIM server only supports the **mksysb** restoration method.

> **Note:** The ability to run the **installios** command from the NIM server against the nim_resources.tar file is enabled with APAR IY85192. Check with your local IBM support representative for the availability of this service.
>
> The **mksysb** backup of the Virtual I/O Server can be extracted from the tar file created in a full backup, so either method is appropriate if the restoration method uses a NIM server.

Example 2-5 Creating a backup of the Virtual I/O Server

```
$ backupios -file /mnt/VIOS_BACKUP_27Jun2006_1205.mksysb -mksysb

/mnt/VIOS_BACKUP_27Jun2006_1205.mksysb  doesn't exist.

Creating /mnt/VIOS_BACKUP_27Jun2006_1205.mksysb

Creating information file for volume group storage01.

Creating information file for volume group volgrp01.
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 45016 files..........................
45016 of 45016 files (100%)
0512-038 savevg: Backup Completed Successfully.
$
```

Both of the methods shown create a backup of the Virtual I/O Server operating system that can be used to recover the Virtual I/O Server using either an HMC or a NIM server.

Backing up user-defined virtual devices

Record the VIOS configuration as part of your backup.

The Virtual I/O Server operating system backup, provided in the previous section, will back up the operating system, but more than that is needed to re-build a server:

- If you are restoring to the same server, some information might be available such as data structures (storage pools or volume groups and logical volumes) held on non-rootvg disks.
- If you are restoring to new hardware, these devices cannot be automatically recovered because the disk structures will not exist.
- If the physical devices exist in the same location and structures such as logical volumes are intact, the virtual devices such as virtual target SCSI and Shared Ethernet Adapters will be recovered during the restoration.

In the DR situation where these disk structures do not exist and network cards will be at different location codes, you need to make sure to back up:

- Any user-defined disk structures such as storage pools or volume groups and logical volumes
- The linking of the virtual device through to the physical devices

These devices will mostly be created at the Virtual I/O Server build and deploy time but will change depending on when new clients are added or changes are made. For this reason, a weekly schedule or manual backup procedure when configuration changes are made is appropriate.

Use the **savevgstruct** command to back up user-defined disk structures. This command writes a back up of the structure of a named volume group (and therefore storage pool) to the /home/ios/vgbackups directory.

For example, to back up the structure in the storage01 volume group, run the command as follows:

```
$ savevgstruct storage01

Creating information file for volume group storage01.

Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
```

Note: The volume groups or storage pools need to be activated for the backup to succeed. Only active volume groups or storage pools will be automatically backed up by the **backupios** command. Use the **lsvg** or **lssp** commands to list and **activatevg** to activate the volume groups or storage pools if necessary before starting the backup.

The **savevgstruct** command is automatically called before the backup commences for all active non-rootvg volume groups or storage pools on a Virtual I/O Server when the **backupios** command is run. Because this command is called before the backup commences, the volume group structures will be included in the system backup. For this reason, you can use the **backupios** command to back up the disk structure as well, so the frequency that this command run might increase.

The last item to back up is the linking information. You can gather this information from the output of the **lsmap** command, as shown in Example 2-6.

Example 2-6 Sample output from the lsmap command

```
$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                  ent2
Backing device       ent0
Physloc              U787B.001.DNW108F-P1-C1-T1

$ lsmap -all
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
```

```
vhost0          U9113.550.105E9DE-V2-C10                     0x00000000

VTD                   target01
LUN                   0x8200000000000000
Backing device        lv02
Physloc

VTD                   vtscsi0
LUN                   0x8100000000000000
Backing device        lv00
Physloc
```

From the previous example, pertaining to the disks, the vhost0 device in slot 10 on the HMC (the C10 value in the location code) is linked to the lv02 device and named target01 and also to the lv00 device and named vtscsi0. For the network, the virtual Ethernet adapter ent1 is linked to the ent0 Ethernet adapter making the ent2 Shared Ethernet Adapter.

Consideration: The previous output does not gather information such as SEA control channels (for SEA Failover), IP addresses to ping, and whether threading is enabled for the SEA devices. These settings and any other changes that have been made (for example MTU settings) must be documented separately, as explained later in this section.

Note: It is also vitally important to use the slot numbers as a reference for the virtual SCSI and virtual Ethernet devices, not the vhost number or ent number.

The vhost and ent devices are assigned by the Virtual I/O Server as they are found at boot time or when the **cfgdev** command is run. If you add in more devices after subsequent boots or with the **cfgdev** command, these will be sequentially numbered.

In the vhost0 example, the important information to note is that it is not vhost0 but that the virtual SCSI server in slot 10 (the C10 value in the location code) is mapped to logical volumes lv00 and lv02. The vhost and ent numbers are assigned sequentially by the Virtual I/O Server at initial discovery time and should be treated with caution for rebuilding user-defined linking devices.

Use these commands to record additional information.

We recommend gathering the following additional information to the **lsmap** command. This information enables the Virtual I/O Server to be rebuilt from the install media if necessary.

- Network settings

  netstat -state, **netstat -routinfo**, **netstat -routtable**, **lsdev -dev entX -attr**, **cfgnamesrv -ls**, **hostmap -ls**, **optimizenet -list**, and **entstat -all entX**

- All physical and logical volume SCSI devices

  lspv, **lsvg**, and **lsvg -lv VolumeGroup**

- All physical and logical adapters

  lsdev -type adapter

- Code levels and users and security

  ioslevel, **viosecure -firewall -view**, **viosecure -view -nonint**, **motd**, **loginmsg**, and **lsuser**

We suggest that you gather this information in the same time frame as the previous disk information. The scripting of the commands and the use of the **tee** command to write the information to a file in the /home/padmin directory is a possible solution. You can schedule this job using the **crontab** command.

The /home/padmin directory is backed up using the **backupios** command; therefore, it is a good location to collect configuration information prior to a backup.

See 2.3, “Scheduling jobs on the Virtual I/O Server” on page 39 for more information.

2.1.3 Restoring the Virtual I/O Server

Test your backup policy by restoring the data.

With all of the different backups described and the frequency discussed, we now describe how to rebuild the server from scratch. The situation we work through is a Virtual I/O Server hosting an AIX 5L operating system-based client partition running on virtual disk and network. We work through the restore from the uninstalled *bare metal* Virtual I/O Server upward and discuss where each backup strategy will be used.

This complete end-to-end solution is only for this extreme disaster recovery scenario. If you need to back up and restore a Virtual I/O Server onto the same server, probably the restoration of the operating system is of interest.

Restoring the HMC configuration

In the most extreme case of a natural or man-made disaster that has destroyed or rendered unusable an entire data center, systems might have to be restored to a disaster recovery site. In this case, you need another HMC and server location to which recover your settings. You should also have a disaster recovery server in place with your HMC profiles ready to start recovering your systems.

The details of this is beyond the scope of this document but would, along with the following section, be the first steps for a DR recovery.

Restoring other IT infrastructure devices

All other IT infrastructure devices, such as network routers, switches, storage area networks and DNS servers, to name just a few, also need to be part of an overall IT disaster recovery solution. Having mentioned them, we say no more about them apart from making you aware that not just the Virtual I/O Server but the whole IT infrastructure will rely on these common services for a successful recovery.

Restoring the Virtual I/O Server operating system

When you have a functioning IT infrastructure and a server managed by an HMC that has a Virtual I/O Server system profile defined with the same number of physical Ethernet adapters and disks as were present in the original system, you need to restore the Virtual I/O Server operating system. This is the entry point for the restoration of the Virtual I/O Server operating system if required as part of a standard backup and restore policy.

We recommend the service APAR IY85192 for AIX 5L Version 5.3 TL5 for installing from a backup.

Depending on the backup method used, you now need to select the appropriate method, which we discuss in the following sections.

Restoring the Virtual I/O Server from CD or DVD backup

The backup procedures described in this chapter create bootable media that you can use to restore as stand-alone backups.

Insert the first disk from the set of backups into the optical drive and boot the machine into SMS mode, making sure the CD or DVD drive is available to the partition. Select, using the SMS menus, to install from the optical drive and work through the usual installation procedure.

> **Note:** If the CD or DVD backup spanned multiple disks, during the install, you will be prompted to insert the next disk in the set with a similar message to the following message:
>
> `Please remove volume 1, insert volume 2, and press the ENTER key.`

Restoring the Virtual I/O Server from tape backup

The procedure for the tape is similar to the CD or DVD. Because this is bootable media, just place the backup media into the tape drive and follow boot into SMS mode. Select to install from the tape drive and follow the same procedure as previously described.

Restoring the Virtual I/O Server from the HMC

If you made a full backup to file (not a **`mksysb`** backup but a full one that creates the nim_resources.tar file), you can use the HMC to install these using the **`installios`** command.

The tar file has to be available on either a DVD or an NFS share. In this scenario, we use the NFS method. Assuming the directory that holds the nim_resources.tar file has been exported from an NFS server, you now log on to the HMC with a suitable user ID and run the **`installios`** command, as shown in Example 2-7, with the input entries appearing in bold.

> **Note:** The trailing slash in the NFS location server5:/export/ios_backup/ must be included in the command as shown.
>
> The configure client network setting must be set to `no` as shown. This is because the physical adapter we are installing the backup through might already be used by an SEA and the IP configuration will fail if this is the case. Log in and configure the IP if necessary after the installation using a console session.

Example 2-7 installios process from the HMC

```
hscroot@server1:~> installios
The following objects of type "managed system" were found.  Please select one:
1. p570-ITSO
Enter a number: 1
The following objects of type "virtual I/O server partition" were found.  Please
select one:
1. VIOS_DR_MACHINE
2. VIO_Server1
Enter a number (1-2): 1
The following objects of type "profile" were found.  Please select one:
1. normal
Enter a number: 1
Enter the source of the installation images [/dev/cdrom]:
server5:/export/ios_backup/
Enter the client's intended IP address: 9.3.5.123
Enter the client's intended subnet mask: 255.255.255.0
Enter the client's gateway: 9.3.5.41
Enter the client's speed [100]:auto
Enter the client's duplex [full]:auto
Would you like to configure the client's network after the
        installation [yes]/no? no
```

```
Retrieving information for available network adapters
This will take several minutes...
The following objects of type "Ethernet adapters" were found.  Please select one:
1. ent U9117.570.107CD9E-V2-C2-T1 523300002002
2. ent U7311.D20.10832BA-P1-C01-T1 00096b6e8458
3. ent U7311.D20.10832BA-P1-C01-T2 00096b6e8459
Enter a number (1-3): 2
Here are the values you entered:
managed system = p570-ITSO
virtual I/O server partition = VIOS_DR_MACHINE
profile = normal
source = server5:/export/ios_backup/
IP address = 9.3.5.123
subnet mask = 255.255.255.0
gateway = 9.3.5.41
speed = 100
duplex = full
configure network = no
ethernet adapters = 00:09:6b:6e:84:58
Press enter to proceed or type Ctrl-C to cancel...
nimol_config MESSAGE: No NIMOL server hostname specified, using
server1.itsc.austin.ibm.com as the default.
Starting RPC portmap daemon
done
Starting kernel based NFS server
done
nimol_config MESSAGE: Added "REMOTE_ACCESS_METHOD /usr/bin/rsh" to the file
"/etc/nimol.conf"
nimol_config MESSAGE: Removed "disable = yes" from the file "/etc/xinetd.d/tftp"
nimol_config MESSAGE: Added "disable = no" to the file "/etc/xinetd.d/tftp"
Shutting down xinetd:
done
Starting INET services. (xinetd)
done
nimol_config MESSAGE: Removed "SYSLOGD_PARAMS=" from the file
"/etc/sysconfig/syslog"
nimol_config MESSAGE: Added "SYSLOGD_PARAMS=-r " to the file
"/etc/sysconfig/syslog"
nimol_config MESSAGE: Removed "local2,local3.*  -/var/log/localmessages" from the
file "/etc/syslog.conf"
nimol_config MESSAGE: Added "local3.*  -/var/log/localmessages" to the file
"/etc/syslog.conf"
nimol_config MESSAGE: Added "local2.* /var/log/nimol.log" to the file
"/etc/syslog.conf"
Shutting down syslog services
done
Starting syslog services
done
nimol_config MESSAGE: Executed /usr/sbin/nimol_bootreplyd -l -d -f /etc/nimoltab
-s server1.itsc.austin.ibm.com.
nimol_config MESSAGE: Successfully configured NIMOL.
nimol_config MESSAGE: target directory: /info/default5
nimol_config MESSAGE: Executed /usr/sbin/iptables -I INPUT 1 -s server5 -j ACCEPT.
nimol_config MESSAGE: source directory: /mnt/nimol
nimol_config MESSAGE: Checking /mnt/nimol/nim_resources.tar for existing
resources.
nimol_config MESSAGE: Executed /usr/sbin/iptables -D INPUT -s server5 -j ACCEPT.
nimol_config MESSAGE: Added "/info/default5 *(rw,insecure,no_root_squash)" to the
file "/etc/exports"
nimol_config MESSAGE: Successfully created "default5".
nimol_install MESSAGE: The hostname "lpar11.itsc.austin.ibm.com" will be used.
nimol_install MESSAGE: Added "CLIENT lpar11.itsc.austin.ibm.com" to the file
"/etc/nimol.conf"
nimol_install MESSAGE: Added
"lpar11.itsc.austin.ibm.com:ip=9.3.5.123:ht=ethernet:gw=9.3.5.41:sm=255.255.255.0:
bf=lpar11.itsc.austin.ibm.com:sa=9.3.5.194:ha=00096b6e8458" to the file
"/etc/nimoltab"
nimol_install MESSAGE: Executed kill -HUP 5149.
nimol_install MESSAGE: Created /tftpboot/lpar11.itsc.austin.ibm.com.
nimol_install MESSAGE: Executed /sbin/arp -s lpar11.itsc.austin.ibm.com
00:09:6b:6e:84:58 -t ether.
nimol_install MESSAGE: Executed /usr/sbin/iptables -I INPUT 1 -s
lpar11.itsc.austin.ibm.com -j ACCEPT.
nimol_install MESSAGE: Created
/info/default5/scripts/lpar11.itsc.austin.ibm.com.script.
nimol_install MESSAGE: Created /tftpboot/lpar11.itsc.austin.ibm.com.info.
nimol_install MESSAGE: Successfully setup lpar11.itsc.austin.ibm.com for a NIMOL
install
# Connecting to VIOS_DR_MACHINE
# Connected
# Checking for power off.
# Power off complete.
# Power on VIOS_DR_MACHINE to Open Firmware.
# Power on complete.
# Client IP address is 9.3.5.123.
# Server IP address is 9.3.5.194.
# Gateway IP address is 9.3.5.41.
# /pci@800000020000011/pci@2/ethernet@1 ping successful.
# Network booting install adapter.
# bootp sent over network.
# Network boot proceeding, lpar_netboot is exiting.
# Finished.
```

At this point, open a terminal console on the server to which you are restoring in case user input is required.

Tip: If the command seems to be taking a long time to restore, this is most commonly caused by a speed or duplex misconfiguration in the network.

NIM is a useful tool in many production environments.

Restoring the Virtual I/O Server using a NIM server

The **installios** command is also available on the NIM server, but at present, it only supports installations from the base media of the Virtual I/O Server. The method we use from the NIM server is to install the **mksysb** image. This can either be the **mksysb** image generated with the **-mksysb** flag in the **backupios** command shown previously or you can extract the **mksysb** image from the nim_resources.tar file.

Note: The ability to run the **installios** command from the NIM server against the nim_resources.tar file is enabled with APAR IY85192. Check with your local IBM support representative for availability.

Whichever method you use to obtain the **mksysb**, after you have this on the NIM server, you need to register the **mksysb** as a NIM resource as shown:

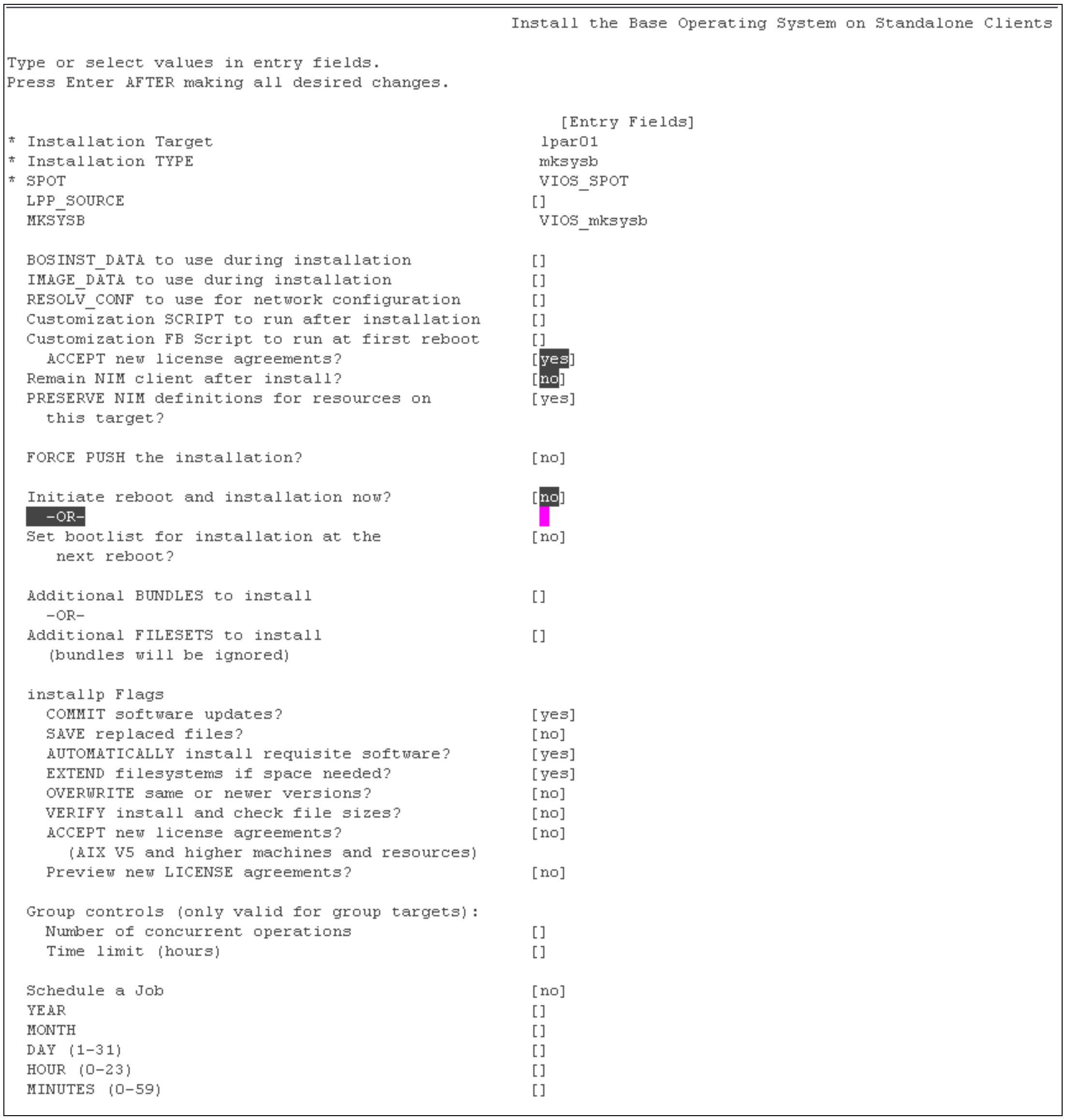
```
# nim -o define -t mksysb -aserver=master
-alocation=/export/ios_backup/VIOS_BACKUP_27Jun2006_1205.mksysb VIOS_mksysb
# lsnim VIOS_mksysb
VIOS_mksysb     resources       mksysb
```

When the **mksysb** is registered as a resource, generate a SPOT from the **mksysb**:

```
# nim -o define -t spot -a server=master -a location=/export/ios_backup/SPOT -a
source=VIOS_mksysb VIOS_SPOT

Creating SPOT in "/export/ios_backup/SPOT" on machine "master" from "VIOS_mksysb"
...
Restoring files from BOS image.  This may take several minutes ...
# lsnim VIOS_SPOT
VIOS_SPOT     resources       spot
```

With the SPOT and the **mksysb** image defined to NIM, install the Virtual I/O Server from the backup. If the machine you are installing is not defined to NIM, make sure that it is now defined as a machine and the enter the **smitty nim_bosinst** fast path. Select the **mksysb** image and SPOT defined previously so that the options are the same as shown in Figure 2-1 on page 31.



```
                                                   Install the Base Operating System on Standalone Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Installation Target                                 lpar01
* Installation TYPE                                   mksysb
* SPOT                                                VIOS_SPOT
  LPP_SOURCE                                         []
  MKSYSB                                              VIOS_mksysb

  BOSINST_DATA to use during installation            []
  IMAGE_DATA to use during installation              []
  RESOLV_CONF to use for network configuration       []
  Customization SCRIPT to run after installation     []
  Customization FB Script to run at first reboot     []
    ACCEPT new license agreements?                   [yes]
  Remain NIM client after install?                   [no]
  PRESERVE NIM definitions for resources on          [yes]
    this target?

  FORCE PUSH the installation?                       [no]

  Initiate reboot and installation now?              [no]
    -OR-
  Set bootlist for installation at the               [no]
     next reboot?

  Additional BUNDLES to install                      []
    -OR-
  Additional FILESETS to install                     []
    (bundles will be ignored)

  installp Flags
    COMMIT software updates?                         [yes]
    SAVE replaced files?                             [no]
    AUTOMATICALLY install requisite software?        [yes]
    EXTEND filesystems if space needed?              [yes]
    OVERWRITE same or newer versions?                [no]
    VERIFY install and check file sizes?             [no]
    ACCEPT new license agreements?                   [no]
      (AIX V5 and higher machines and resources)
    Preview new LICENSE agreements?                  [no]

  Group controls (only valid for group targets):
    Number of concurrent operations                  []
    Time limit (hours)                               []

  Schedule a Job                                     [no]
  YEAR                                               []
  MONTH                                              []
  DAY (1-31)                                         []
  HOUR (0-23)                                        []
  MINUTES (0-59)                                     []
```

Figure 2-1   Restoring the Virtual I/O Server using NIM

Important: Note that the “Remain NIM client after install” field is set to no. If this is not set to no, the last step for the NIM installation is to configure an IP address onto the physical adapter through which the Virtual I/O Server has just been installed. This IP address is used to register with the NIM server. If this is the adapter used by an existing Shared Ethernet Adapter, it will cause error messages similar to those shown here. If this is the case, reboot the Virtual I/O Server if necessary, and then log on to the Virtual I/O Server using a terminal session and remove any IP address information and the SEA. After this, re-create the SEA and place the IP address back onto the Virtual I/O Server.

```
inet0 changed
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
Method error (/usr/lib/methods/chgif):
        0514-068 Cause not known.
0821-510 ifconfig: error calling entry point for /usr/lib/drivers/if_en: The
specified device does not exist.
0821-103 : The command /usr/sbin/ifconfig en0 inet  9.3.5.111 arp netmask
255.255.255.0 mtu 1500 up failed.
0821-007 cfgif: ifconfig command failed.
        The status of"en0" Interface in the current running system is
uncertain.
0821-103 : The command /usr/lib/methods/cfgif -len0 failed.
0821-510 ifconfig: error calling entry point for /usr/lib/drivers/if_en: The
specified device does not exist.
0821-103 : The command /usr/sbin/ifconfig en0 inet 9.3.5.111 arp netmask
255.255.255.0 mtu 1500 up failed.
0821-229 chgif: ifconfig command failed.
The status of"en0" Interface in the current running system is uncertain.

mktcpip: Problem with command: chdev
, return code = 1
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
```

Now that you set up the NIM server to push out the backup image, the Virtual I/O Server server needs to have the remote IPL setup completed. For this procedure, see the AIX information → Installation and Migration → Partitions category of the IBM eServer™ pSeries and AIX Information Center at:

http://publib16.boulder.ibm.com/pseries/index.htm

Tip: One of the main reasons for installation problems using NIM is the NFS exports from the NIM server. Make sure that the /etc/exports file is correct on the NIM server.

The installation of the Virtual I/O Server should complete, but here is a big difference between restoring to the existing server and restoring to a new disaster recovery server. One of the NIM install options is to preserve the NIM definitions for resources on the target. With this option, NIM will attempt to restore any virtual devices that were defined in the original backup. This depends on the same devices being defined in the partition profile (virtual and physical) such that the location codes have not changed.

This means that virtual target SCSI devices and shared Ethernet adapters should all be recovered without any need to re-create them (assuming the logical partition profile has not changed). If restoring to the same machine, there is a dependency that the non-rootvg volume groups are present to be imported and any logical volume structure contained on these are intact. To demonstrate this, a Virtual I/O Server was booted from a diagnostics CD and the Virtual I/O Server operating system disks were formatted and certified, destroying all data (this was done for demonstration purposes). The other disks containing volume groups and storage pools were not touched.

Using a NIM server, the backup image was restored to the initial Virtual I/O Server operating system disks. Examining the virtual devices after the installation, we found that the vtscsi and shared Ethernet adapters are all recovered, as shown in Example 2-8.

Example 2-8 Restoration of Virtual I/O Server with recover devices set to the same logical partition

```
$ lsdev -virtual
name            status                                             description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
target01        Available  Virtual Target Device - Logical Volume
vtscsi0         Available  Virtual Target Device - Logical Volume
ent2            Available  Shared Ethernet Adapter
$ lsmap -all
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
vhost0          U9113.550.105E9DE-V2-C10                     0x00000000

VTD                   target01
LUN                   0x8200000000000000
Backing device        lv02
Physloc

VTD                   vtscsi0
LUN                   0x8100000000000000
Backing device        lv00
Physloc

$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                   ent2
Backing device        ent0
Physloc               U787B.001.DNW108F-P1-C1-T1
```

If you restore to a different logical partition where you have defined similar virtual devices from the HMC recovery step provided previously, you will find that there are no linking devices.

Note: The devices will always be different between machines because the machine serial number is part of the virtual device location code for virtual devices. For example:

```
$ lsdev -dev ent1 -vpd
  ent1     U9113.550.105E9DE-V2-C2-T1  Virtual I/O Ethernet Adapter (l-lan)
.
. Lines omitted for clarity
.
```

This is because the backing devices are not present for the linking to occur; the physical location codes have changed, so the mapping fails. Example 2-9 shows the same restore of the Virtual I/O Server originally running on a p5-550 onto a p5-570 that has the same virtual devices defined in the same slots.

Example 2-9 Devices recovered if restored to a different server

```
$ lsdev -virtual
name            status                                            description
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
$ lsmap -all -net
SVEA   Physloc
------ --------------------------------------------
ent2   U9117.570.107CD9E-V2-C2-T1

SEA                 NO SHARED ETHERNET ADAPTER FOUND
$ lsmap -all
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.570.107CD9E-V2-C10                     0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND
$
```

You now need to recover the user-defined virtual devices and any backing disk structure.

Recovering user-defined virtual devices and disk structure

On our original Virtual I/O Server, we had two additional disks from those in the rootvg. If these were SAN disks or disks that were just directly mapped to virtual I/O clients (we are linking the hdisk devices), we could just restore the virtual device links. However, if we had a logical volume or storage pool structure on the disks, we need to restore this structure first. To do this, you need to use the volume group data files.

The volume group or storage pool data files should have been captured as part of the backup process earlier. These files should be located in the /home/ios/vgbackups directory if you performed a full backup using the **backupios** command. The following command lists all of the available backups:

```
$ restorevgstruct -ls
total 16
-rw-r--r--   1 root     staff          1486 Jul 10 17:56 storage01.data
-rw-r--r--   1 root     staff          1539 Jul 10 17:56 volgrp01.data
$
```

In the following example (Example 2-10), there are some new blank disks and the same storage01 and volgrp01 volume groups to restore.

Example 2-10   Disks and volume groups to restore

```
$ lspv
NAME            PVID                                VG              STATUS
hdisk0          00c7cd9e1f89130b                    rootvg          active
hdisk1          00c7cd9e1adcd58a                    rootvg          active
hdisk2          0021768aaae4ae06                    None
hdisk3          0021768accc91a48                    None
hdisk4          none                                None
hdisk5          0021768aa445b99d                    None
```

The **restorevgstruct** command restores the volume group structure onto the empty disks. Example 2-11 shows how to call the command.

Example 2-11   Restoring the volume group structure

```
$ $ restorevgstruct -vg storage01 hdisk2
hdisk2
storage01
lv00

Will create the Volume Group:   storage01
Target Disks:   Allocation Policy:
        Shrink Filesystems:     no
        Preserve Physical Partitions for each Logical Volume:   no
```

After you restore all of the logical volume structures, the only remaining step is to restore the virtual devices linking the physical backing device to the virtual. To restore these, use the **lsmap** outputs recorded from the backup steps in “Backing up user-defined virtual devices” on page 23, or build documentation. As previously noted, it is important to use the slot numbers and backing devices when restoring these linkings.

The restoration of the Shared Ethernet Adapters will need the linking of the correct virtual Ethernet adapter to the correct physical adapter. Usually, the physical adapters will be placed into a VLAN within the network infrastructure of the organization. It is important that the correct virtual VLAN is linked to the correct physical VLAN. Any network support team or switch configuration data can help with this task.

The DR restore involves a bit more manual re-creating of virtual linking devices (vtscsi and SEA) and relies on good user documentation. If there is no multipath setup on the Virtual I/O Server to preserve, another solution can be to install the Virtual I/O Server fresh from the installation media and then restore from the build documentation.

After running the **mkvdev** commands to re-create the mappings, the Virtual I/O Server will host virtual disks and networks that can be used to rebuild the AIX 5L or Linux clients.

Restoring the AIX 5L or Linux operating system

After you have the Virtual I/O Server operational and all of the devices re-created, you are ready to start restoring any AIX 5L or Linux clients. The procedure for this should already be defined within your organization and, most likely, will be identical to that for any server using dedicated disk and network resources. The method depends on the solution employed and should be defined by you.

For AIX 5L clients, this information is in the Information Center:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/backmeth.htm

2.2 Defining a new LPAR

Planning before creating a new configuration is a best practice.

When defining new LPARs, those using virtual devices or not, planning is crucial. Before you start planning, we suggest that you have a name and slot numbering conventions in place, which apply for all the managed systems in the enterprise. A numbering convention that might be helpful is in 3.3.2, "Virtual device slot numbers" on page 59 and 4.3.2, "Virtual device slot numbers" on page 97. In addition, IBM provides tools, such as the System Planning Tool, that might be useful in planning and design the managed system, available at:

http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/

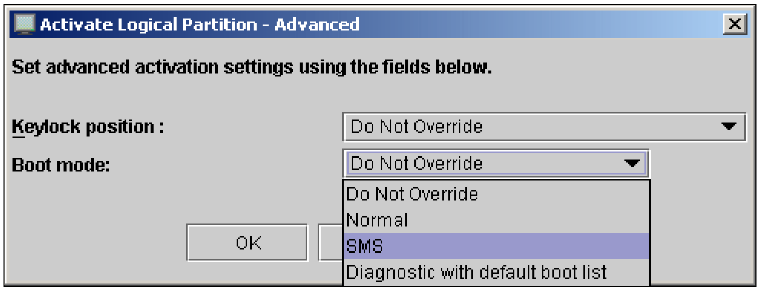
2.2.1 Creating new Virtual I/O Servers

For the steps to create a new Virtual I/O Server, see *Advanced POWER Virtualization on IBM System p5*, SG24-7940. When you start creating the Virtual /O Server LPAR, you will, on the first window, have to select the partition environment. If the Virtual I/O Server is unavailable, virtualization is not activated. To activate virtualization, visit the following Web page to read about how to get the activation code. Note that this is a feature code and it must be part of your machine configuration. For the activation code, see:

http://www.ibm.com/systems/p/cod/

You need the machine type and the serial number from Server Properties tab at the HMC. After entering the machine type and the serial number on the Web site, locate the VET code, which is the virtualization activation code.

When you create the partition profile, remember that there is no need for a normal and an SMS version of all the profiles in order to boot the client into SMS mode. You can change the boot mode on the Activation window by selecting **Advanced** and changing the boot mode to **SMS**, as shown in Figure 2-2.


Activate Logical Partition - Advanced

Set advanced activation settings using the fields below.

Keylock position : Do Not Override

Boot mode: Do Not Override

Figure 2-2 Activate Logical Partition

In Virtual I/O Server Version 1.2 and later, it is possible to virtualize optical devices through the Virtual I/O Server to the virtual I/O clients. In Example 2-12 on page 37, the virtual device is presented as a SCSI CD-ROM device. Note that you can assign the media to one virtual I/O client at a time. If another virtual I/O client requires the media device, it has to be unassigned from the current virtual I/O client and assigned to the new virtual I/O client. To share the optical device, you can use NFS or a similar product. If you use NFS to share the optical media within a managed system, you can use virtual Ethernet to access it. You cannot boot the virtual I/O clients from the optical media using an NFS shared optical device.

Example 2-12 SMS menu on the virtual I/O client

```
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.        -      Virtual Ethernet
                 ( loc=U9113.550.105E9DE-V3-C3-T1 )
2.        -      SCSI CD-ROM
                 ( loc=U9113.550.105E9DE-V3-C21-T1-W8200000000000000-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:
```

Use realistic estimates for CPU and memory when creating a partition profile.

When selecting memory values for the Virtual I/O Server LPAR, select the maximum value with care. If you select a large number, such as 128 GB, you will pin a lot of memory. The hypervisor firmware will reserve 1/64th of the value entered as the maximum value in the hypervisor firmware system memory; so if you select 128 GB, you will reserve 2 GB memory in the hypervisor system memory for a value you might never need. You can obtain the hypervisor memory used in the IBM System Planning Tool (SPT).

When you select the number of CPUs, use a realistic estimate for the CPUs. If you have a two CPU workload that might expand up to four CPUs, do not enter 12 CPUs. Unless your production workload validates a smaller value, start with an allocation of at least a whole CPU for the Virtual I/O Server if you plan to have high network traffic. In general, network traffic increases CPU utilization. Disk traffic does not, in general, require the same amount of CPU because the I/Os are queued to slower devices. For more about CPU sizing, see 5.4.2, “Virtual I/O Server CPU planning” on page 126.

If you want to make a dual Virtual I/O Servers scenario with Shared Ethernet Adapter Failover, on the primary Virtual I/O Server, select the value **1** in the Trunk priority panel, and use the value **2** on the standby Virtual I/O. Then, create the control path between the two Virtual I/O Servers that is used by the Shared Ethernet Adapter Failover. Ensure that the Access external network is not selected for this virtual adapter. If you want to use link aggregation, make sure the network switch supports IEEE 802.3ad. For information about defining virtual Ethernet, see 3.6, “Extending VLANs into virtual networks” on page 64.

2.2.2 Dynamic LPAR assignments

Dynamic LPAR assignments and updates to the partition profile must be planned.

When the you remove, change, or add I/O adapters, memory, CPU, or virtual adapters using dynamic LPAR, they are not automatically reflected in the partitions profile defined for the partition. Therefore, whenever you reconfigure, we recommend that you update the profile. A good procedure is first to put the changes into the partition profile and then make the changes dynamically to prevent them from being lost in the case of an deactivate or a shut down. The changes made to the partition in the partition profile are not reflected in the LPAR if you perform a reboot, such as **`shutdown -Fr`**; the changes are activated only from a Not Activated state.

An alternative way to save the active partition assignments is by using the save option, which is selected from the partition context menu, as shown in Figure 2-3. When you save the new profile, select a meaningful name such as `20jun06_newnormal`, as shown in Figure 2-4. After testing the new partition profile, rename the old partition profile to, for example, `21jun06_oldnormal` and rename or copy the new partition profile `20jun06_newnormal` to `normal`. Remember to document what you changed in the new partition profile. In addition, yo might want to clean up any old unused partition profiles from the last save or rename at this time.

If you have multiple workloads that require multiple partition profiles do not clean up the partition profiles and do not rename the partition profiles. Just activate them and do not use a date naming but a naming that this meaningful to this scenario, such as DB2_high.

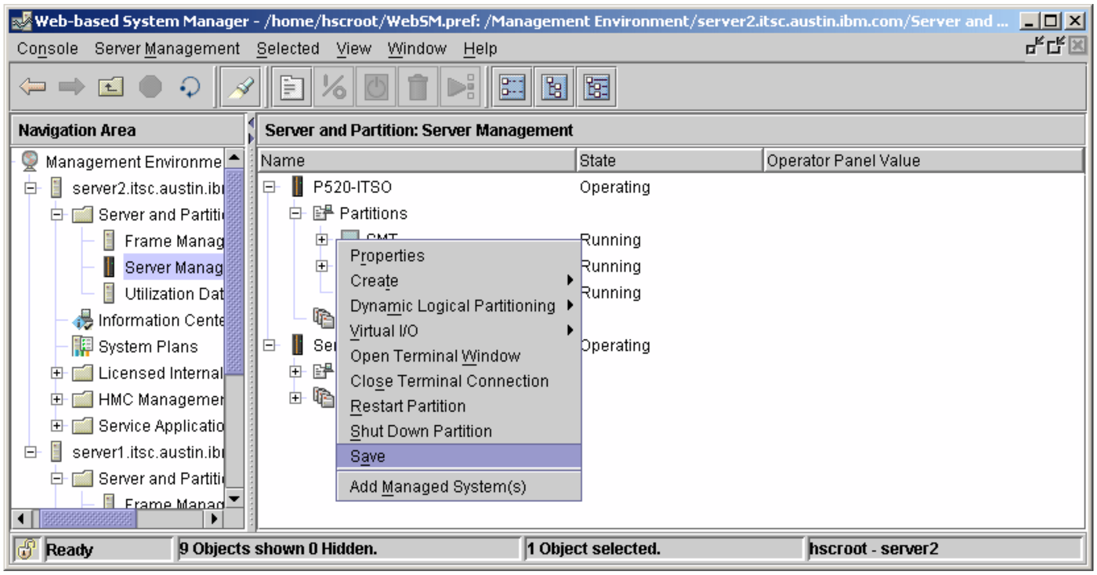
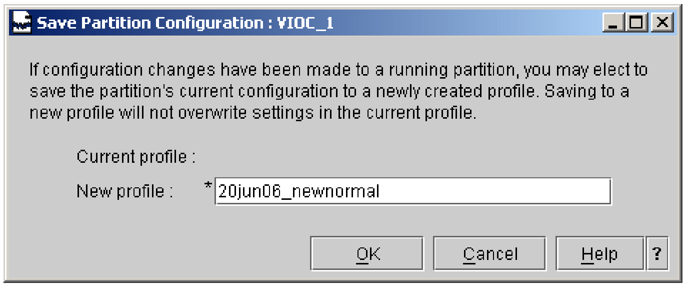


Figure 2-3 Partition context menu



Save Partition Configuration : VIOC_1

If configuration changes have been made to a running partition, you may elect to save the partition's current configuration to a newly created profile. Saving to a new profile will not overwrite settings in the current profile.

Current profile :

New profile : * 20jun06_newnormal

Figure 2-4 Save the running partition configuration

2.3 Scheduling jobs on the Virtual I/O Server

Use cron to automate vital operations

With Virtual I/O Server Version 1.3, the **crontab** command is available to enable you to submit, edit, list, or remove cron jobs. A cron job is a command run by the cron daemon at regularly scheduled intervals such as system tasks, nightly security checks, analysis reports, and backups.

With the Virtual I/O Server, a cron job can be submitted by specifying the **crontab** command with the **-e** flag. The **crontab** command invokes an editing session that enables you to modify the padmin users' crontab file and create entries for each cron job in this file.

Note: When scheduling jobs, use the padmin user's crontab file. You cannot create or edit other users' crontab files.

When you finish creating entries and exit the file, the **crontab** command copies it into the /var/spool/cron/crontabs directory and places it in the padmin file.

The following syntax is available to the **crontab** command:

```
crontab [ -e padmin | -l padmin | -r padmin | -v padmin ]
```

-e padmin Edits a copy of the padmin's crontab file. When editing is complete, the file is copied into the crontab directory as the padmin's crontab file.

-l padmin Lists padmin's crontab file.

-r padmin Removes the padmins crontab file from the crontab directory.

-v padmin Lists the status of the padmin's cron jobs.

2.4 Automating the partition startup and shutdown sequence

Make the VIOS services available first upon system initialization.

If you have an environment where the Virtual I/O Server provides virtual devices for AIX 5L or Linux operating system-based clients, and you want to shut down or start the server, you need to differentiate between these two types of servers. If the AIX 5L or Linux operating system-based servers are using virtual devices, make sure that the Virtual I/O Servers are started first.

You can make the startup of the Virtual I/O Servers easier by placing them in a system profile and starting this system profile. These system profiles contain logical partitions and an associated partition profile to use. For more information about system profiles, see the IBM Systems Hardware Information Center:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphbl/iphblmanagesprofile.htm

To document the complete solution, we also describe starting the entire IBM System p5 server in a power off state. To automate the entire procedure, use the SSH remote command execution functionality of the HMC to perform this.

From a central control console running SSH, you can issue remote commands to the HMC to perform all of the operations needed to power on a system, start the system profile for the Virtual I/O Servers, and then start the system profile for the all of the AIX 5L or Linux operating system based-clients. To automate this procedure, perform an SSH key exchange from our management console onto the HMC.

The process for this is in the IBM Systems Hardware Information Center:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphai/settingupsecurescriptexecutionsbetweensshclientsandthehmc.htm

Tip: If you have performed the SSH key exchange, you can call the command remotely. If you want to perform the **date** command, for example, to print the date from the HMC remotely, run the following command:

```
# ssh hscroot@<hostname or IP address of HMC> date
Fri Jun 30 10:54:28 CDT 2006
```

With the SSH key exchange, you will notice that there is no password prompt, so commands can be run remotely from a script.

The HMC commands provided in the following sections are examples of how to automate the start and were accurate at the time of writing. Check that the commands and syntax have not changed for your environment.

The shutdown procedure is the reverse of the following steps.

2.4.1 Remotely powering on a System p5 server to partition standby

Use the **chsysstate** command on the HMC for the remote power on of a System p5 server. To power on a system to partition standby mode, run the following command, where the managed system name is the name of the server:

```
chsysstate -m <managed system name> -o onstandby -r sys
```

Tip: To monitor the status of the server startup, use the **lsrefcode** command and check the LED status codes. An example of this command is:

```
lsrefcode -r sys -m <managed system name> -F refcode
```

2.4.2 Remotely starting a Virtual I/O Server

To start the Virtual I/O Server profiles, we set up two system profiles, as shown in Figure 2-5 on page 41, namely the Virtual_IO_Servers and Virtual_IO_Client.

The first system profile we call starts the Virtual I/O Server:

```
chsysstate -m <managed system name> -o on -r sysprof -n Virtual_IO_Servers
```

Tip: Use the **lsrefcode** command to monitor the state of the partitions being started, for example:

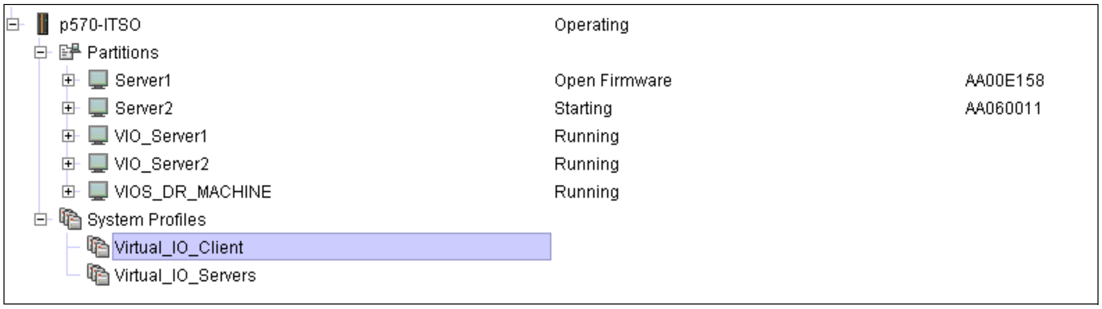
```
lsrefcode -r lpar -m <managed system name> -F lpar_name,refcode
```

2.4.3 Remotely starting logical partitions with virtual devices

After the Virtual I/O Servers start, you can call the system profile to start the AIX 5L and Linux operating system-based logical partitions on the server:

```
chsysstate -m <managed system name> -o on -r sysprof -n Virtual_IO_Client
```

After the commands run, if you look on the HMC, you should find that the Virtual I/O Servers and clients are all started or in the process of starting, similar to those shown in Figure 2-5 on page 41.



```
p570-ITSO                          Operating
  Partitions
    Server1                        Open Firmware     AA00E158
    Server2                        Starting          AA060011
    VIO_Server1                    Running
    VIO_Server2                    Running
    VIOS_DR_MACHINE                Running
  System Profiles
    Virtual_IO_Client
    Virtual_IO_Servers
```

Figure 2-5 Virtual I/O Server and AIX 5L or Linux partitions started from command line

Tip: If you do not want to use the system profiles, an advanced solution is to use the **lssyscfg** command and examine the lpar_env field. With this, you can build up a list of Virtual I/O Servers and AIX 5L or Linux operating system-based logical partitions on a managed server, as in the following example:

```
hscroot@server1:~> lssyscfg -r lpar -m p570-ITSO  -F name,lpar_env
Server1,aixlinux
VIOS_DR_MACHINE,vioserver
VIO_Server1,vioserver
VIO_Server2,vioserver
Server2,aixlinux
```

2.4.4 Automation summary

The examples provided show the steps followed for the startup procedure. The shutdown procedure is the reverse of these steps with the AIX 5L or Linux operating system-based clients shut down, followed by the Virtual I/O Servers, and lastly the System p5 server hardware.

2.5 Virtual I/O Server maintenance

We describe two scenarios for upgrading a Virtual I/O Server in this section. We recommend that you have a dual Virtual I/O Servers environment to perform regular service to provide a continuous connection of your clients to their virtual I/O resources. For clients using non-critical virtual resources, or when you have service windows that allow a Virtual I/O Server to be rebooted, we discuss a single Virtual I/O Server scenario.

2.5.1 Single Virtual I/O Server scenario

VIOS service requires client considerations.

When applying routine service that requires a reboot in a single Virtual I/O Server environment, you need to determine how this will affect virtual I/O clients using the resources of that Virtual I/O Server. Ensure that the update or upgrade of the Virtual I/O Server does not disrupt any critical resources.

Tip: Back up the Virtual I/O Servers and the virtual I/O clients if a current backup is not available, and document the virtual Ethernet and SCSI devices before the upgrade or update. This reduces the time used in a recovery scenario.

To avoid complication during an upgrade or update, we advise you to check the environment before upgrading or updating the Virtual I/O Server. The following list is an sample of useful commands for the virtual I/O client and Virtual I/O Server:

lsvg rootvg On the Virtual I/O Server and virtual I/O client, check for stale PPs and stale PV.

lsvg -pv rootvg On the Virtual I/O Server, check for missing disks.

netstat -cdlistats On the Virtual I/O Server, check that the Link status is Up on all used interfaces.

errpt On the virtual I/O client, check for CPU, memory, disk, or Ethernet errors, and resolve them before continuing.

lsvg -p rootvg On the virtual I/O client, check for missing disks.

netstat -v On the virtual I/O client, check that the Link status is Up on all used interfaces.

Before starting an upgrade, take a backup of the Virtual I/O Server and the virtual I/O clients if an current backup is not available. To back up the Virtual I/O Serve, use the **backupios** command in 2.1, “Backing up and restoring the Virtual I/O Server” on page 16 and back up the virtual I/O client using **mksysb**, **savevg**, IBM Tivoli® Storage Manager, or similar backup products.

To update the Virtual I/O Server, use the following steps, in this case from an attached optical drive:

1. Shut down the virtual I/O clients connected to the Virtual I/O Server, or disable any virtual resource that is in use.
2. Apply the update with the **updateios** command. Press y to start the update. In this case, we installed from an optical drive, but you can copy the CD to the NIM server by using **bffcreate** command and then mount that file system on the Virtual I/O Server using NFS.

```
$ updateios -dev /dev/cd0 -install -accept

*******************************************************************************
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
  -----------------
.
. (Lines omitted for clarity)
.
RESOURCES
```

```
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                     Needed Space             Free Space
      /usr                                 7248                1098264
      -----                          --------                   ------
      TOTAL:                               7248                1098264

  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]?
```

3. Reboot the standby Virtual I/O Server when the update finishes:

```
$ shutdown -restart

SHUTDOWN PROGRAM
Mon Nov 30 21:57:23 CST 1970

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
.
. (Lines omitted for clarity)
.
```

4. Check the new level with the **ioslevel** command.
5. Check the disks, Ethernet adapters, and so on, on the Virtual I/O Server.
6. Open the virtual I/O client or reconnect the virtual I/O resources, and check the virtual I/O client.

Verify the Virtual I/O Server environment and document the update.

2.5.2 Dual Virtual I/O Servers scenario

When applying an update to the Virtual I/O Server in a dual Virtual I/O Servers environment, you will be able to do so without having downtime to the virtual I/O services. If the Virtual I/O Server is upgraded from Version 1.1 to 1.2 or later, and you want to migrate at the same time to new functions, such as from Network Interface Backup to Shared Ethernet Adapter Failover on the clients, you have to plan network impacts on the virtual I/O client in order to change the virtual network setup. It is not mandatory to migrate from Network Interface Backup to Shared Ethernet Adapter Failover, but it is an example of using the new function.

Tip: Back up the Virtual I/O Servers and the virtual I/O clients if a current backup is not available, and document the virtual Ethernet and SCSI device before the upgrade or update. This reduces the time used in a recovery scenario.

Verify there are no existing problems before performing an upgrade.

The update of a Virtual I/O Server in a dual Virtual I/O Servers environment is the same if it is a migration from Version 1.2 to 1.3 or applying an update to an existing Virtual I/O Server. It is a best practice to check the virtual Ethernet and disk devices on the Virtual I/O Server and virtual I/O client before starting the update on either of the Virtual I/O Servers. Check the physical adapters to verify connections. As shown in Example 2-13, Example 2-14, and Example 2-15 on page 45, all the virtual adapters are up and running.

Example 2-13 The netstat -v comand on the virtual I/O client

```
netstat -v
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational

Hypervisor Send Failures: 0
  Receiver Failures: 0
  Send Errors: 0

Hypervisor Receive Failures: 0

ILLAN Attributes: 0000000000003002 [0000000000002000]
.
. (Lines omitted for clarity)
.
#
```

Example 2-14 The netstat -cdlistats command on the Virtual I/O Server 1

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Trunk Adapter: True
  Priority: 1  Active: True
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

Example 2-15 The netstat -cdlistats command on the Virtual I/O Server 2

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Trunk Adapter: True
  Priority: 2  Active: False
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

Upon further investigation, when looking at the physical adapters (Example 2-16 and Example 2-17), they are both down, so there is no connectivity. If this a backup line or similar, your system administrator might not have noticed this. Check the physical adapters.

Example 2-16 The netstat -cdlistats command on the Virtual I/O Server 1

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902) Specific Statistics:
------------------------------------------------------------------------
Link Status : Down
Media Speed Selected: Auto negotiation
.
. (Lines omitted for clarity)
.
$
```

Example 2-17 The netstat -cdlistats command on the Virtual I/O Server 2

```
$ netstat -cdlistat
.
. (Lines omitted for clarity)
.
10/100/1000 Base-TX PCI-X Adapter (14106902) Specific Statistics:
-----------------------------------------------------------------
Link Status : Down
Media Speed Selected: Auto negotiation
.
. (Lines omitted for clarity)
.
$
```

Checking the disk status depends on how the disks are shared from the Virtual I/O Server.

If you have an MPIO setup similar to Figure 2-6, run the following commands before and after the first Virtual I/O Server update to verify the disk path status:

`lspath` On the virtual I/O client, check all the path to the disks. They should all be in the enabled state.

`lsattr -El hdisk0` On the virtual I/O client, check the MPIO heartbeat for hdisk0, the attribute hcheck_mode is set to `nonactive`, and hcheck_interval is `60`. If you run IBM SAN storage, check that reserve_policy is `no_reserve`; other storage vendors might require other values for reserve_policy. This command should be executed on all disks from the Virtual I/O Server.

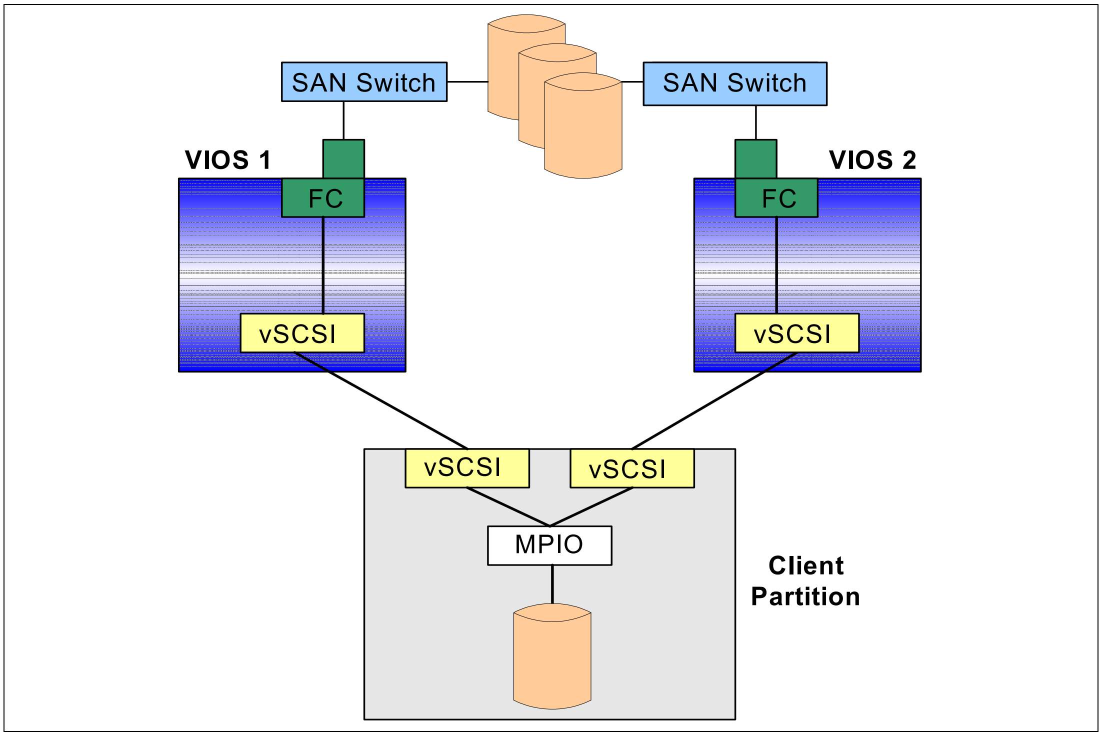


Figure 2-6 Client MPIO

If you use an LVM disk environment is similar to Figure 2-7 on page 47, check the LVM status for the disk shared from the Virtual I/O Server with the following commands:

`lsvg rootvg` On the virtual I/O client, check for stale PPs, and the quorum must be off.

`lsvg -p rootvg` On the virtual I/O client, check for missing hdisk.

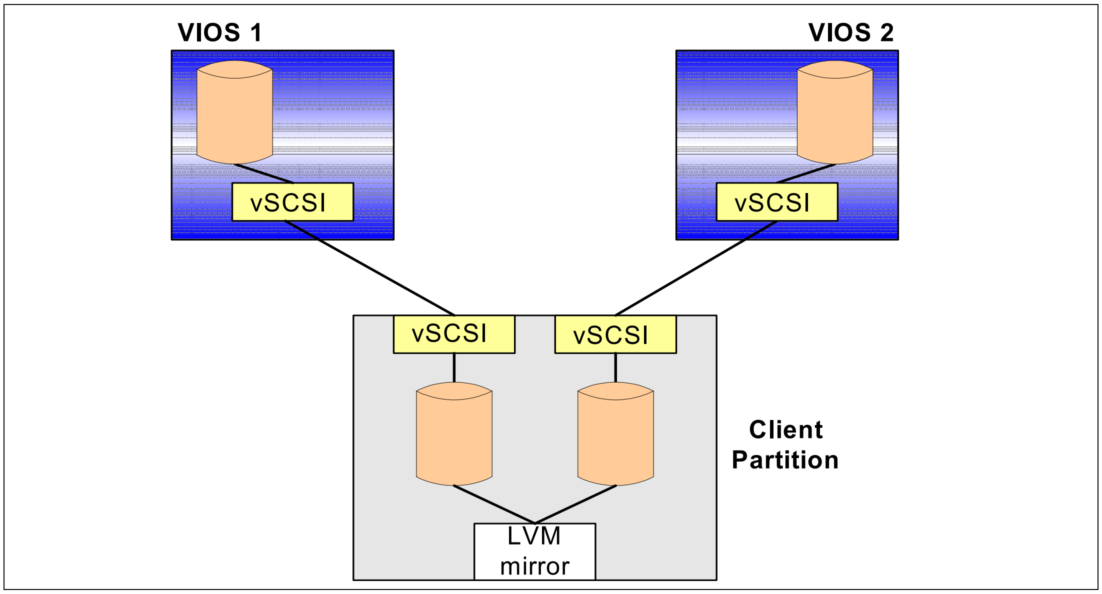


Figure 2-7 Client LVM mirroring

After checking the environment and resolving any issues, back up the Virtual I/O Server and virtual I/O client if an current backup is not available.

To update or upgrade the Virtual I/O Server:

1. Find the standby Virtual I/O Server and enter the **netstat** command. At the end of the output, locate the priority of the Shared Ethernet Adapter adapter and whether it is active. In this case, the standby adapter is not active, so you can begin the upgrade of this server.

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Trunk Adapter: True
  Priority: 2  Active: False
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

2. Apply the update with the **updateios** command and press y to start the update. In this case, we install from an optical drive, but you copy the update using **bffcreate** command to the NIM server and then mount that file system on the Virtual I/O Server.

```
$ updateios -dev /dev/cd0 -install -accept

*******************************************************************************
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
  -----------------
.
. (Lines omitted for clarity)
.
RESOURCES
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                     Needed Space             Free Space
      /usr                                 7248                1098264
      -----                              --------               ------
      TOTAL:                               7248                1098264

  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]?
```

3. Reboot the standby Virtual I/O Server when the update completes:

```
$ shutdown -restart

SHUTDOWN PROGRAM
Mon Nov 30 21:57:23 CST 1970

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
.
. (Lines omitted for clarity)
```

4. Check the new level with the **ioslevel** command.

5. Verify that the standby Virtual I/O Server and the virtual I/O client is connected to Virtual I/O Server environment. If you have an MPIO environment, as shown in Figure 2-6 on page 46, run the **lspath** command on the virtual I/O client and verify that the all paths are enabled. If you have an LVM environment, as shown in Figure 2-7 on page 47, you will have to run the **varyonvg** command, and the volume group should begin to sync. If not, run the **syncvg -v** command on the volume groups that used the virtual disk from the Virtual I/O Server environment to sync each volume group.

```
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            511         488         102..94..88..102..102
hdisk1            missing           511         488         102..94..88..102..102
# varyonvg rootvg
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            511         488         102..94..88..102..102
hdisk1            active            511         488         102..94..88..102..102
# lsvg rootvg
VOLUME GROUP:       rootvg                  VG IDENTIFIER:  00c478de00004c00000
00006b8b6c15e
VG STATE:           active                  PP SIZE:        64 megabyte(s)
VG PERMISSION:      read/write              TOTAL PPs:      1022 (65408 megabytes)
MAX LVs:            256                     FREE PPs:       976 (62464 megabytes)
LVs:                9                       USED PPs:       46 (2944 megabytes)
OPEN LVs:           8                       QUORUM:         1
TOTAL PVs:          2                       VG DESCRIPTORS: 3
STALE PVs:          0                       STALE PPs:      0
ACTIVE PVs:         2                       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
#
```

6. Verify that the Ethernet services connecting to the Virtual I/O Server are using the Shared Ethernet Adapter Failover scenario using the **netstat -cdlistat** command on Virtual I/O Server or the **netstat -v** command on the virtual I/O client for Network Interface Backup. Check the link status for the Network Interface Backup adapters.

7. If you use Shared Ethernet Adapter Failover, shift the standby and primary connection to the Virtual I/O Server using the **chdev** command and check with the **netstat -cdlistats** command that the state has changed, as shown in this example:

```
$ chdev -dev ent4 -attr ha_mode=standby
ent4 changed
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Trunk Adapter: True
  Priority: 1  Active: False
Filter MCast Mode: False
.
. (Lines omitted for clarity)
.
$
```

8. Apply the update to the Virtual I/O Server, which now is the standby Virtual I/O Server, using the **updateios** command.
9. Reboot the Virtual I/O Server with the **shutdown -restart** command.
10. Check the new level with the **ioslevel** command.
11. Verify that the standby Virtual I/O Server and the virtual I/O client are connected to Virtual I/O Server environment. If you have an MPIO environment, as shown in Figure 2-6 on page 46, run the **lspath** command on the virtual I/O client and verify that the all paths are enabled. If you have an LVM environment, as shown in Figure 2-7 on page 47, run the **varyonvg** command, and the volume group should begin to sync. If not, use the **syncvg -v** command on each volume group that uses virtual disk from the Virtual I/O Server environment.
12. Verify that the Ethernet connects to the Virtual I/O Server using the **netstat -cdlistats** command and use the **netstat -v** command to check for link status.
13. Reset the Virtual I/O Server role back to primary using the **chdev** command, as shown in the following example:

```
$ chdev -dev ent4 -attr ha_mode=auto
ent4 changed
$
```

The update is now complete.

3

Networking

Network connectivity in the virtual environment is extremely flexible. This chapter describes best practices for virtual network configuration. We discuss Shared Ethernet Adapter configuration, along with high availability scenarios, VLAN tagging, security zones, and tuning packet sizes for best performance.

3.1 IP addresses on the Virtual I/O Server

Ensure network connectivity to your Virtual I/O Server.

The Virtual I/O Server can be accessed from the HMC using a secure private HMC to service processor network to open a console session. This makes a dedicated network address on the Virtual I/O Server for administration optional. However, if the Virtual I/O Server does not appear on any network at all, dynamic resource allocation will not be enabled for the Virtual I/O Server because there is no way to connect to it. The mkvterm and vtmenu panels on the HMC assist you with this configuration.

3.1.1 Multiple IP addresses on the Virtual I/O Server

Generally, the Virtual I/O Server only requires a single IP address that is configured on a management VLAN and is accessible by the HMC. The communication between the HMC and the Virtual I/O Server is important because it is this connection that enables dynamic resource allocation.

There are situations where multiple IP addresses are required to extend high availability. These are related to setting up Shared Ethernet Adapter (SEA) Failover and Network Interface Backup (NIB) on the Virtual I/O Server.

IP addresses in a Shared Ethernet Adapter Failover environment

When using Shared Ethernet Adapter (SEA) Failover on the Virtual I/O Server, there are some types of network failures that do not trigger a failover of the Shared Ethernet Adapter (SEA). This is because keep-alive messages are only sent over the control channel. No keep-alive messages are sent over other SEA networks or over the external network. But the SEA Failover feature can be configured to periodically check that a given IP address can be reached. The SEA periodically pings this IP address, so it can detect certain network failures. In this case, the Shared Ethernet Adapters must have network interfaces with IP addresses associated to be able to use this periodic self diagnostic.

These IP addresses must be unique and must use different IP addresses on each Shared Ethernet Adapter. The Shared Ethernet Adapters must have IP addresses to provide return-to-address for the ICMP-Echo-Requests sent and ICMP-Echo-Replies received when the given IP address are pinged. Figure 3-1 on page 53 is an example of how to configure this optional feature. For details about how to configure the overall Shared Ethernet Adapter Failover feature, refer to 3.9, “Creating Shared Ethernet Adapters to support SEA Failover” on page 76.

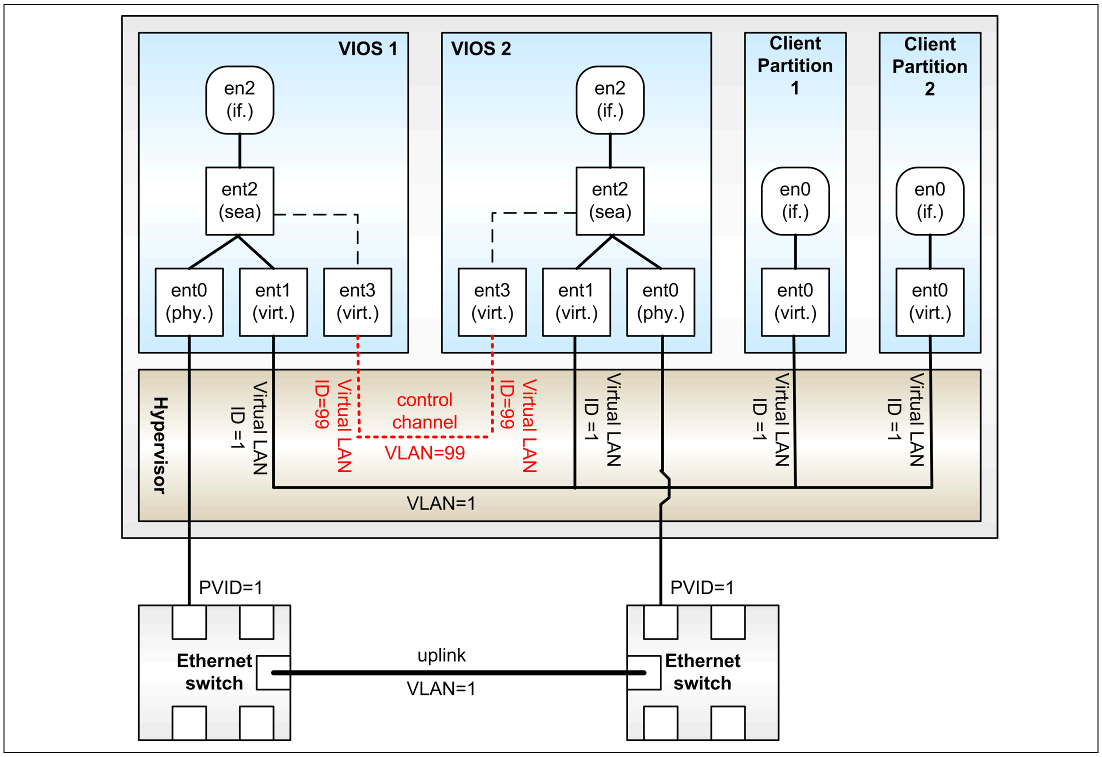


Figure 3-1 SEA Failover example

Figure 3-1 shows the configuration and the text that follows provides an example of how to configure this optional feature:

1. On each Virtual I/O Server, use the **mkvdev -sea** command to configure a Shared Ethernet Adapter using the netaddr attribute to allow the Shared Ethernet Adapter to periodically check that a given IP address can be reached:

```
$ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1 -attr ha_mode=auto
ctl_chan=ent3 netaddr=9.3.5.41
ent2 Available
en2
et2
```

2. Use the **lsdev** command to confirm that the netaddr attribute is configured with an IP address:

```
$ lsdev -dev ent2 -attr
attribute     value    description
user_settable

ctl_chan      ent3     Control Channel adapter for SEA failover                         True
ha_mode       auto     High Availability Mode                                           True
largesend     0        Enable Hardware Transmit TCP Resegmentation                      True
netaddr       9.3.5.41 Address to ping                                                  True
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
thread        1        Thread mode enabled (1) or disabled (0)                          True
virt_adapters ent1     List of virtual adapters associated with the SEA (comma separated) True
```

3. Then, configure an IP address on the Shared Ethernet Adapter (en2) using the **mktcpip** command.

IP addresses when using Network Interface Backup

To provide network redundancy in a virtualized environment, use the Network Interface Backup (NIB) feature on a client partition using dual Virtual I/O Servers. This feature can also be implemented on the Virtual I/O Server itself to provide increased network redundancy in the event a network switch becomes unavailable. Figure 3-2 provides an example configuration using a single Virtual I/O Server.

> **Note:** If you have two Virtual I/O Servers connected into different switches, NIB might not be necessary. The SEA Failover mechanism for the virtual I/O clients can be used to protect against a switch failure. If you are running dual Virtual I/O Servers, and one is taken out of operation for service or replacement, you might want NIB to provide redundancy through this remaining Virtual I/O Server. You can combine NIB and SEA; they are not mutually exclusive.

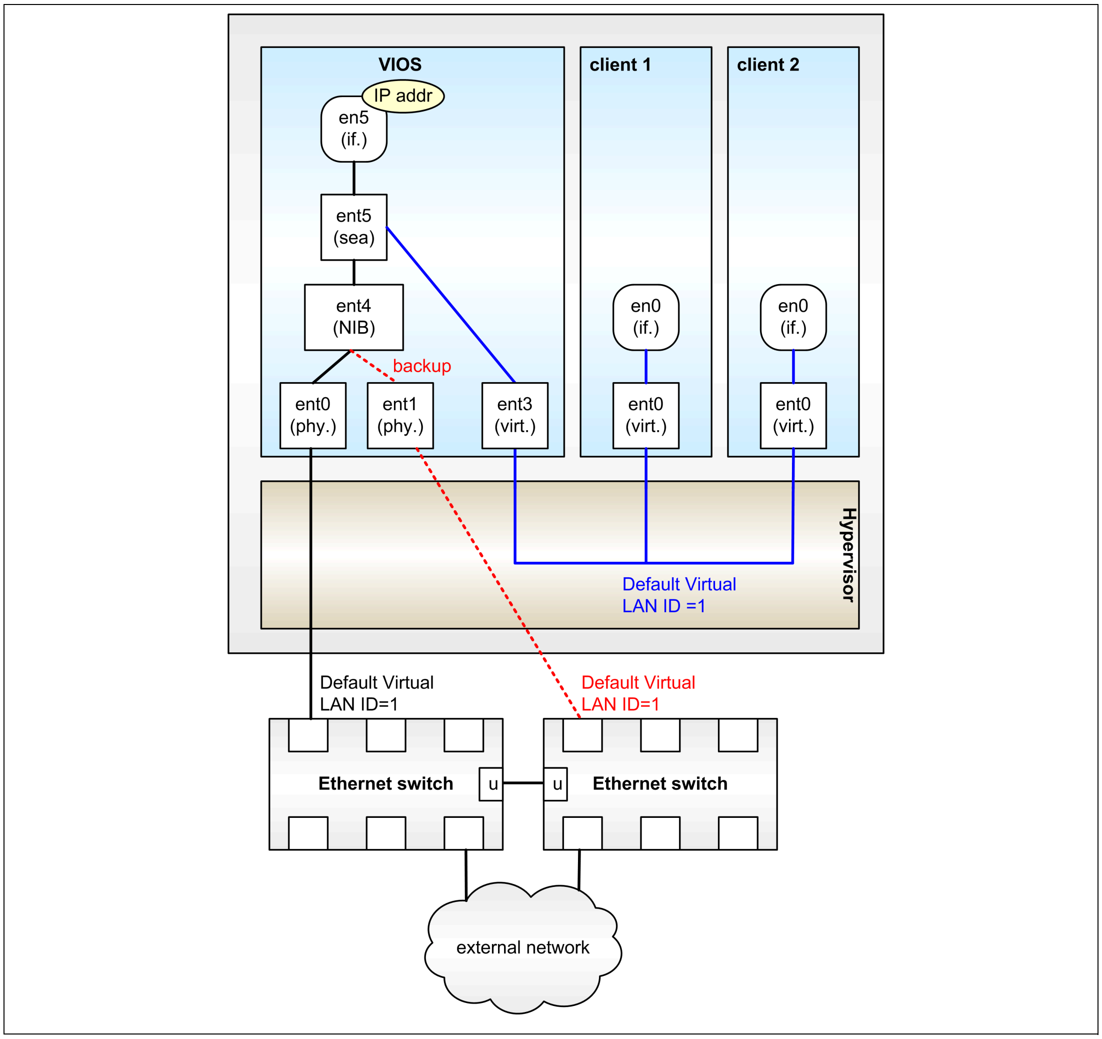


Figure 3-2 NIB on a single Virtual I/O Server configuration

Figure 3-2 on page 54 shows the configuration and the text that follows provides an example configuration using a single Virtual I/O Server:

1. Configure the Network Interface Backup device on the Virtual I/O Server using the **mkvdev** command. In this example, ent0 is the primary adapter, ent1 is the backup adapter, and the netaddr attribute is used as the Internet address to ping. The default gateway is usually used for the netaddr attribute. Ensure that the address used is reachable through the SEA using the routing in effect.

   ```
   $ mkvdev -lnagg ent0 -attr backup_adapter=ent1 netaddr=9.3.5.198
   ent4 Available
   en4
   et4
   ```

   > **Note:** If one of the network interfaces (en0 or en1) on the client partition has already been configured as part of a NIM installation (or clone), it is necessary to take the interface down as well and remove it so that it is not configured before the Network Interface Backup interface is configured.

2. Using the **lsdev** command, confirm the configuration of the Network Interface Backup device:

   ```
   $ lsdev -dev ent4 -attr
   attribute       value          description
   user_settable

   adapter_names   ent0           EtherChannel Adapters                     True
   alt_addr        0x000000000000 Alternate EtherChannel Address            True
   auto_recovery   yes            Enable automatic recovery after failover  True
   backup_adapter  ent1           Adapter used when whole channel fails     True
   hash_mode       default        Determines how outgoing adapter is chosen True
   mode            standard       EtherChannel mode of operation            True
   netaddr         9.3.5.198      Address to ping                           True
   num_retries     3              Times to retry ping before failing        True
   retry_time      1              Wait time (in seconds) between pings      True
   use_alt_addr    no             Enable Alternate EtherChannel Address     True
   use_jumbo_frame no             Enable Gigabit Ethernet Jumbo Frames      True
   ```

3. Create the Shared Ethernet Adapter using the Network Interface Backup device as the physical adapter:

   ```
   $ mkvdev -sea ent4 -vadapter ent3 -default ent3 -defaultid 1
   ent5 Available
   en5
   et5
   ```

4. Using the **lsdev** command, confirm the creation of the SEA device:

   ```
   $ lsdev -type adapter
   name            status                                         description
   ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
   ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
   ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
   ent3            Available  Virtual I/O Ethernet Adapter (l-lan)
   ent4            Available  EtherChannel / IEEE 802.3ad Link Aggregation
   ent5            Available  Shared Ethernet Adapter
   ide0            Available  ATA/IDE Controller Device
   sisioa0         Available  PCI-X Dual Channel U320 SCSI RAID Adapter
   vhost0          Available  Virtual SCSI Server Adapter
   vhost1          Available  Virtual SCSI Server Adapter
   vsa0            Available  LPAR Virtual Serial Adapter
   ```

5. Using the **mktcpip** command, configure an IP address on the newly created SEA:

   ```
   $ mktcpip -hostname lpar02 -inetaddr 9.3.5.112 -interface en5 -netmask
   255.255.255.0 -gateway 9.3.5.41 -nsrvaddr 9.3.4.2 -nsrvdomain
   itsc.austin.ibm.com
   ```

6. Using the **netstat** command, check the IP address on the Shared Ethernet Adapter:

   ```
   $ netstat -num -state
   Name  Mtu   Network     Address           Ipkts Ierrs    Opkts Oerrs  Coll
   en5   1500  link#5      0.2.55.2f.a7.e        3936     0     1181     0     0
   en5   1500  9.3.5       9.3.5.112             3936     0     1181     0     0
   lo0   16896 link#1                           14370     0    14567     0     0
   lo0   16896 127         127.0.0.1            14370     0    14567     0     0
   lo0   16896 ::1                              14370     0    14567     0     0
   ```

3.1.2 Where to configure the IP address on the Virtual I/O Server

When deciding where to configure the IP address on the Virtual I/O Server, the recommended approach is to configure an IP address on the Shared Ethernet Adapter itself. This ensures that network connectivity to the Virtual I/O Server is independent of the internal virtual network configuration, and also facilitates the implementation of the Shared Ethernet Adapter Failover and Network Interface Backup (on the Virtual I/O Server) environments, where the remote address to ping feature is required.

Due to the these points, we recommend using a standardized approach by configuring the IP address on the Shared Ethernet Adapter of the Virtual I/O Server.

Note: Some installation-specific situations require that an IP address is configured somewhere other than the Shared Ethernet Adapter. Consult with IBM support services if you experience virtual Ethernet performance issues.

3.2 Changing IP addresses or VLAN

This section describes how to change the IP address and the VLAN within a virtual I/O environment and the impact on the Virtual I/O Server and the virtual I/O client.

3.2.1 On the Virtual I/O Server

The following sections pertain to the Virtual I/O Server.

Changes to the IP address

The IP address on the Shared Ethernet Adapter is used for:

- Heartbeat for Shared Ethernet Adapter Failover
- RMC communication for dynamic LPAR on the Virtual I/O Server
- Logon process on the Virtual I/O Server
- NIM installation or restore (the **installios** command) of the Virtual I/O Server
- Performance monitoring using the **topas** command
- Update or upgrade the Virtual I/O Server from the NIM server
- Backup of the Virtual I/O Server to the NIM server or other network servers

The IP address assigned to the Shared Ethernet Adapter has no meaning to the virtual I/O client. Therefore, the IP address on the Shared Ethernet Adapter device can be changed without affecting the virtual I/O client using the Shared Ethernet Adapter device.

If the IP address must be changed on en4 from 9.3.5.112 to 9.3.5.113 and the host name from lpar02 to lpar03, use the following command:

```
mktcpip -hostname lpar03 -inetaddr 9.3.5.113 -interface en4
```

Note that if you want to change the adapter at the same time, such as from en4 to en8, you have to delete the TCP/IP definitions on en4 first by using the **rmtcpip** command and then running the **mktcpip** command on en8.

Changes to the VLAN

You can add or remove the VLANs on an existing tagged virtual Ethernet adapter using dynamic LPAR without interrupting the service running on that Virtual I/O Server. Remember to change the partition profile to reflect the dynamic change by either using the save option or the properties option on the partition context menu.

To start bridging a new tagged VLAN ID, you can create a new virtual Ethernet with a throwaway PVID and any tagged VIDs you want. Then, use dynamic LPAR to move it into the Virtual I/O Server, and use the **chdev** command to add the new adapter to the list of virt_adapters of the SEA. It will immediately begin bridging the new VLAN ID without an interruption.

You can add a new physical Ethernet adapter, a new Shared Ethernet Adapter, and a new virtual Ethernet adapter to make a tagged virtual Ethernet adapter. Doing this, you can move from an untagged to tagged virtual Ethernet adapter. This requires a small planned service window as the virtual I/O clients are moved from the untagged to the tagged adapter, such as any change of IP address would require in a non-virtualized environment.

This also applies when you move from tagged to untagged virtual Ethernet adapters. We recommend that you plan and document a change from untagged to tagged virtual Ethernet adapters or vice versa.

3.2.2 On the virtual I/O client

The following sections pertain to the virtual I/O client.

Changes to the IP address

To change the IP address on a virtual Ethernet adapter on the virtual I/O client, use SMIT or the **mktcpip** command. In this example, we change the IP address from 9.3.5.113 to 9.3.5.112 and the host name from lpar03 to lpar02. The virtual Ethernet adapter can be modified in the same way you modify a physical adapter, using the following command:

```
mktcpip -h lpar02 -a 9.3.5.112 -i en0
```

Changes to the VLAN

If you want to change the VLAN information at the Virtual I/O Server, it is possible to add or remove the VLANs on an existing tagged virtual Ethernet adapter using dynamic LPAR without interrupting the network service running to the virtual I/O clients. Adding additional IP addresses at the virtual I/O clients can be done as an alias IP address and it will not interrupt the network service on that virtual I/O client. Keep in mind, as with all dynamic LPAR changes, to change the partition profile by either using the save option or the properties option on the partition context menu.

With virtual I/O clients, you cannot change from a untagged to a tagged virtual Ethernet adapter without interrupting that network service. You can add a new virtual Ethernet adapter and make that a tagged virtual Ethernet adapter. In this way, you can move from an untagged to tagged virtual Ethernet adapter requiring a small planned service window as the virtual I/O client is moved from the untagged to the tagged adapter. This is the same interruption as a change of IP address would require in a non-virtualized environment.

This also applies when you move from tagged to untagged virtual Ethernet adapters. We recommend that you plan and document a change from untagged to tagged virtual Ethernet adapters or tagged to untagged.

3.3 Managing the mapping of network devices

One of the keys to managing a virtual environment is keeping track of what virtual objects correspond to what physical objects. In the network area, this can involve physical and virtual network adapters, and VLANs that span across hosts and switches. This mapping is critical to manage performance and to understand what systems will be affected by hardware maintenance.

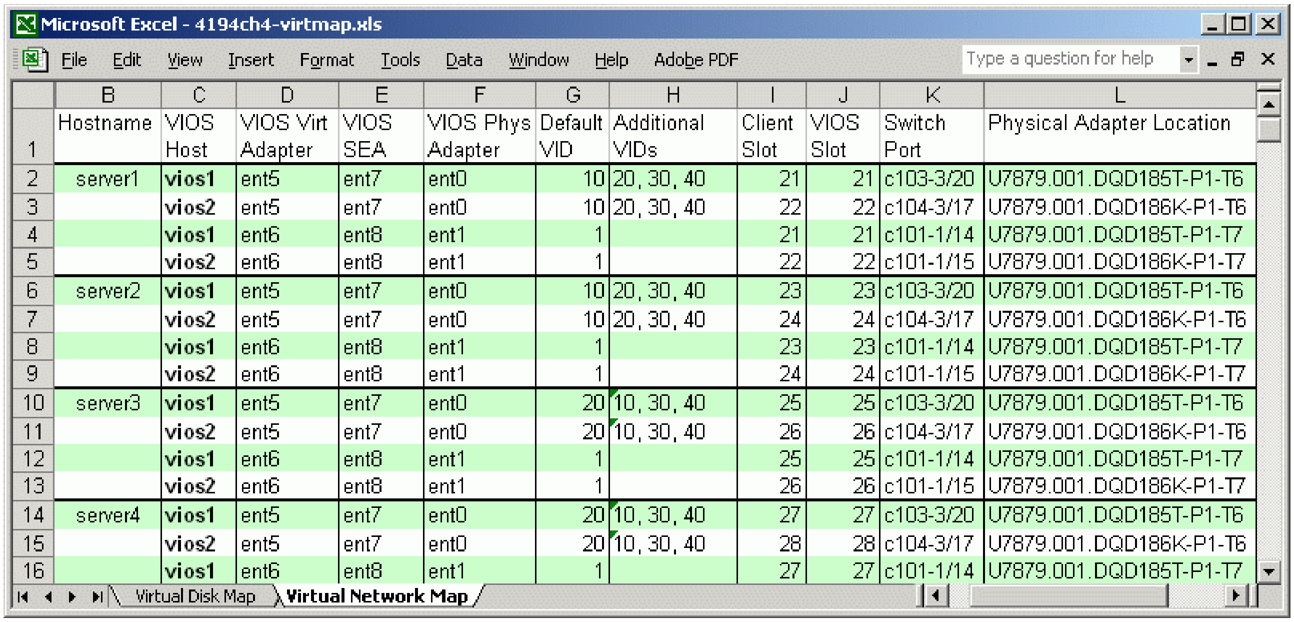
For information about mapping storage devices, see 4.3, “Managing the mapping of LUNs to VSCSI to hdisks” on page 94.

In environments that require redundant network connectivity, this section focuses on the SEA Failover method in preference to the Network Interface Backup method of providing redundancy.

Depending on whether you choose to use 802.1Q tagged VLANs, you might need to track the following information:

- Virtual I/O Server
  - Server host name
  - Physical adapter device name
  - Switch port
  - SEA device name
  - Virtual adapter device name
  - Virtual adapter slot number
  - Port virtual LAN ID (in tagged and untagged usages)
  - Additional virtual LAN IDs
- Virtual I/O client
  - Client host name
  - Virtual adapter device name
  - Virtual adapter slot number
  - Port virtual LAN ID (in tagged and untagged usages)
  - Additional virtual LAN IDs

Because of the number of fields to be tracked, we recommend the use of a spreadsheet or database program, such as shown in Figure 3-3 on page 59, to track this information. Record the data when the system is installed, and track it over time as the configuration changes.



	B	C	D	E	F	G	H	I	J	K	L
1	Hostname	VIOS Host	VIOS Virt Adapter	VIOS SEA	VIOS Phys Adapter	Default VID	Additional VIDs	Client Slot	VIOS Slot	Switch Port	Physical Adapter Location
2	server1	vios1	ent5	ent7	ent0	10	20, 30, 40	21	21	c103-3/20	U7879.001.DQD185T-P1-T6
3		vios2	ent5	ent7	ent0	10	20, 30, 40	22	22	c104-3/17	U7879.001.DQD186K-P1-T6
4		vios1	ent6	ent8	ent1	1		21	21	c101-1/14	U7879.001.DQD185T-P1-T7
5		vios2	ent6	ent8	ent1	1		22	22	c101-1/15	U7879.001.DQD186K-P1-T7
6	server2	vios1	ent5	ent7	ent0	10	20, 30, 40	23	23	c103-3/20	U7879.001.DQD185T-P1-T6
7		vios2	ent5	ent7	ent0	10	20, 30, 40	24	24	c104-3/17	U7879.001.DQD186K-P1-T6
8		vios1	ent6	ent8	ent1	1		23	23	c101-1/14	U7879.001.DQD185T-P1-T7
9		vios2	ent6	ent8	ent1	1		24	24	c101-1/15	U7879.001.DQD186K-P1-T7
10	server3	vios1	ent5	ent7	ent0	20	10, 30, 40	25	25	c103-3/20	U7879.001.DQD185T-P1-T6
11		vios2	ent5	ent7	ent0	20	10, 30, 40	26	26	c104-3/17	U7879.001.DQD186K-P1-T6
12		vios1	ent6	ent8	ent1	1		25	25	c101-1/14	U7879.001.DQD185T-P1-T7
13		vios2	ent6	ent8	ent1	1		26	26	c101-1/15	U7879.001.DQD186K-P1-T7
14	server4	vios1	ent5	ent7	ent0	20	10, 30, 40	27	27	c103-3/20	U7879.001.DQD185T-P1-T6
15		vios2	ent5	ent7	ent0	20	10, 30, 40	28	28	c104-3/17	U7879.001.DQD186K-P1-T6
16		vios1	ent6	ent8	ent1	1		27	27	c101-1/14	U7879.001.DQD185T-P1-T7

Figure 3-3 Network tracking spreadsheet

3.3.1 Virtual network adapters and VLANs

Virtual network adapters operate at memory speed. In many cases where additional physical adapters are needed, there is no need for additional virtual adapters. However, transfers that remain inside the virtual environment can benefit from using large MTU sizes on separate adapters. This can lead to improved performance and reduced CPU utilization for transfers that remain inside the virtual environment. See 3.11, “Jumbo frame and path MTU discovery” on page 79 for more information.

The POWER Hypervisor™ supports tagged VLANs that can be used to separate traffic within the system. Separate adapters can be used to accomplish the same goal. Which method you choose, or a combination of both, should be based on common networking practice in your data center.

3.3.2 Virtual device slot numbers

Slot numbers are shared between virtual storage and virtual network devices. In complex systems, there will tend to be far more storage devices than network devices, because each virtual SCSI device can only communicate with one server or client. We recommend that the slot numbers through 20 be reserved for network devices on all LPARs in order to keep the network devices grouped together. In some complex network environments with many adapters, more slots might be required for networking. For more information about storage slot numbering, see 4.3.2, “Virtual device slot numbers” on page 97.

The maximum number of virtual adapter slots per LPAR should be increased above the default value of 10 when you create an LPAR. The appropriate number for your environment depends on the number of LPARs and adapters expected on each system. Each unused virtual adapter slot consumes a small amount of memory, so the allocation should be balanced with expected requirements. Use the System Planning Tool available from the following URL to plan memory requirements for your system configuration:

http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html

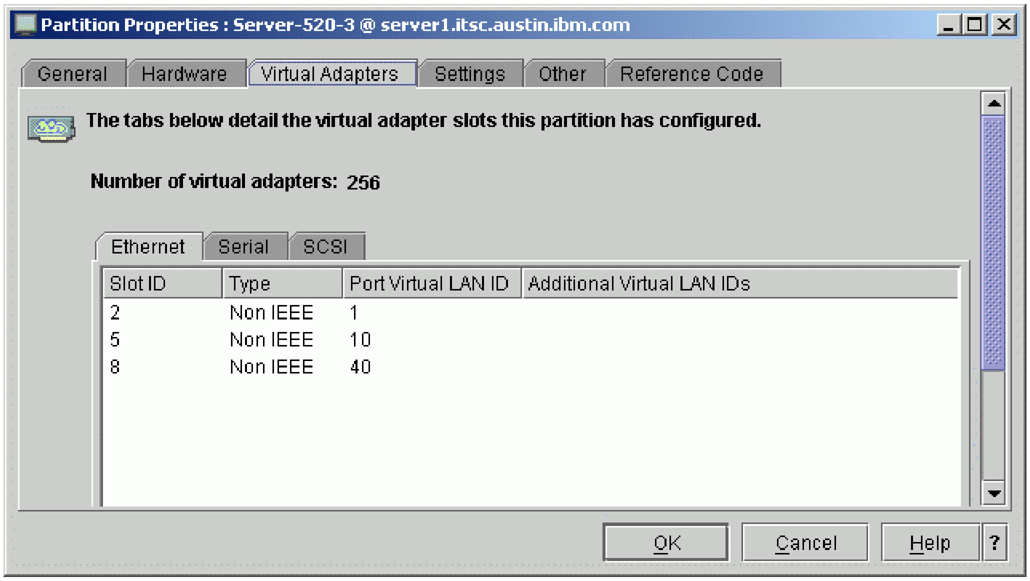
3.3.3 Tracing configuration

Despite the best intentions in record keeping, it sometimes becomes necessary to manually trace virtual network connections back to the physical hardware.

In a virtual I/O client partition with multiple virtual network adapters, the slot number of each adapter can be determined using the adapter physical location from the **lscfg** command. In the case of virtual adapters, this field includes the card slot following the letter C as shown in the following example:

```
# lscfg -l ent\*
  ent2           U9111.520.10DDEDC-V5-C5-T1  Virtual I/O Ethernet Adapter (l-lan)
  ent1           U9111.520.10DDEDC-V5-C8-T1  Virtual I/O Ethernet Adapter (l-lan)
  ent0           U9111.520.10DDEDC-V5-C2-T1  Virtual I/O Ethernet Adapter (l-lan)
```

You can use the slot numbers from the physical location field to trace back through the HMC and determine what connectivity and VLAN tags are available on each adapter, as shown in Figure 3-4. In this case, ent0 is in slot 2 on VLAN1, ent1 is in slot 8 on VLAN 40, and ent2 is in slot 5 on VLAN 10.



The tabs below detail the virtual adapter slots this partition has configured.

Number of virtual adapters: 256

Slot ID	Type	Port Virtual LAN ID	Additional Virtual LAN IDs
2	Non IEEE	1	
5	Non IEEE	10	
8	Non IEEE	40	

Figure 3-4 Virtual Ethernet adapter slot assignments

3.4 Managing multiple network security zones

VLAN tagging provides a means to isolate network traffic within a single link.

It is possible to extend multiple VLAN tagged networks into a virtualization environment, because the hypervisor emulates an Ethernet switch supporting IEEE 802.1Q VLAN tagging. If you select the IEEE 802.1Q compatible adapter check box when you create the virtual adapter, this ensures that every data frame is tagged and only presented for the virtual Ethernet adapter in the same VLAN.

In the sample shown in Figure 3-5 on page 61, we use VLAN 10 and VLAN 20, so the partitions 1 and 3 can communicate with each other using the hypervisor firmware, but partition 2 cannot communicate with 1 or 3. In addition, the hypervisor is designed in a way that no operation within a client partition can gain control of or use a shared resource that is not assigned to that client partition.

You can change the assignment for the client partition on the Virtual I/O Server, so the user ID and password has to be keep secure, such as the administrator password for the Ethernet switches. In general, you should put an IP address on the Virtual I/O Server.

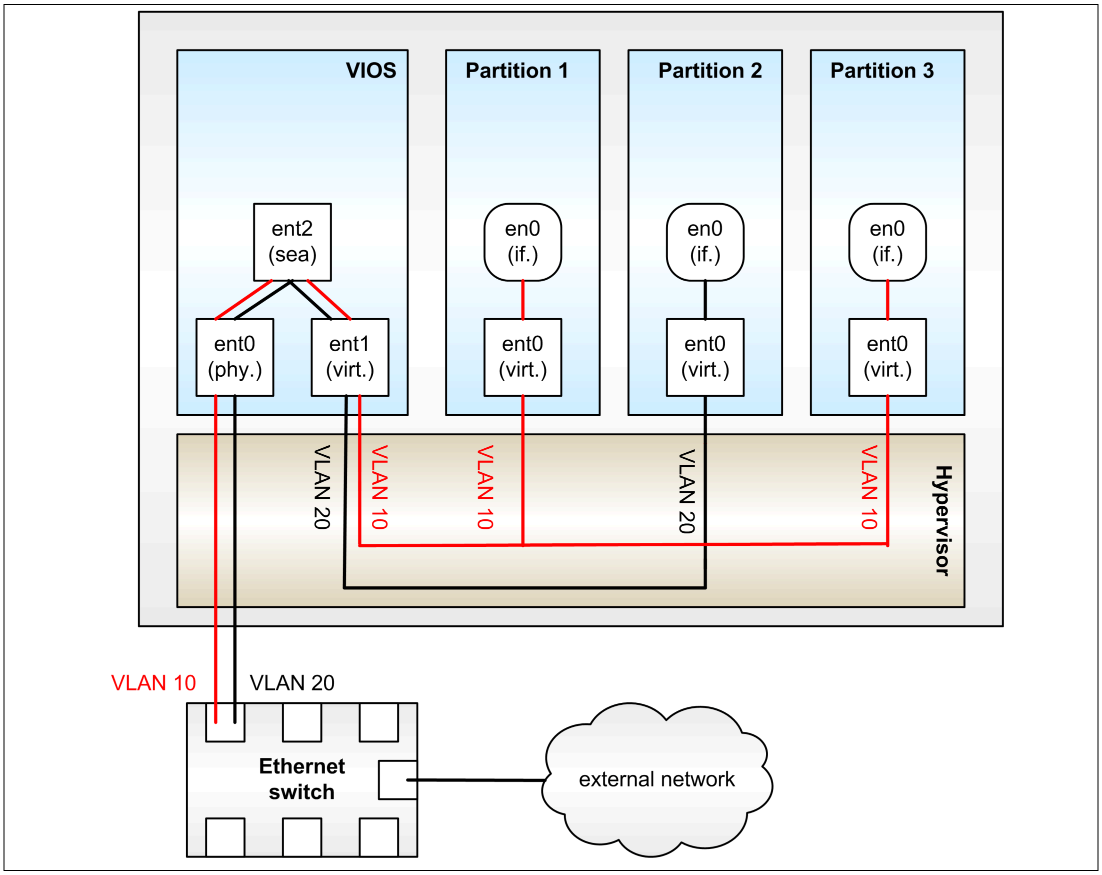


Figure 3-5 VLAN tagged environment

This implementation requires that the enterprise security policy encompasses:

- The enterprise security policy recognizes IEEE 802.1Q VLAN tagging.

  The IEEE 802.1Q VLAN tagging is implemented in the Advanced POWER Virtualization hypervisor firmware. The Virtual I/O Server is able to have up to 21 VLANs per Shared Ethernet Adapter, but in order to use this, the physical network port must support the same number of VLANs. The physical VLAN policy within the enterprise will therefore determine the virtual VLAN policy. For more information, see 3.6, “Extending VLANs into virtual networks” on page 64.

- The security policy allows a network switch to have multiple VLANs.

  If the enterprise policy allows multiple VLANs to share a network switch (non-physical security), you can implement the scenario shown in Figure 3-5. If it is a security requirement that a network switch only have one VLAN, every VLAN will require a separate managed system. If you just make a separate Virtual I/O Server in a managed system, the hypervisor firmware will act like one switch with multiple VLANs, which in this case, is not allowed by the security policy outside the Virtual I/O Server.

Note: Security in a virtual environment depends on the HMC or Integrated Virtualization Manager (IVM) and the Virtual I/O Server. Access to the HMC, IVM, and VIOS must be closely monitored to prevent unauthorized modification of existing network and VLAN assignments, or establishing new network assignments on LPARs within the managed systems.

Ensure that security zones are part of your corporate security policy.

If the requirement is a user network (VLAN 10 and 20) and an administration network (VLAN 80 and 90), see Figure 3-7 on page 63, you must extend the VLAN into the client partitions. This can be done in two ways (Figure 3-6).

- Partition 1 shows VLAN devices in the client partition, on top of the virtual Ethernet adapter. This mean that if the VLAN ID changes, you must also change the VLAN tag ID in client partition and not only in the hypervisor layer and Virtual I/O Server. You must have access to the client partitions in order to change the VLAN ID. This can facilitate additional security if the HMC, VIOS, and the partition administrators are not a shared role.
- Partition 2 presents the VLAN tagged as two virtual Ethernet adapters. The client partition is not aware of the VLAN tagging since this handled by the virtual devices. A change of VLAN ID for a client partitions is done from the HMC or VIOS only. Using this method is an advantage if the HMC, VIOS, and the partition administrator roles are shared.

Since the HMC does not offer VLAN tagging, it is recommend that you put the HMC behind a firewall. For more information about extending VLAN into virtual networks, see 3.6, “Extending VLANs into virtual networks” on page 64.

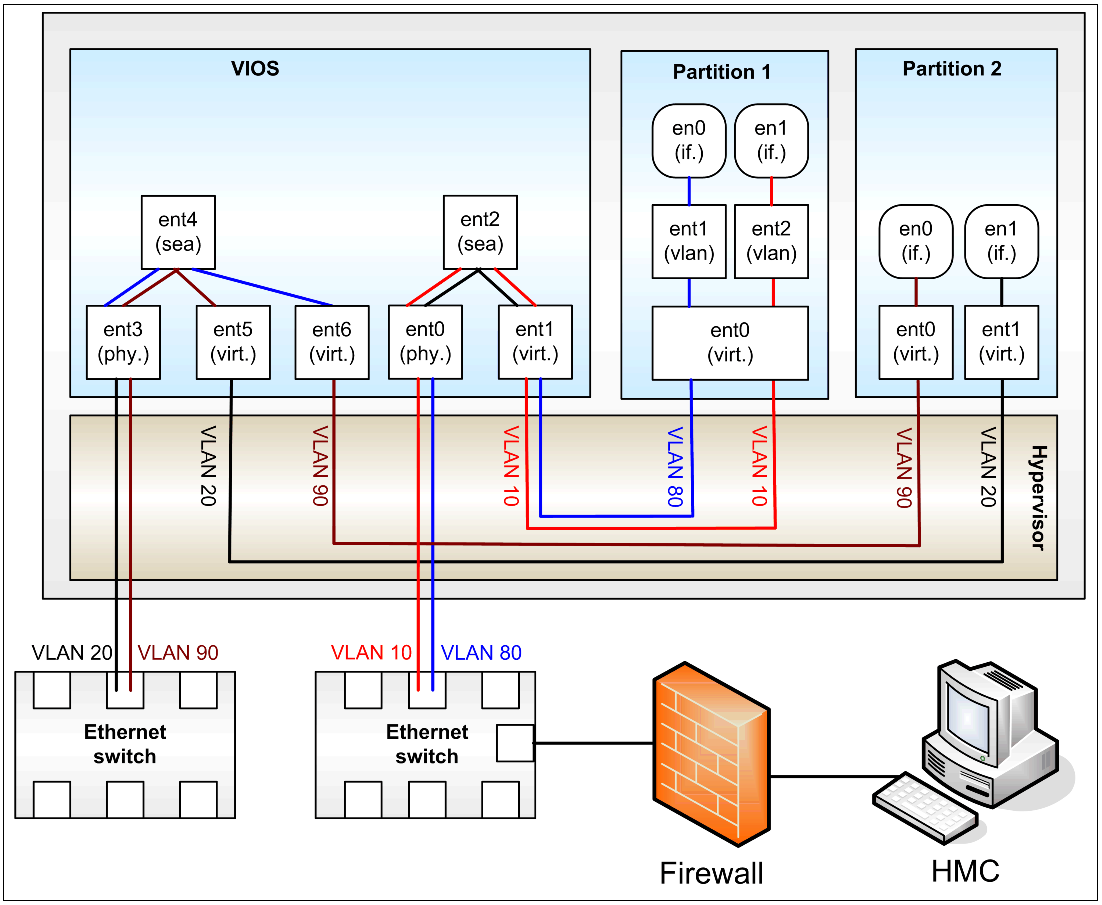


Figure 3-6 HMC in a VLAN tagged environment

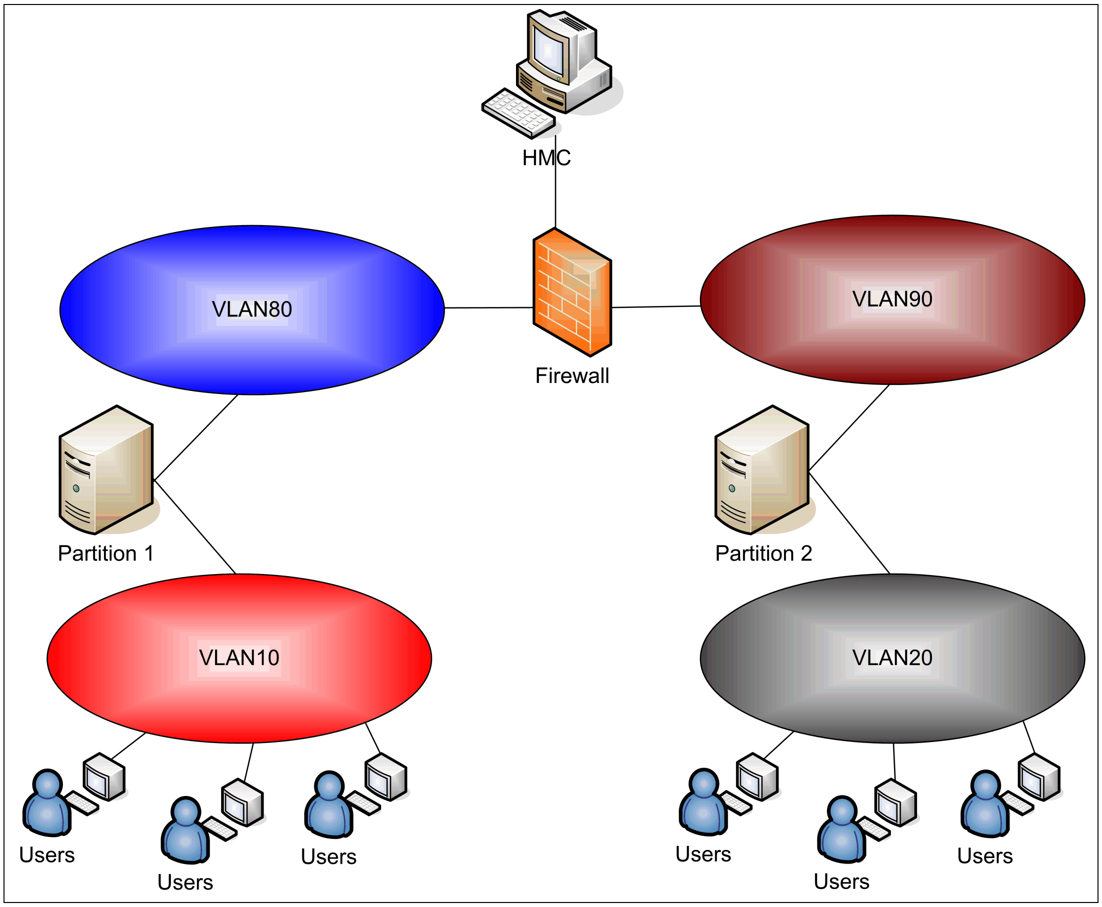


Figure 3-7 Cross-network VLAN tagging with a single HMC

3.5 Enabling network port security and the Virtual I/O Server

One of the options available with a number of today's network appliances is the ability to lock down a port to a single or set of Media Access Control (MAC) addresses. Using the SEA functions, you can provide network connectivity for many logical partitions with virtual network adapters. If you want to combine port security with the Virtual I/O Server, determine which MAC addresses you have to lock down on your physical network ports.

If you have an IP address assigned to the SEA within the Virtual I/O Server, this is the first place that network traffic can come from and the MAC address of the SEA. It should be added to the list. Example 3-1 on page 63 shows how to find the MAC address.

Example 3-1 Finding the MAC address of the SEA on a Virtual I/O Server

```
$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                 ent2
Backing device      ent0
Physloc             U787B.001.DNW108F-P1-C1-T1

$ entstat ent2 |grep Hardware
```

```
Hardware Address: 00:02:55:d3:dd:00
```

The SEA acts as a layer two bridge, so if there is no IP address on the SEA, there will be no need to lock down the SEA MAC address.

For any packets bridged by the SEA, all MAC addresses will be preserved (the MAC addresses of the virtual Ethernet adapter in the AIX 5L or Linux logical partition will be the MAC address seen in the network packet). For each AIX 5L or Linux logical partition with a virtual Ethernet adapter whose traffic will be bridged by the SEA, you must also add the MAC addresses of the virtual Ethernet adapter.

If you are using link aggregation on the Virtual I/O Server for the physical network adapters or Network Interface Backup to provide virtual network redundancy, ensure that the correct MAC address is used. In most cases, the MAC address from the first adapter specified is used, but a user-defined MAC address can be used in these cases (this alternative MAC address is also valid for regular physical or virtual adapters). This can be different from the default MAC addresses of the adapter cards or virtual Ethernet adapters.

3.6 Extending VLANs into virtual networks

The use of VLAN technology provides a 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. This section assumes that you understand the concepts of VLANs. We discuss in detail the recommended method of configuring and setting up a VLAN using VLAN tagging.

3.6.1 Scenario for setting up a VLAN

In a Virtual I/O Server environment, it is the Virtual I/O Server that provides the link between the internal virtual and external physical LANs. This can introduce an increased level of complexity due to multiple VLANs within a server that needs to be connected to multiple VLANs outside the server in a secure manner. The Virtual I/O Server, therefore, needs to be connected to all of the VLANs but must not allow packets to move between the VLANs.

In this scenario, as shown in Figure 3-8 on page 65, the following requirements must be met:

- All client partitions must be able to communicate with other client partitions on the same VLAN.
- All client partitions must be able to communicate to a single virtual Ethernet adapter in the Virtual I/O Server. This is achieved by using IEEE 802.1Q on the Virtual I/O Servers virtual Ethernet adapter to allow more than one virtual LAN ID to be accepted at the virtual Ethernet adapter.
- Ensure that the VLAN tags are not stripped from arriving packets from the client partitions. Stripped VLAN tags would result in the Virtual I/O Server not being able to forward the packets to the correct external VLAN.
- Enable the Shared Ethernet Adapter (SEA) to allow packets from multiple VLANs.

Figure 3-8 shows four partitions and a single Virtual I/O Server. In this example the VLAN interfaces are defined on an SEA adapter so that the Virtual I/O Server can communicate on these VLANs.

The following virtual LAN IDs are used:

- 10
- 20
- 30

In addition, the default virtual LAN ID 199 is also used. This default virtual LAN ID must be unique and not used by any clients in the network or physical Ethernet switch ports. For further details, see 3.6.4, “Ensuring VLAN tags are not stripped on the Virtual I/O Server” on page 68.

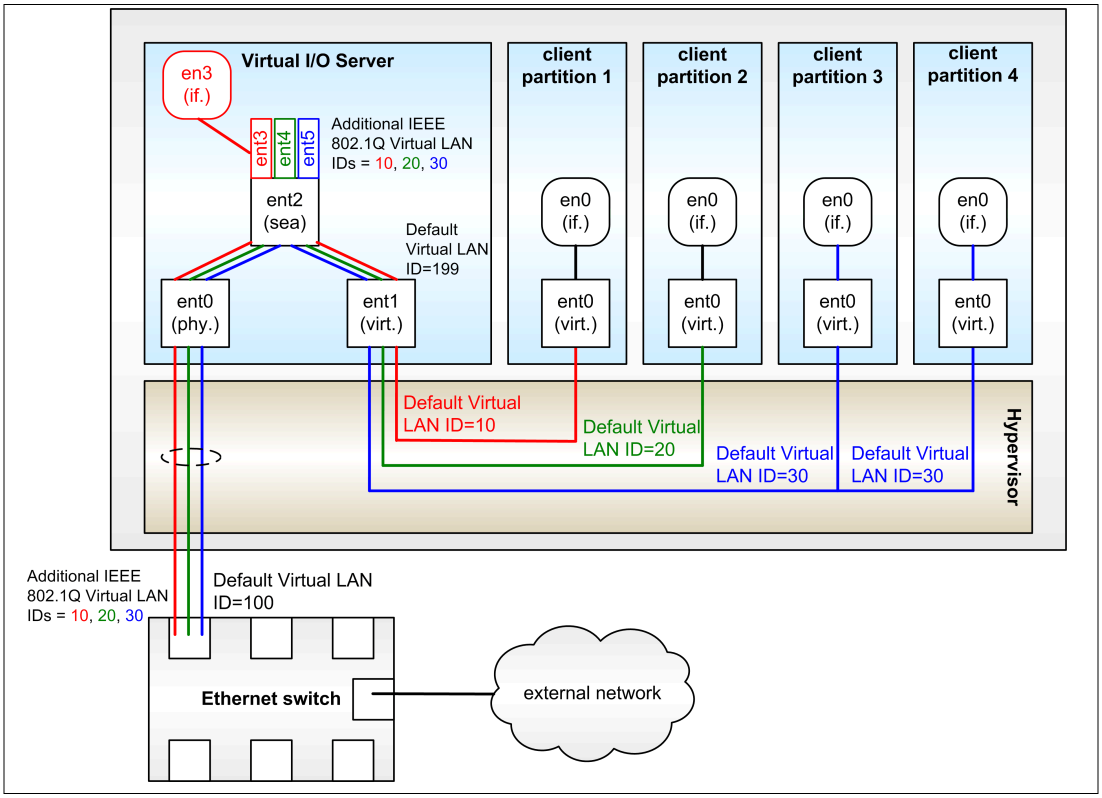

Virtual I/O Server
en3 (if.)
ent3
ent4
ent5
Additional IEEE 802.1Q Virtual LAN IDs = 10, 20, 30
ent2 (sea)
Default Virtual LAN ID=199
ent0 (phy.)
ent1 (virt.)
client partition 1
client partition 2
client partition 3
client partition 4
en0 (if.)
ent0 (virt.)
Default Virtual LAN ID=10
Default Virtual LAN ID=20
Default Virtual LAN ID=30
Default Virtual LAN ID=30
Hypervisor
Additional IEEE 802.1Q Virtual LAN IDs = 10, 20, 30
Default Virtual LAN ID=100
Ethernet switch
external network

Figure 3-8 VLAN configuration scenario

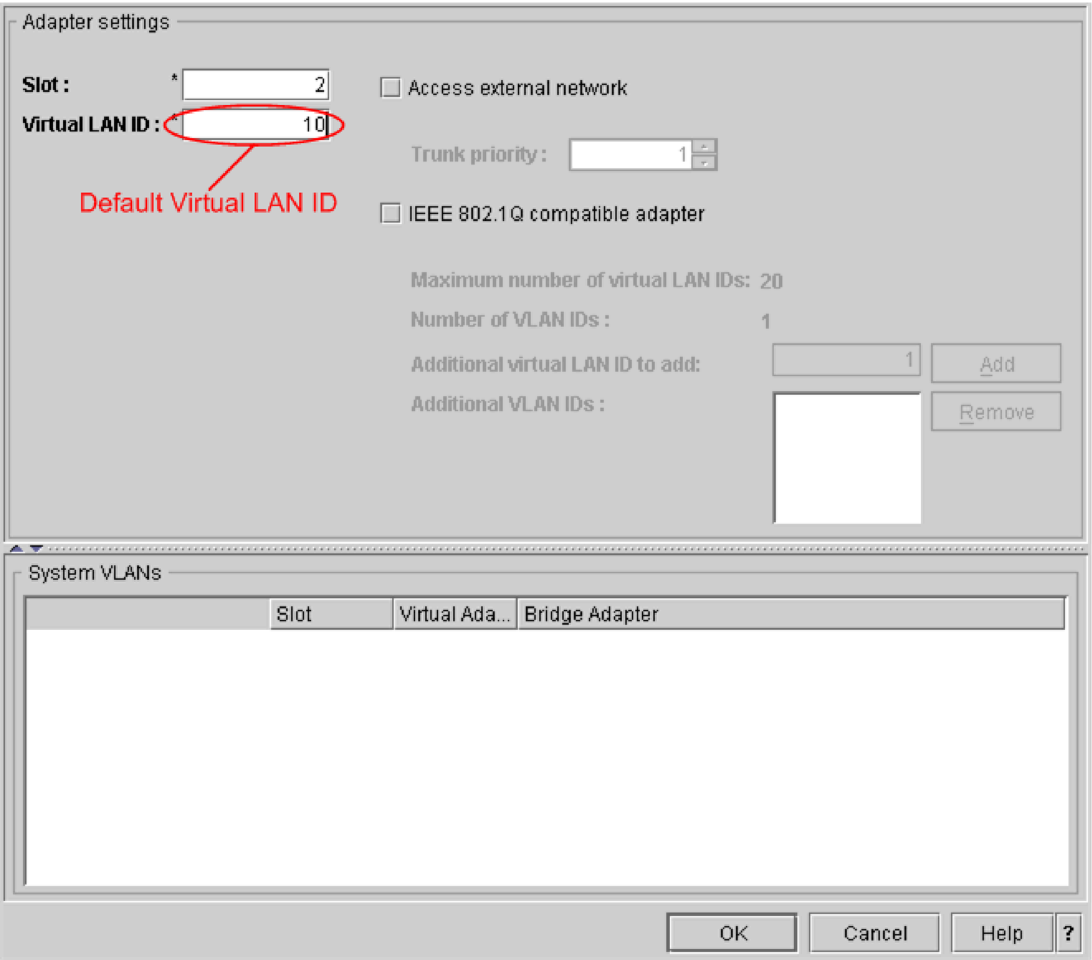
Note: Not all physical network switches support VLAN tagging. If you want to extend VLAN tagging outside your virtual network to your physical network, your physical network must also support VLAN tagging. It is a common error to forget to configure your physical network to match your virtual network VLAN settings.

3.6.2 Setting up the internal VLAN

As shown in Figure 3-8 on page 65, the configuration and setup of the internal VLAN is a straightforward process. Using the scenario shown in Figure 3-8 on page 65, perform the following steps on each client partition to configure the internal VLAN:

1. Log on to the HMC.
2. Begin creating a partition.

3. In the virtual Ethernet adapter window, assign each client partition a virtual Ethernet adapter with a default virtual LAN ID, as shown in Figure 3-8 on page 65. Figure 3-9 shows an example of creating a virtual Ethernet adapter for client partition 1.
4. Make sure both the Access external network and IEEE 802.1Q compatible adapter flags are cleared.
5. Finalize the partition creation.



Adapter settings

Slot : * 2

Virtual LAN ID : * 10

☐ Access external network

Trunk priority : 1

☐ IEEE 802.1Q compatible adapter

Maximum number of virtual LAN IDs: 20

Number of VLAN IDs : 1

Additional virtual LAN ID to add: 1

Additional VLAN IDs :

System VLANs

	Slot	Virtual Ada...	Bridge Adapter

Figure 3-9 Virtual Ethernet configuration for client partition using the HMC

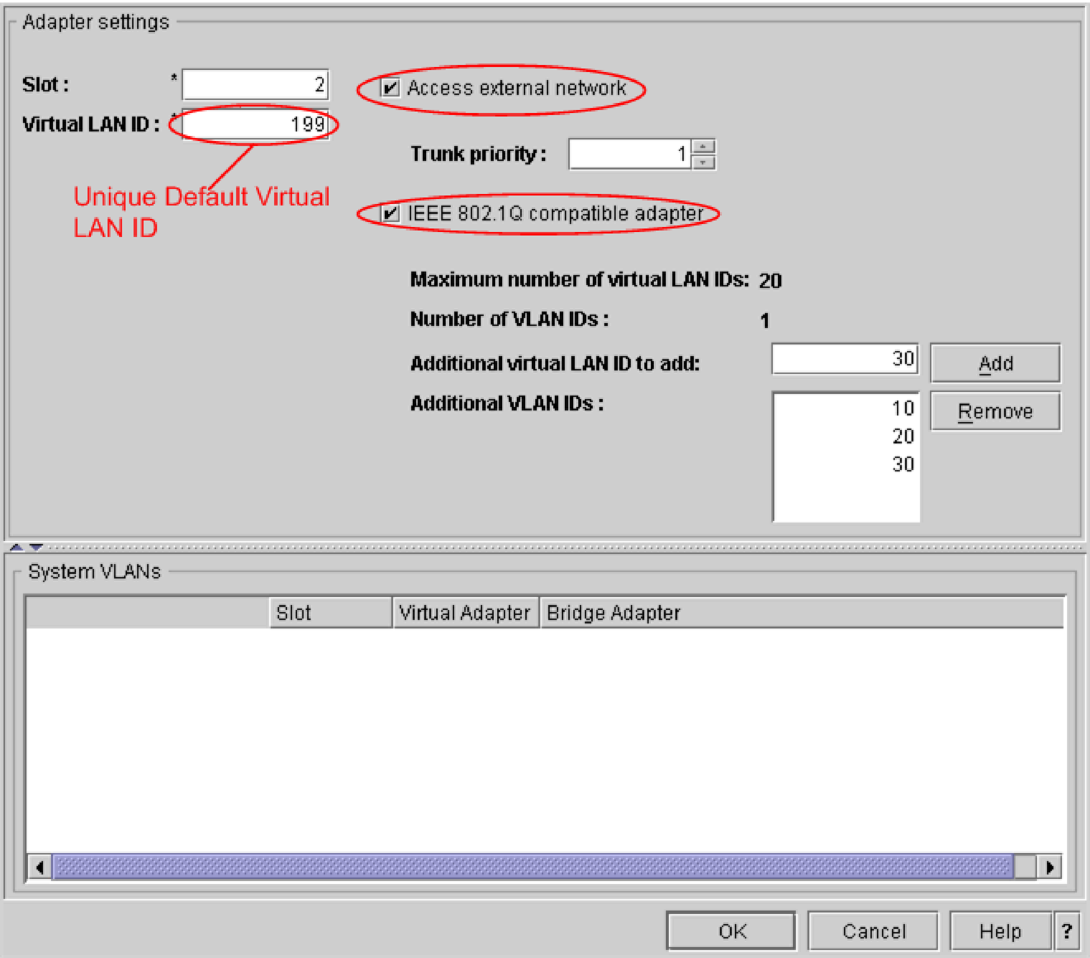
After the configuration completes, the virtual LAN ID on the virtual Ethernet adapter will be added to the packets from the client partition. This ID is used to route packets to the correct partition and then strip them off before delivery. For example, as shown in Figure 3-8 on page 65, client partition 3 and client partition 4 have the same virtual LAN ID, and thus are able to communicate directly. Client partition 1 and client partition 2 have different virtual LAN IDs, and thus are unable to communicate with client partition 3 and client partition 4 or each other, but are ready to communicate with other machines on VLANs external to the machine.

3.6.3 Configuring the virtual Ethernet adapter on the Virtual I/O Server

The Virtual I/O Server is a single relay point between multiple internal VLANs and the external physical Ethernet adapter or adapters and LAN. To bridge the virtual and physical networks, the Shared Ethernet Adapter (SEA) device is used to link one or more virtual adapters to the physical adapter. Using the scenario in Figure 3-8 on page 65, we implement a single virtual Ethernet adapter connecting to multiple internal VLANs.

To correctly set up this configuration, the virtual Ethernet adapter that is created in the Virtual I/O Server needs to name all the internal VLANs that are connected, as shown Figure 3-10. Perform the following steps:

1. Log on to the HMC.
2. In the virtual Ethernet adapter window, assign the Virtual I/O Server a virtual Ethernet adapter with a unique default virtual LAN ID. This ID must not be used by any client partition or physical network. In this scenario, we use the virtual LAN ID 199.
3. Select the **IEEE 802.1Q compatible adapter** flag.
4. Add additional virtual LAN IDs associated with the client partitions. In this scenario, the additional virtual LAN IDs are `10`, `20`, `30`.
5. Make sure that the **Access external network** flag is selected. Leave the Trunk priority as the default unless using a SEA Failover configuration.



Slot : * 2

Virtual LAN ID : * 199

☑ Access external network

Trunk priority : 1

☑ IEEE 802.1Q compatible adapter

Maximum number of virtual LAN IDs: 20

Number of VLAN IDs : 1

Additional virtual LAN ID to add: 30

Additional VLAN IDs : 10 20 30

	Slot	Virtual Adapter	Bridge Adapter

Figure 3-10 VIrtual Ethernet configuration for Virtual I/O Server using the HMC

In this instance, the Access external network flag is selected because this Virtual I/O Server is using the Shared Ethernet Adapter function to transfer packets to and from the external network. The IEEE 802.1Q compatible adapter flag is selected to allow the Shared Ethernet Adapter to transmit packets on additional virtual LAN IDs. These additional virtual LAN IDs are the same as the virtual LAN IDs configured in the client partitions; therefore, packets that arrive from the client partition with the VLAN tag added by the hypervisor will be allowed to pass through the Shared Ethernet Adapter.

3.6.4 Ensuring VLAN tags are not stripped on the Virtual I/O Server

It is important that the virtual LAN ID added to the packets leaving the client partitions (the default virtual LAN ID number) are not removed on entering the Virtual I/O Server. This will happen when a default configuration is used.

It is for this reason that the default virtual LAN ID on the Virtual I/O Servers virtual Ethernet adapter, as shown in Figure 3-10 on page 67, must be set to an unused virtual LAN ID (in this scenario, 199).

If a packet arrives with this virtual LAN ID (199), the virtual LAN ID tag would be stripped off (untagged) and could not then be forwarded on to the correct external VLAN. If this was sent through the Shared Ethernet Adapter to the external physical network, it would arrive at the Ethernet switch as untagged. The resulting outcome depends on the settings on the physical Ethernet switch. The packet might be discarded or sent on to a default VLAN, but the chances of it going to virtual LAN ID 199 are remote unless the network administrator has explicitly set this up (for example, the Ethernet switch port has a default virtual LAN ID of 199).

3.6.5 Configuring the Shared Ethernet Adapter to access multiple VLANs

In this scenario, use the **mkvdev -sea** command on the Virtual I/O Server that takes the virtual Ethernet adapter and physical Ethernet adapter to create the Shared Ethernet Adapter. To configure the Shared Ethernet Adapter and access the multiple VLANs, perform the following steps:

1. Use the **mkvdev** command on the Virtual I/O Server to create the Shared Ethernet Adapter. In this scenario, the virtual Ethernet adapter ent1 and physical Ethernet adapter ent0 are used to create the new Shared Ethernet Adapter ent2.

   ```
   $ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 199
   ent2 Available
   en2
   et2
   ```

   When creating the Shared Ethernet Adapter, the `-default ent1` option is the default internal virtual VLAN to send the packet onto if the packet is untagged. In this scenario, there is only one virtual Ethernet adapter, but this would be used in the more complex case of multiple virtual Ethernet adapters specified in the command. Also, the `-defaultid 199` option is the virtual LAN ID to use for untagged packets (in effect, this is the default virtual LAN ID of the SEA). We are not expecting untagged packets, and by using an unused number like this, these packets are not delivered because no client will accept VLAN 199 packets. Running the **lsdev** command displays the newly created SEA:

   ```
   $ 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  Shared Ethernet Adapter
   ide0          Available  ATA/IDE Controller Device
   sisioa0       Available  PCI-X Dual Channel U320 SCSI RAID Adapter
   vhost0        Available  Virtual SCSI Server Adapter
   vsa0          Available  LPAR Virtual Serial Adapter
   ```

2. Using the **mkvdev -vlan** command, configure the newly created Shared Ethernet Adapter (ent2) to allow packets from the additional virtual LAN IDs (10, 20, 30):

   ```
   $ mkvdev -vlan ent2 -tagid 10
   ent3 Available
   en3
   et3
   $ mkvdev -vlan ent2 -tagid 20
   ent4 Available
   en4
   et4
   $ mkvdev -vlan ent2 -tagid 30
   ent5 Available
   en5
   et5
   ```

   Note that this creates a new Ethernet adapter for each of the previous commands.

3. Using the **mktcpip** command, configure an IP address on one of the new VLANs to allow administrators network access to the Virtual I/O Server. In this scenario, as shown in Figure 3-8 on page 65, we use en3, which resides on VLAN 10, to configure an IP address for Virtual I/O Server network connectivity.

Note: Typically, the IP address is only required on the Virtual I/O Server for administrative purposes. As such, configure the IP address on the management VLAN only to allow administrators network access to the Virtual I/O Server.

3.7 Link aggregation on the Virtual I/O Server

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 limitation of a single network adapter and avoid bottlenecks when sharing one network adapter among many client partitions. For more information about the different types of link aggregation technologies, refer to *Advanced POWER Virtualization on IBM System p5*, SG24-7940. This section discusses the use of a link aggregation device as part of a Shared Ethernet Adapter.

3.7.1 Creating the link aggregation

Figure 3-11 on page 70 shows a single Virtual I/O Server configuration using link aggregation of two physical Ethernet adapters and a backup adapter.

Note: We recommended using the backup adapter option on a single Virtual I/O Server configuration only. For dual Virtual I/O Servers configurations, the link aggregation without the backup adapter should be implemented because the SEA Failover feature will provide the necessary failover capability.

It is possible that during an SEA Failover, up to 15-30 packets are lost while the network reroutes the traffic.

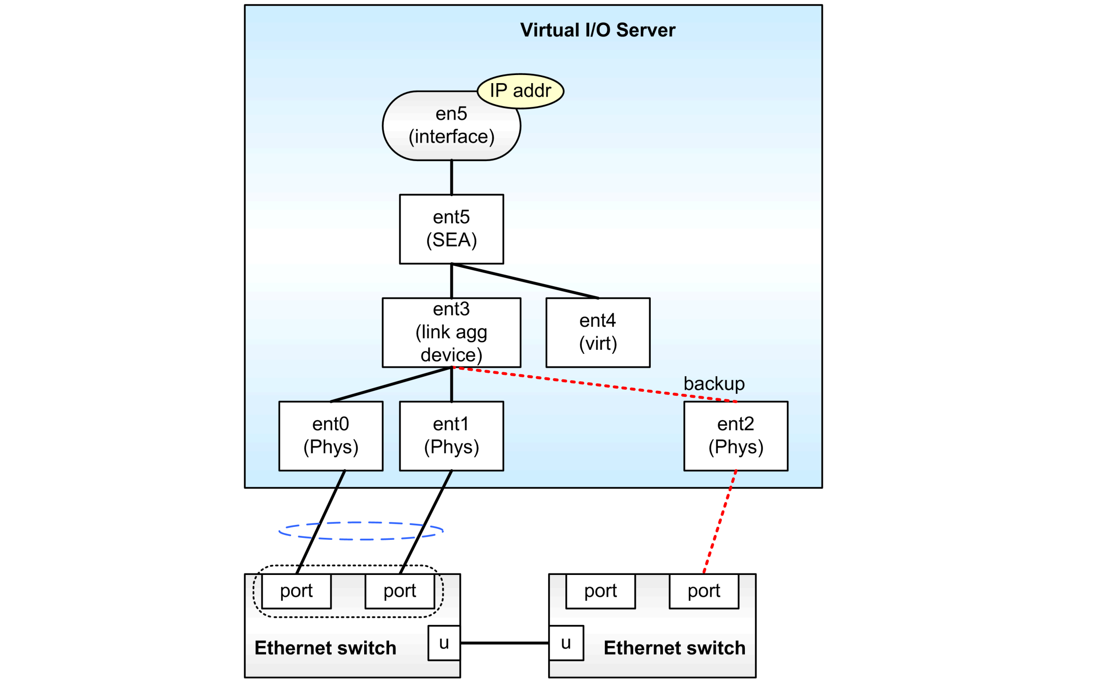


Figure 3-11 Link aggregation using a single Virtual I/O Server

Figure 3-11 shows a single Virtual I/O Server configuration using link aggregation of two physical Ethernet adapters and a backup adapter and the text that follows outlines the steps that are required to configure this:

1. Using the example shown in Figure 3-11, use the following physical Ethernet adapters to configure the link aggregation:
   - ent0
   - ent1
   - ent2

   ```
   $ lsdev -type adapter
   name            status                                    description
   ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
   ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
   ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
   ide0            Available  ATA/IDE Controller Device
   sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
   sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
   vhost0          Available  Virtual SCSI Server Adapter
   vsa0            Available  LPAR Virtual Serial Adapter
   ```

   > **Note:** All network adapters that form the link aggregation (not including the backup adapter) must be connected to the same network switch.

2. Set the attributes required, such as media speed and jumbo frames, to the desired setting for each adapter in the link aggregation. All adapters in the link aggregation should have the same attribute settings.

   ```
   $ chdev -dev ent0 -attr media_speed=Auto_Negotiation jumbo_frames=yes
   ent0 changed
   ```

3. Confirm that the attributes have taken effect:

```
$ lsdev -dev ent0 -attr
attribute     value          description
user_settable

alt_addr       0x000000000000   Alternate ethernet address                     True
busintr        88               Bus interrupt level                            False
busmem         0xc0100000       Bus memory address                             False
chksum_offload yes              Enable hardware transmit and receive checksum  True
compat_mode    no               Gigabit Backward compatability                 True
copy_bytes     2048             Copy packet if this many or less bytes         True
delay_open     no               Enable delay of open until link state is known True
failback       yes              Enable auto failback to primary                True
failback_delay 15               Failback to primary delay timer                True
failover       disable          Enable failover mode                           True
flow_ctrl      yes              Enable Transmit and Receive Flow Control       True
intr_priority  3                Interrupt priority                             False
intr_rate      10000            Max rate of interrupts generated by adapter    True
jumbo_frames   yes              Transmit jumbo frames                          True
large_send     yes              Enable hardware TX TCP resegmentation          True
media_speed    Auto_Negotiation Media speed                                    True
rom_mem        0xc0040000       ROM memory address                             False
rx_hog         1000             Max rcv buffers processed per rcv interrupt    True
rxbuf_pool_sz  2048             Rcv buffer pool, make 2X rxdesc_que_sz         True
rxdesc_que_sz  1024             Rcv descriptor queue size                      True
slih_hog       10               Max Interrupt events processed per interrupt   True
tx_que_sz      8192             Software transmit queue size                   True
txdesc_que_sz  512              TX descriptor queue size                       True
use_alt_addr   no               Enable alternate ethernet address              True
```

Note: When using the Network Backup Interface option, we recommend connecting the backup adapter to a separate switch, as highlighted in Figure 3-11 on page 70, to provide network switch redundancy in the event the primary network switch becomes unavailable.

4. On the Virtual I/O Server, run the **mkvdev -lnagg** command to create the link aggregation device. Because this is a single Virtual I/O Server configuration, the network backup option will be configured using ent2 as the backup network device.

```
$ mkvdev -lnagg ent0 ent1 -attr backup_adapter=ent2
ent3 Available
en3
et3
```

Note: If one of the network interfaces (en0 or en1) on the client partition already has been configured as part of a NIM installation, it is necessary to take the interface down in addition to removing it so that it is not configured before the Network Interface Backup interface is configured.

Use the **-attr** flag to specify the type of link aggregation required. The attributes available are standard, round_robin, or 8023ad. The correct setting depends on the configuration of the physical network switch. Most commonly, the standard option is selected; however, check with your network administrator to determine what type of link aggregation is configured on the network switch. In addition, you can also use the **netaddr** flag to specify an address to ping for the backup adapter.

Note: For dual Virtual I/O Servers configurations, the link aggregation without the backup adapter should be implemented because the SEA Failover feature will provide the necessary failover capability.

5. List the EtherChannel device using the **lsdev** command:

```
$ lsdev -type adapter
name            status                                      description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
$ lsdev -dev ent3 -attr
attribute       value          description
user_settable

adapter_names   ent0,ent1      EtherChannel Adapters                       True
alt_addr        0x000000000000 Alternate EtherChannel Address              True
auto_recovery   yes            Enable automatic recovery after failover    True
backup_adapter  ent2           Adapter used when whole channel fails       True
hash_mode       default        Determines how outgoing adapter is chosen   True
mode            standard       EtherChannel mode of operation              True
netaddr         0              Address to ping                             True
noloss_failover yes            Enable lossless failover after ping failure True
num_retries     3              Times to retry ping before failing          True
retry_time      1              Wait time (in seconds) between pings        True
use_alt_addr    no             Enable Alternate EtherChannel Address       True
use_jumbo_frame yes            Enable Gigabit Ethernet Jumbo Frames        True
```

6. Using the HMC, add a new virtual Ethernet adapter to the Virtual I/O Server. Make sure the **Access external network** flag is selected. Refer to 2.2.2, “Dynamic LPAR assignments” on page 38 for assistance in using dynamic LPAR.

```
$ lsdev -type adapter
name            status                                      description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

7. Configure a Shared Ethernet Adapter device using the newly created ent3 link aggregation device and the ent4 virtual Ethernet adapter:

```
$ mkvdev -sea ent3 -vadapter ent4 -default ent4 -defaultid 2
ent5 Available
en5
```

```
et5
$ lsdev -type adapter
name            status                                                description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ent5            Available  Shared Ethernet Adapter
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

3.7.2 Adding a physical Ethernet adapter to an existing link aggregation

To increase network bandwidth requirements for client partitions using the Shared Ethernet Adapter in the Virtual I/O Server, configure additional physical adapters to an existing link aggregation device in a dynamic manner. Using the example shown in Figure 3-11 on page 70, to add a new adapter to an existing link aggregation, perform the following steps:

1. Using the HMC, add a new physical Ethernet adapter to the Virtual I/O Server. Refer to 2.2.2, “Dynamic LPAR assignments” on page 38 for assistance in using dynamic LPAR.
2. Run the **lsdev** command to confirm that the new adapter is configured on the Virtual I/O Server:

```
$ lsdev -type adapter
name            status                                                description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ent5            Available  Shared Ethernet Adapter
ent6            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

3. Using the **chdev** and **lsdev** commands, confirm and change the attributes of the new adapter (if required) to be the same as the physical Ethernet adapters in the existing link aggregation (for example, media_speed, jumbo_frames), as follows:

```
$ lsdev -dev ent6 -attr
attribute      value            description
user_settable

alt_addr       0x000000000000   Alternate ethernet address                      True
busintr        88               Bus interrupt level                             False
busmem         0xc0100000       Bus memory address                              False
chksum_offload yes              Enable hardware transmit and receive checksum  True
compat_mode    no               Gigabit Backward compatability                  True
copy_bytes     2048             Copy packet if this many or less bytes          True
delay_open     no               Enable delay of open until link state is known True
failback       yes              Enable auto failback to primary                 True
```

```
failback_delay 15               Failback to primary delay timer              True
failover       disable          Enable failover mode                         True
flow_ctrl      yes              Enable Transmit and Receive Flow Control     True
intr_priority  3                Interrupt priority                           False
intr_rate      10000            Max rate of interrupts generated by adapter  True
jumbo_frames   yes               Transmit jumbo frames                        True
large_send     yes              Enable hardware TX TCP resegmentation        True
media_speed    Auto_Negotiation Media speed                                  True
rom_mem        0xc0040000       ROM memory address                           False
rx_hog         1000             Max rcv buffers processed per rcv interrupt  True
rxbuf_pool_sz  2048             Rcv buffer pool, make 2X rxdesc_que_sz       True
rxdesc_que_sz  1024             Rcv descriptor queue size                    True
slih_hog       10               Max Interrupt events processed per interrupt True
tx_que_sz      8192             Software transmit queue size                 True
txdesc_que_sz  512              TX descriptor queue size                     True
use_alt_addr   no               Enable alternate ethernet address            True
```

4. Log in to the **oem_setup_env** shell and use the **ethchan_config** command to configure the new adapter to the existing link aggregation. Then, exit the **oem_setup_env** shell. In future updates of the Virtual I/O Server, the use of the **oem_setup_env** command might be removed and replaced with a new command option or command.

```
$ oem_setup_env
# ethchan_config -a -p ent5 ent3 ent6
# exit
```

5. Using the **lsdev** command, confirm that the new adapter is part of the link aggregation:

```
$ lsdev -dev ent3 -attr
attribute       value          description
user_
settable

adapter_names   ent0,ent1,ent6 EtherChannel Adapters                        True
alt_addr        0x000000000000 Alternate EtherChannel Address               True
auto_recovery   yes            Enable automatic recovery after failover     True
backup_adapter  ent2           Adapter used when whole channel fails        True
hash_mode       default        Determines how outgoing adapter is chosen    True
mode            standard       EtherChannel mode of operation               True
netaddr         0              Address to ping                              True
noloss_failover yes            Enable lossless failover after ping failure True
num_retries     3              Times to retry ping before failing           True
retry_time      1              Wait time (in seconds) between pings         True
use_alt_addr    no             Enable Alternate EtherChannel Address        True
use_jumbo_frame no              Enable Gigabit Ethernet Jumbo Frames         True
```

To add a backup adapter, use the **ethchan_config** command with the **-b** flag:

```
#ethchan_config -a -b -p <SEA device> <EtherChannel device> <adapter to add as
backup>
```

To remove an adapter, use the **ethchan_config** command replacing the **-a** flag with the **-d** flag:

```
#ethchan_config -a -p <SEA device> <EtherChannel device> <adapter to add as
primary>
```

Important: When adding new adapters to a link aggregation, make sure that the ports on the physical network switch have also been configured to support the new adapters in the link aggregation.

3.8 Network availability options

In this section, we discuss the various options and advantages when using different types of network availability configurations in a virtualized environment.

3.8.1 Single Virtual I/O Server configuration high availability option

For configurations that use a single Virtual I/O Server, network availability can be increased by using the Network Interface Backup feature on the Virtual I/O Server. This setup provides physical Ethernet adapter redundancy for all client partitions using the Shared Ethernet Adapter. This type of configuration will protect the client partition if either a physical adapter or physical network switch become unavailable. For information about how to configure Network Interface Backup on the Virtual I/O Server, refer to 3.7, “Link aggregation on the Virtual I/O Server” on page 69.

3.8.2 Dual Virtual I/O Servers enhanced availability options

Dual Virtual I/O Servers configurations provide two common options for increasing network availability:

- The first is the implementation of the Network Backup Interface (NIB) feature on the client partitions.
- The second, which requires no additional configuration on the client partitions, is the Shared Ethernet Adapter Failover feature introduced in Virtual I/O Server V1.2.

This section describes when to use each feature.

Advantages of Network Interface Backup

NIB can provide better resource utilization as a result of the following:

- With Shared Ethernet Adapter Failover, only one of the two Shared Ethernet Adapters is actively used at any time, while the other Shared Ethernet Adapter is only a standby. Therefore, the bandwidth of the physical Ethernet adapters of the standby Shared Ethernet Adapter is not used.
- With NIB, you can distribute the clients over both Shared Ethernet Adapters in such a way that half of them will use the Shared Ethernet Adapter on the first Virtual I/O Server as the primary adapter, and the other half will use the Shared Ethernet Adapter on the second Virtual I/O Server as the primary adapter. This enables the bandwidth of the physical Ethernet adapters in both the Virtual I/O Servers Shared Ethernet Adapters to be used concurrently by different client partitions. You are able to do this using additional pairs for additional VLANS and also requiring additional hardware.

When to use Network Interface Backup

Using Network Interface Backup might be suitable over the Shared Ethernet Adapter Failover option when:

- You have an existing Network Interface Backup configuration, are upgrading from Virtual I/O Server Version 1.1, and do not want to change the Ethernet setup.
- You are only running AIX 5L operating system-based partitions.
- You are not using VLAN tagging.
- You want to load balance client partitions to use both Virtual I/O Servers.

Advantages of Shared Ethernet Adapter Failover

The Shared Ethernet Adapter Failover provides the following advantages over Network Interface Backup:

- Shared Ethernet Adapter Failover is implemented on the Virtual I/O Server, which simplifies the virtual network administration.
- The client partitions only require a single virtual Ethernet adapter and VLAN with no fail-over logic implemented, making the configuration of clients easier.
- When using the Network Interface Backup approach, the client partition configuration is more complex because all clients have to configure a second virtual Ethernet adapter on a different VLAN and a link aggregation adapter with the NIB feature.
- Shared Ethernet Adapter Failover has the added support of IEEE 802.1Q VLAN tagging.
- SEA Failover simplifies NIM installation because only a single virtual Ethernet device is required on the client partition. The Ethernet configuration does not need to be modified after a NIM installation.
- A single ping is needed; while on NIB, each client must perform a ping.

When to use SEA Failover

We recommend using Shared Ethernet Adapter Failover rather than the Network Interface Backup option for the following:

- You use VLAN tagging.
- You are running the Linux operating system.
- You do not need load balancing per Shared Ethernet Adapter between the primary and standby Virtual I/O Servers.

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.

3.9 Creating Shared Ethernet Adapters to support SEA Failover

For many clients, dual Virtual I/O Servers provide the maximum flexibility by allowing required functions such as Virtual I/O Server maintenance to be performed on each Virtual I/O Server in turn to minimize client interruption. To provide the network resilience (3.8, “Network availability options” on page 75), we can use the SEA Failover method. This method extends the existing SEA device and includes (at least) two new parameters over and above an SEA configured in a single Virtual I/O Server setup:

`ctl_chan` Virtual Ethernet adapter used to communicate status to the SEA in the other Virtual I/O Server such that between them the Virtual I/O Servers can decide which SEA is providing the bridging from the virtual to physical networks.

`ha_mode` The mode for the SEA to function in. The auto setting is the most common, but other settings, such as standby, can be used to force SEA Failover.

When creating these Shared Ethernet Adapters in the dual Virtual I/O Servers setup, it is very important that the control channel and high availability mode are specified at creation time. Without these, the Shared Ethernet Adapters do not know about each other, and both can start bridging network traffic from the virtual network to the physical network. This creates a loop in the network that can cause a network interruption. The diagram in Figure 3-12 on page 77 shows how the SEA Failover is set up.

Important: If you do not define the **`ctl_chan`** attribute to the Shared Ethernet Adapters, they will not be able to negotiate which one is providing the bridging functionality. At this point, both SEAs will be bridging and a spanning tree loop can be formed. To avoid this, always specify the these parameters with the SEA creation.

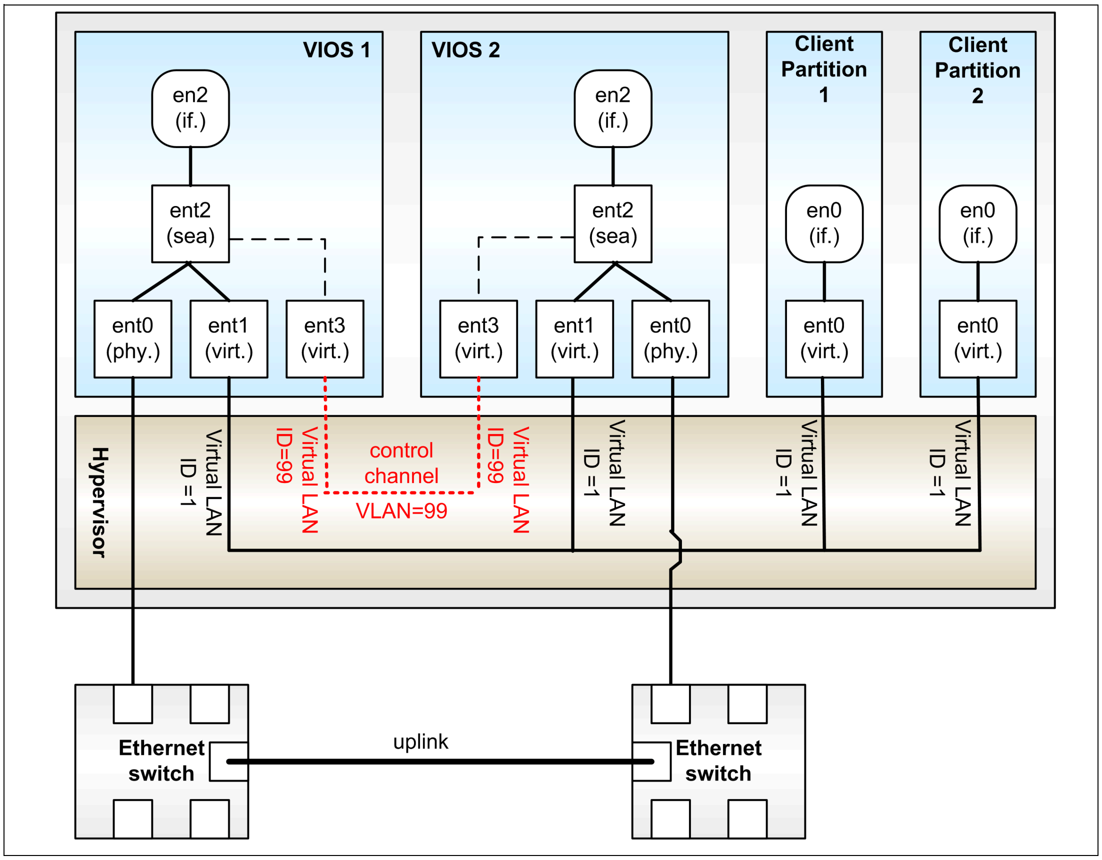


Figure 3-12 Diagram of SEA Failover setup

When creating these SEAs in an SEA Failover pair, it is important to specify the **`ctl_chan`** and **`ha_mode`** attributes so that the SEA option read as the following:

```
$mkvdev  -sea TargetDevice -vadapter VirtualEthernetAdapter ...
               -default DefaultVirtualEthernetAdapter
               -defaultid SEADefaultPVID [-attr Attributes=Value ...]
               [-migrate]
$mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid=10 -attr ha_mode=auto
ctl_chan=ent2
```

When creating an SEA in a single Virtual I/O Server, the command used is similar to:

```
$mkvdev  -sea TargetDevice -vadapter VirtualEthernetAdapter ...
               -default DefaultVirtualEthernetAdapter
               -defaultid SEADefaultPVID [-attr Attributes=Value ...]
               [-migrate]
$mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid=10
```

3.10 SEA threading on the Virtual I/O Server

The Virtual I/O Server enables you to virtualize both disk and network traffic for AIX 5L and Linux operating system-based clients. The main difference between these types of traffic is the persistence of them. If the Virtual I/O Server has to move network data around, it must do this immediately because network data has no persistent storage. For this reason, the network services provided by the Virtual I/O Server (such as the Shared Ethernet Adapter) run with the highest priority. Disk data for virtual SCSI devices is run at a lower priority than the network because the data is stored on the disk and there is less of a danger of losing it due to time outs. The devices are also normally slower in speed.

The thread attribute changes the SEA thread scheduling.

The shared Ethernet process of the Virtual I/O Server prior to Version 1.3 runs at the interrupt level that was optimized for high performance. With this approach, it ran with a higher priority than the virtual SCSI if there was very high network traffic. If the Virtual I/O Server did not provide enough CPU resource for both, the virtual SCSI performance could experience a degradation of service.

With Virtual I/O Server Version 1.3, the shared Ethernet function can be implemented using kernel threads. This enables a more even distribution of the processing power between virtual disk and network.

This threading can be turned on and off per Shared Ethernet Adapter by changing the thread attribute and can be changed while the SEA is operating without any interruption to service. A value of one indicates that threading is to be used and zero indicates the original interrupt method:

```
$ lsdev -dev ent2 -attr thread
value

0
$ chdev -dev ent2 -attr thread=1
ent2 changed
$ lsdev -dev ent2 -attr thread
value

1
```

The performance difference without threading works out at around 8% less (using our intensive test loads) CPU needed for the same network throughput; but with the burst nature of network traffic, we recommend enabling threading (this is now the default). By this, we mean that network traffic will come in spikes, as users log on or as Web pages load, for example. These spikes might coincide with disk access. For example, a user logs on to a system, generating a network activity spike, because during the logon process some form of password database stored on the disk will most likely be accessed or the user profile read.

The one scenario where you should consider disabling threading is where you have a Virtual I/O Server dedicated for network and another dedicated for disk. This is only recommended when mixing extreme disk and network loads together on a CPU constricted server.

As discussed in 5.4, “Sizing your Virtual I/O Server” on page 125, usually the network CPU requirements will be higher than disk. In addition, you will probably have the disk Virtual I/O Server setup to provide a network backup with SEA Failover if you want to remove the other Virtual I/O Server from the configuration for scheduled maintenance. In this case, you will have both disk and network running through the same Virtual I/O Server, so threading is recommended.

3.11 Jumbo frame and path MTU discovery

This section provides the information about maximum transfer unit (MTU) and how to use jumbo frames. We also describe the path MTU discovery changes in AIX 5L Version 5.3 and provide recommendations about virtual Ethernet tuning with path MTU discovery.

We discuss the following topics:

- Maximum transfer unit
- Path MTU discovery
- How to use jumbo frame and recommendations for virtual Ethernet

3.11.1 Maximum transfer unit

There is a limit on the frame size for Ethernet, IEEE 802.x local area networks, and other networks. The maximum length of an Ethernet frame is 1526 bytes, so it can support a data field length of up to 1500 bytes. In IEEE 802.3, the data field length depends on the transmission speeds. Table 3-1 provides typical maximum transmission units (MTUs).

Table 3-1 Typical maximum transmission units (MTUs)

Network	MTU (bytes)
Official maximum MTU	65535
Ethernet (10 or 100 MBps)	1500
Ethernet (gigabit)	9000
FDDI	4352
X.25	576
Official minimum MTU	68

If you send data through a network, and the data is larger than the network's MTU, it can become fragmented by breaking the data up into smaller pieces. Each fragment must be equal to or smaller than the MTU.

The MTU size can affect the network performance between source and target systems. The use of a large MTU sizes allows the operating system to send fewer packets of a larger size to reach the same network throughput. The larger packets reduce the processing required in the operating system, because each packet requires the same amount of processing. If the workload is only sending small messages, the larger MTU size will not help.

The maximum segment size (MSS) also is affected by the MTU size. The MSS is the largest data that the TCP layer can send to the destination IP. When a connection is established, each system can announce an MSS value. If one system does not receive an MSS from the other system, it uses the default MSS value.

In AIX 5L Version 5.2 or earlier, the default MSS value was 512 bytes, but AIX 5L Version 5.3 supports 1460 bytes as the default value. If you apply APAR IY57637 in AIX 5L Version 5.2, the default MSS value is changed to 1460 bytes.

The **`no -a`** command displays the value of the default MSS as `tcp_mssdflt`. You will receive the following information shown in Example 3-2 on page 80.

Example 3-2 The default MSS value in AIX 5L V5.3

```
# no -a | grep tcp
       tcp_bad_port_limit = 0
                  tcp_ecn = 0
       tcp_ephemeral_high = 65535
        tcp_ephemeral_low = 32768
             tcp_finwait2 = 1200
           tcp_icmpsecure = 0
          tcp_init_window = 0
    tcp_inpcb_hashtab_siz = 24499
              tcp_keepcnt = 8
             tcp_keepidle = 14400
             tcp_keepinit = 150
            tcp_keepintvl = 150
     tcp_limited_transmit = 1
              tcp_low_rto = 0
             tcp_maxburst = 0
              tcp_mssdflt = 1460
          tcp_nagle_limit = 65535
        tcp_nagleoverride = 0
               tcp_ndebug = 100
              tcp_newreno = 1
           tcp_nodelayack = 0
        tcp_pmtu_discover = 1
            tcp_recvspace = 16384
            tcp_sendspace = 16384
            tcp_tcpsecure = 0
             tcp_timewait = 1
                  tcp_ttl = 60
           tcprexmtthresh = 3
```

If the source network does not receive an MSS when the connection is first established, the system will use the default MSS value. Most network environments are Ethernet, and this can support at least a 1500 byte MTU.

For example, if you execute an FTP application when the MSS value is not received, in AIX 5L Version 5.2 or earlier, the application only uses 512 bytes MSS during the first connection because the default MSS value is 512 bytes, and this can cause degradation in performance. The MSS is negotiated for every connection, so the next connection can use a different MSS.

3.11.2 Path MTU discovery

Every network link has a maximum packet size described by the MTU, described in 3.11.1, “Maximum transfer unit” on page 79. The datagrams can be transferred from one system to another through many links with different MTU values. If the source and destination system have different MTU values, it can cause fragmentation or dropping of packets while the smallest MTU for the link is selected. The smallest MTU for all the links in a path is called the path MTU, and the process of determining the smallest MTU along the entire path from the source to the destination is called path MTU discovery (PMTUD).

With AIX 5L Version 5.2 or earlier, the Internet control MTU discovery (ICMP) echo request and ICMP echo reply packets are used to discover the path MTU using IPv4. The basic procedure is simple. When one system tries to optimize its transmissions by discovering the path MTU, it sends packets of its maximum size. If these do not fit through one of the links between the two systems, a notification from this link is sent back saying what the maximum size is this link will support. The notifications return an ICMP “Destination Unreachable” message to the source of the IP datagram, with a code indicating “fragmentation needed and DF set” (type 3, type 4).

When the source receives the ICMP message, it lowers the send MSS and tries again using this lower value. This is repeated until the maximum possible value for all of the link steps is found.

Possible outcomes during the path MTU discovery procedure include:

- Packet can get across all the links to the destination system without being fragmented.
- Source system can get an ICMP message from any hop along the path to the destination system, indicating that the MSS is too large and not supported by this link.

This ICMP echo request and reply procedure has a few considerations. Some system administrators do not use path MTU discovery because they believe that there is a risk of denial of service (DoS) attacks.

Also, if you already use the path MTU discovery, routers or firewalls can block the ICMP messages being returned to the source system. In this case, the source system does not have any messages from the network environment and sets the default MSS value, which might not be supported across all links.

The discovered MTU value is stored in the routing table using a cloning mechanism in AIX 5L Version 5.2 or earlier, so it cannot be used for multipath routing. This is because the cloned route is always used instead of alternating between the two multipath network routes. For this reason, you can see the discovered MTU value using the **netstat -rn** command.

In AIX 5L Version 5.3, there are some changes in the procedure for path MTU discovery. Here the ICMP echo reply and request packets are not used anymore. AIX 5L Version 5.3 uses TCP packets and UDP datagrams rather than ICMP echo reply and request packets. In addition, the discovered MTU will not be stored in the routing table. Therefore, it is possible to enable multipath routing to work with path MTU discovery.

When one system tries to optimize its transmissions by discovering the path MTU, a pmtu entry is created in a Path MTU (PMTU) table. You can display this table using the **pmtu display** command, as shown in Example 3-3. To avoid the accumulation of pmtu entries, unused pmtu entries will expire and be deleted when the pmtu_expire time is exceeded.

Example 3-3 Path MTU display

```
# pmtu display

   dst           gw            If    pmtu     refcnt   redisc_t    exp

 --------------------------------------------------------------------------

9.3.5.120     127.0.0.1       lo0   16896        6         24        0

9.3.4.155     9.3.5.41        en0    1500        1          0        0

9.3.5.195     9.3.5.120       en0    1500        0         20        4

127.0.0.1     127.0.0.1       lo0   16896        2          1        0
```

Path MTU table entry expiration is controlled by the pmtu_expire option of the **no** command. pmtu_expire is set to 10 minutes by default.

IPv6 never sends ICMPv6 packets to detect the PMTU. The first packet of a connection always starts the process. In addition, IPv6 routers must never fragment packets and must always return an ICMPv6 Packet too big message if they are unable to forward a packet because of a smaller outgoing MTU. Therefore, for IPv6, there are no changes necessary to make PMTU discovery work with multipath routing.

3.11.3 Using jumbo frames

Jumbo frame support on physical Ethernet adapters under the AIX 5L operating system has a simple design. It is controlled with an attribute on the physical adapter. Virtual Ethernet adapters support all possible MTU sizes automatically.

You can bridge jumbo packets on a virtual Ethernet connection.

There is no attribute for jumbo frames on a virtual Ethernet adapter. If an interface is configured on top of a virtual Ethernet adapter, there is an MTU value on the virtual Ethernet interface. Sending jumbo frames from the Shared Ethernet Adapter (SEA) interface is not available on the Virtual I/O Server Version 1.3, but bridging jumbo packets is. At the time of writing, packets to and from the SEA interface itself use an MTU of 1500.

However, the primary purpose of SEA is to bridge network communication between the virtual I/O clients and the external network. If the virtual adapter in the virtual I/O clients and the physical Ethernet adapter in the Virtual I/O Server associated with the SEA are all configured to MTU 9000 or have jumbo frames enabled, respectively, the traffic from the virtual I/O clients to the external network can have an MTU of 9000. Although the SEA cannot initiate network traffic using jumbo frames, it is able to bridge this traffic.

To configure jumbo frame communications between virtual I/O clients and an external network, use the following steps:

1. The virtual Ethernet adapter does not have either of the attributes jumbo frame or large send as seen on physical Ethernet adapters, as shown here:

```
# lsattr -El ent0
alt_addr       0x000000000000 Alternate Ethernet Address        True
chksum_offload yes            Checksum Offload Enable           True
copy_buffs     32             Transmit Copy Buffers             True
copy_bytes     65536          Transmit Copy Buffer Size         True
max_buf_huge   64             Maximum Huge Buffers              True
max_buf_large  64             Maximum Large Buffers             True
max_buf_medium 256            Maximum Medium Buffers            True
max_buf_small  2048           Maximum Small Buffers             True
max_buf_tiny   2048           Maximum Tiny Buffers              True
min_buf_huge   24             Minimum Huge Buffers              True
min_buf_large  24             Minimum Large Buffers             True
min_buf_medium 128            Minimum Medium Buffers            True
min_buf_small  512            Minimum Small Buffers             True
min_buf_tiny   512            Minimum Tiny Buffers              True
trace_debug    no             Trace Debug Enable                True
use_alt_addr   no             Enable Alternate Ethernet Address True
```

2. Virtual I/O Server (VIOS) configuration.

   You need to change an attribute on the VIOS. Change the MTU value of the physical adapter. In the case of a physical interface, you can configure the MTU 9000 by setting the jumbo frame setting with the following command:

   ```
   $ chdev -dev ent0 -attr jumbo_frames=yes
   ent0 changed
   ```

   If this physical adapter is in use by the SEA, remove the SEA and then re-create it. You can use the **`lsdev -dev ent0 -attr`** command to check the jumbo frame attribute of the physical adapter.

3. Virtual I/O client.

   On virtual devices, the following procedure can be applied to the virtual Ethernet adapter for changing the MTU value:

   ```
   lpar10:/]chdev -l en0 -a mtu=9000
   en0 changed
   lpar10:/]lsattr -El en0
   alias4                      IPv4 Alias including Subnet Mask           True
   alias6                      IPv6 Alias including Prefix Length         True
   arp           on            Address Resolution Protocol (ARP)          True
   authority                   Authorized Users                           True
   broadcast                   Broadcast Address                          True
   mtu           9000          Maximum IP Packet Size for This Device     True
   netaddr       9.3.5.120     Internet Address                           True
   netaddr6                    IPv6 Internet Address                      True
   netmask       255.255.255.0 Subnet Mask                                True
   prefixlen                   Prefix Length for IPv6 Internet Address    True
   remmtu        576           Maximum IP Packet Size for REMOTE Networks True
   rfc1323                     Enable/Disable TCP RFC 1323 Window Scaling True
   security      none          Security Level                             True
   state         up            Current Interface Status                   True
   tcp_mssdflt                 Set TCP Maximum Segment Size               True
   tcp_nodelay                 Enable/Disable TCP_NODELAY Option          True
   tcp_recvspace               Set Socket Buffer Space for Receiving      True
   tcp_sendspace               Set Socket Buffer Space for Sending        True
   ```

> **Tip:** Check the port on the switch that is connected to the real adapter associated with the SEA. This must have jumbo frames enabled.

3.11.4 Virtual Ethernet tuning recommendation with path MTU discovery

The purpose of this section is to test what throughput might be expected in a VLAN as the Interface Specific Network Options (ISNO) are changed. Each virtual I/O client is assigned one virtual processor and 0.5 shared processor in capped mode with simultaneous multithreading enabled.

Figure 3-13 on page 84 shows the connections between virtual I/O clients, passing through the POWER Hypervisor using virtual Ethernet adapters. The TCP_STREAM benchmark program was running in simplex and duplex mode at different tcp_sendspace and tcp_recvspace on both partitions.

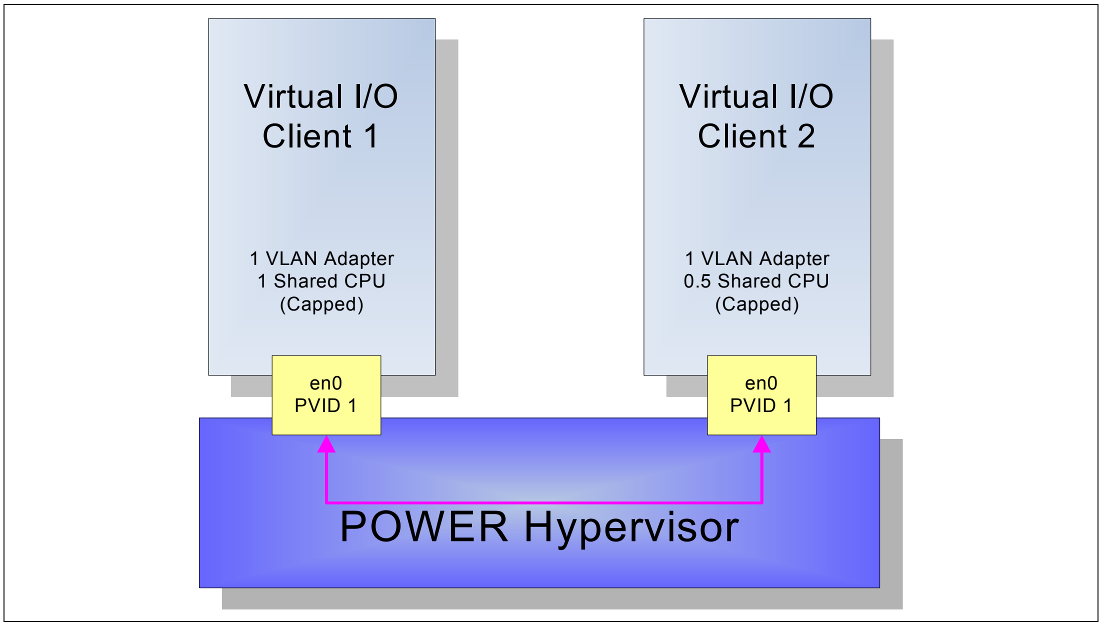


Figure 3-13 VLAN performance configuration

VLAN performance

Figure 3-14, Figure 3-15 on page 85, and Figure 3-16 on page 85 show how throughput can vary using different tcp_sendspace and tcp_recvspace in MTU 1500, 9000, and 65394 sizes.

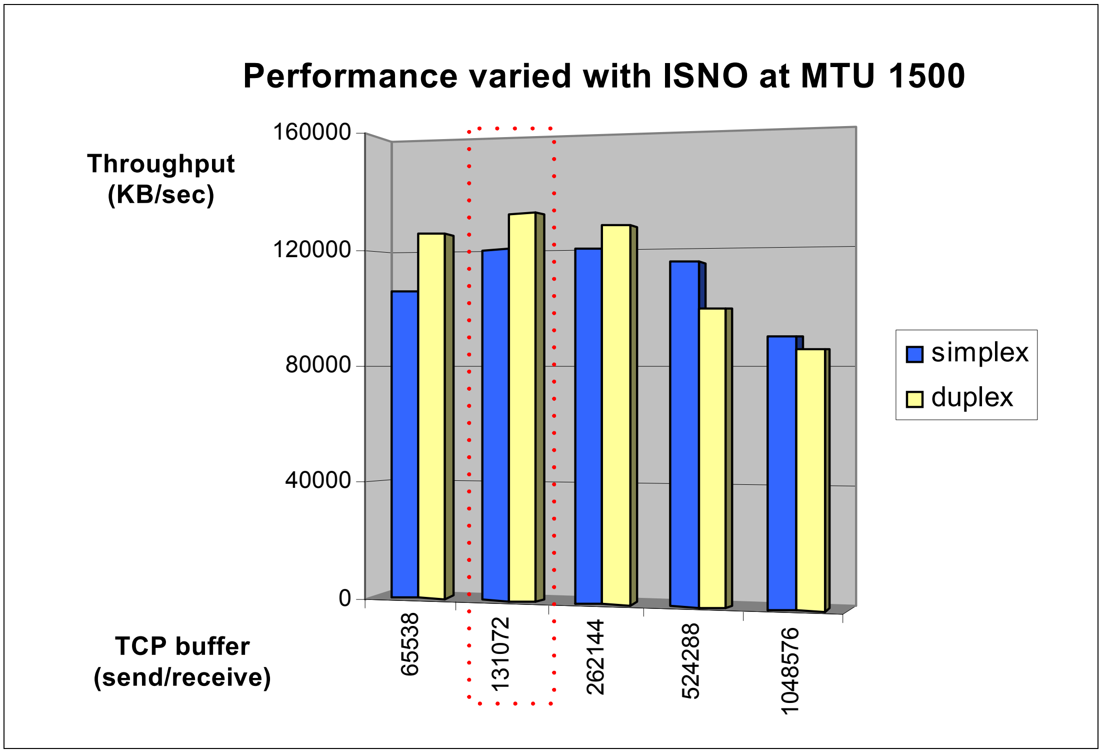


Figure 3-14 Performance varied with ISNO at MTU 1500

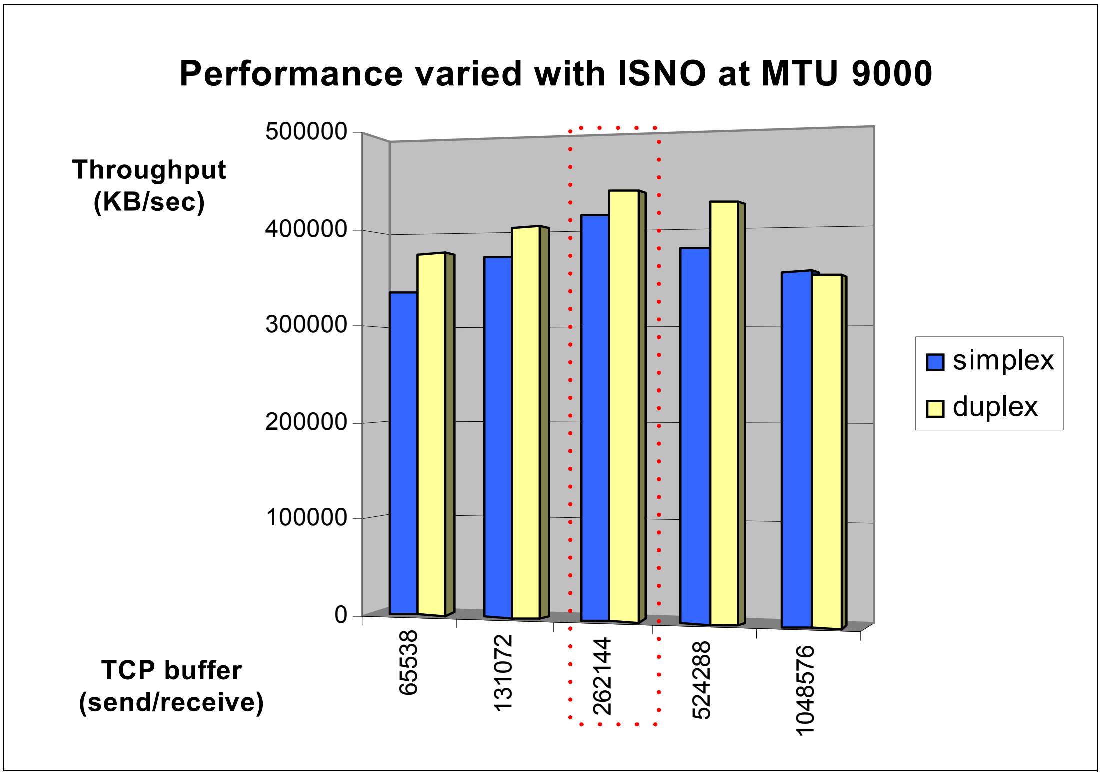


Figure 3-15 Performance varied with ISNO at MTU 9000

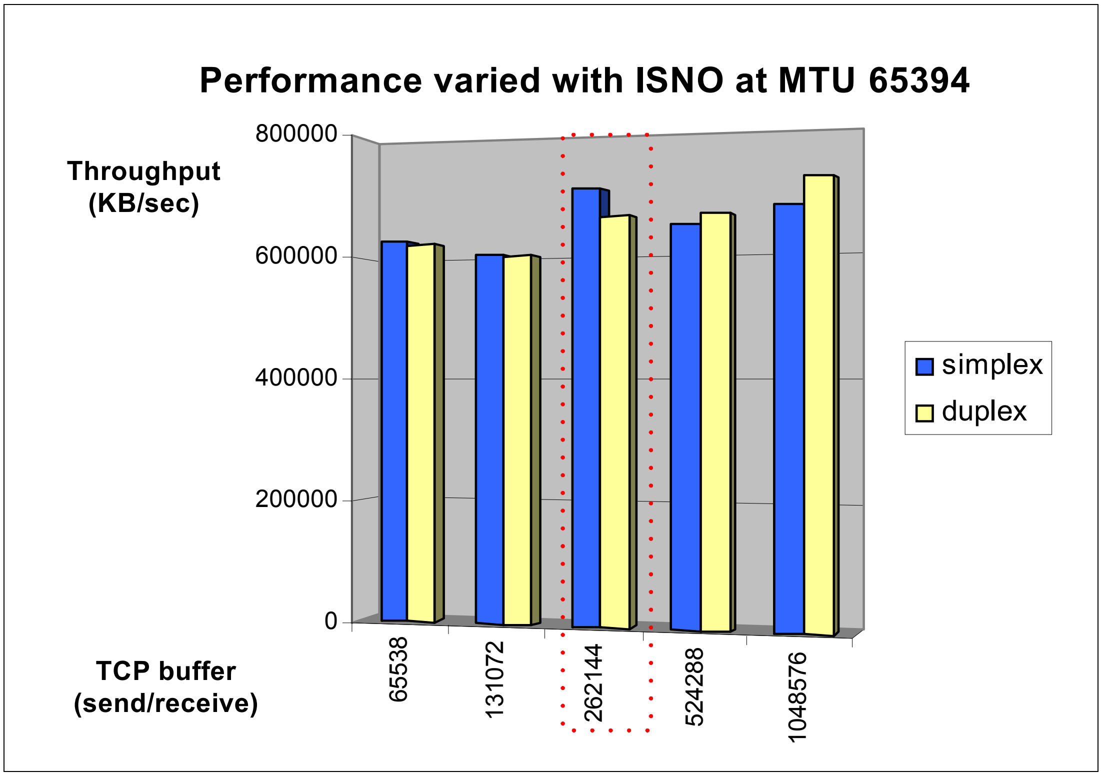


Figure 3-16 Performance varied with ISNO at MTU 65394

The MTU and ISNO settings, as shown, can be tuned to provide better network performance. Note the following considerations to get the best performance in a virtual network:

- Virtual Ethernet performance is based on CPU capacity entitlement and TCP/IP parameters such as MTU size, buffer size, and rfc1323 settings.
- If you have large data packets, selecting a high MTU size improves performance because more data per packet can be sent, and therefore, the data is sent using fewer packets.
- Keep the attributes tcp_pmtu_discover and chksum_offload set to their default values.
- Do not turn off simultaneous multithreading unless your applications require it.

3.11.5 TCP checksum offload

The TCP checksum offload option enables the network adapter to verify the TCP checksum on transmit and receive, which saves the host CPU from having to compute the checksum. This feature is used to detect a corruption of data in the packet on physical adapters, so virtual Ethernet adapters do not need the checksum process. The checksum offload option in virtual Ethernet adapters is enabled by default, as in physical Ethernet adapters. If you want the best performance between virtual Ethernet adapters, enable the checksum offload option on the source and destination system. If it is disabled in the source or destination system, the hypervisor detects the state and validates the checksum when it is needed.

3.11.6 Largesend option

IBM System p Gigabit or higher Ethernet adapters support TCP segmentation offload (also called largesend). This feature extends the TCP largesend feature to virtual Ethernet adapters and Shared Ethernet Adapters (SEA). In largesend environments, the TCP will send a big *chunk* of data to the adapter when TCP knows that adapter supports largesend. The adapter will break this big TCP packet into multiple smaller TCP packets that will fit the outgoing MTU of the adapter, saving system CPU load and increasing network throughput.

The TCP largesend feature is extended from LPAR all the way up to the real adapter of VIOS. The TCP stack on the LPAR will determine if the VIOS supports largesend. If VIOS supports TCP largesend, the LPAR sends a big TCP packet directly to VIOS.

If virtual Ethernet adapters are used in a LPAR-LPAR environment, however, the large TCP packet does not need to be broken into multiple smaller packets. This is because the underlying hypervisor will take care of sending the big chunk of data from one LPAR to another LPAR.

This feature allows the use of a large MTU for LPAR-LPAR communication, resulting in very significant CPU savings and increasing network throughput.

The largesend option using Virtual I/O Server Version 1.3 is available on the SEA when it is used as a bridge device. The largesend option for packets originating from the SEA interface is not available using Virtual I/O Server Version 1.3 (packets coming from the Virtual I/O Server itself).

To use the SEA largesend feature, ensure that the largesend attribute has a value of 1 for all the virtual adapters of the LPARs.

You can check the largesend option on the SEA using Virtual I/O Server Version 1.3, as shown in Example 3-4. In this example, it is set to off.

Example 3-4   Largesend option for SEA

```
$ lsdev -dev ent6 -attr
attribute    value    description
user_settable

ctl_chan             Control Channel adapter for SEA failover
True
ha_mode      disabled High Availability Mode
True
largesend    0       Enable Hardware Transmit TCP Resegmentation
True
netaddr      0        Address to ping
True
pvid         1       PVID to use for the SEA device
True
pvid_adapter  ent5   Default virtual adapter to use for non-VLAN-tagged packets
True
real_adapter  ent0    Physical adapter associated with the SEA
True
thread       1       Thread mode enabled (1) or disabled (0)
True
virt_adapters ent5     List of virtual adapters associated with the SEA (comma
separated) True
```

For more information about tuning the performance of your network, see the IBM eServer pSeries and AIX Information Center:

`http://publib16.boulder.ibm.com/pseries/index.htm`

Alternatively, for additional performance test result and further recommendations, see *Advanced POWER Virtualization on IBM @server p5 Servers: Architecture and Performance Considerations*, SG24-5768.

4

Storage

The Virtual I/O Server maps physical storage to virtual I/O clients. This chapter outlines the best practices for managing storage in the virtual environment, keeping track of physical storage, and allocating it to virtual I/O clients.

We address maintenance scenarios such as replacing disks and expanding the size of an exported logical volume.

This chapter also includes migration scenarios, including moving a partition with virtual storage from one server to another, and moving existing storage (for example, physical or dedicated) into the virtual environment where possible.

4.1 Managing and exporting physical storage on the VIOS

Use the SAN to manage storage requirements.

The Virtual I/O Server (VIOS) presents disk storage to virtual I/O clients (VIOC) as virtual SCSI disks. These virtual disks must be mapped to physical storage by the VIOS. There are three different ways to perform this mapping, each with its own advantages:

- Physical volumes
- Logical volumes
- Storage pools

The general rule for selecting between these options is that disk devices being accessed through a SAN should be exported as physical volumes, with storage allocation managed in the SAN. Internal and SCSI attached disk devices should be exported with either logical volumes or storage pools so that storage can be allocated within the server. The following sections describe each option in more detail.

4.1.1 Physical volumes

The Virtual I/O Server can export physical volumes intact to virtual I/O clients. This method of exporting storage has several advantages over logical volumes:

- Physical disk devices can be exported from two or more Virtual I/O Servers concurrently for multipath redundancy.
- The code path for exporting physical volumes is shorter, which might lead to better performance.
- Physical disk devices can be moved from one Virtual I/O Server to another with relative ease.
- In some cases, existing LUNs from physical servers can be migrated into the virtual environment with the data intact.
- One consideration for exporting physical volumes is that the size of the device is not managed by the Virtual I/O Servers, and the Virtual I/O Server does not allow partitioning of a single device among multiple clients. This is generally only a concern for internal and SCSI attached disks.

There is not generally a requirement to subdivide SAN attached disks, because storage allocation can be managed at the storage server. In the SAN environment, provision and allocate LUNs for each LPAR on the storage servers and export them from the Virtual I/O Server as physical volumes.

When SAN disk is available, all storage associated with a virtual I/O client should be stored in the SAN, including rootvg and paging space. This makes management simpler because partitions will not be dependent on both internal logical volumes and external LUNs. It also makes it easier to move LPARs from one Virtual I/O Server to another. For more information, see 4.6, “Moving an AIX 5L LPAR from one server to another” on page 107.

4.1.2 Logical volumes

The Virtual I/O Server can export logical volumes to virtual I/O clients. This method does have some advantages over physical volumes:

- Logical volumes can subdivide physical disk devices between different clients.
- The logical volume interface is familiar to those with AIX 5L experience.

Note: Using the rootvg on the Virtual I/O Server to host exported logical volumes is currently not recommended. Certain types of software upgrades and system restores might alter the logical volume to target device mapping for logical volumes within rootvg, requiring manual intervention.

We recommend that the virtual I/O client use LVM mirroring if redundancy is required.

When an internal or SCSI attached disk is used, the logical volume manager enables disk devices to be subdivided between different virtual I/O clients. For small servers, this enables several LPARs to share internal disks or RAID arrays.

Logical volumes cannot be accessed by multiple Virtual I/O Servers concurrently, so they cannot be used with multipath I/O (MPIO) on the virtual I/O client.

4.1.3 Storage pools

When managed by the Integrated Virtualization Manager (IVM), the Virtual I/O Server can export storage pool backing devices to virtual I/O clients. This method is similar to logical volumes, and it does have some advantages over physical volumes:

- Storage pool backing devices can subdivide physical disk devices between different clients.
- The storage pool interface is easy to use through IVM.

Important: The default storage pool within IVM is the root volume group of the Virtual I/O Server. Be careful not to allocate backing devices within the root volume group because certain types of software upgrades and system restores might alter the logical volume to target device mapping for logical volumes within rootvg, requiring manual intervention.

Systems in a single server environment under the management of IVM are often not attached to a SAN, and these systems typically use internal and SCSI attached disk storage. The IVM interface allows storage pools to be created on physical storage devices so that a single physical disk device can be divided among several virtual I/O clients.

As with logical volumes, storage pool backing devices cannot be accessed by multiple Virtual I/O Servers concurrently, so they cannot be used with MPIO on the virtual I/O client.

We recommend that the virtual I/O client use LVM mirroring if redundancy is required.

4.1.4 Best practices for exporting logical volumes

The IVM and HMC managed environments present similar interfaces for storage management under different names. The storage pool interface under IVM is essentially the same as the logical volume manager interface under the HMC, and in some cases, the documentation will use the terms interchangeably. The remainder of this section uses the term volume group to refer to both volume groups and storage pools, and the term logical volume to refer to both logical volumes and storage pool backing devices.

Logical volumes enable the Virtual I/O Server to subdivide a physical volume between multiple virtual I/O clients. In many cases, the physical volumes used will be internal disks, or RAID arrays built of internal disks.

A single volume group should not contain logical volumes used by virtual I/O clients and logical volumes used by the Virtual I/O Server operating system. Keep Virtual I/O Server file systems within the rootvg, and use other volume groups to host logical volumes for virtual I/O clients.

A single volume group or logical volume cannot be accessed by two Virtual I/O Servers concurrently. Do not attempt to configure MPIO on virtual I/O clients for VSCSI devices that reside on logical volumes. If redundancy is required in logical volume configurations, use LVM mirroring on the virtual I/O client to mirror across different logical volumes on different Virtual I/O Servers.

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.

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

When exporting logical volumes to clients, the mapping of individual logical volumes to virtual I/O clients is maintained on the VIOS. The additional level of abstraction provided by the logical volume manager makes it important to track the relationship between physical disk devices and virtual I/O clients. For more information, see 4.3, “Managing the mapping of LUNs to VSCSI to hdisks” on page 94.

4.2 Expanding the size of virtual storage devices

An exported storage device on the Virtual I/O Server can be either a logical volume or physical volume. The size of an exported logical volume can be expanded on the Virtual I/O Server and many types of SAN-based physical volumes can be expanded as well. Consult your storage vendor for information about whether your storage subsystem supports expanding the size of an existing LUN.

The following steps outline the procedure to extend an exported logical volume and make the virtual I/O client recognize the change. Although there are some extra commands on the Virtual I/O Server to modify the size of a logical volume, the same commands are used on the virtual I/O client for both internal and external device changes.

> **Note:** You cannot re-size physical volumes in the rootvg on AIX 5L with the volume group activated in classic or enhanced concurrent mode. The rootvg can be extended by adding additional physical volumes.

After a volume has been extended either in the SAN, or using the logical volume manager on the Virtual I/O Server, the administrator needs to execute the **chvg -g** command on the virtual I/O client (if it is AIX 5L). The **chvg** command examines all the disks in the volume group to see if they have grown in size. You might be required to execute **varyoffvg** and **varyonvg** commands on the volume group for LVM to see the size change on the disks.

To extend a logical volume on the Virtual I/O Server and recognize the change on the virtual I/O client, perform the following steps:

1. Example 4-1 on page 93 shows the status of a logical volume and extending the size of the logical volume on the Virtual I/O Server. To change the logical volume (LV) size, use the **extendlv** command as follows:

   ```
   extendlv  LogicalVolume Size [PhysicalVolume ...]
   ```

Example 4-1 Check the status LV and extending LV size at VIOS

```
$ lslv db_lv
LOGICAL VOLUME:     db_lv                  VOLUME GROUP:   db_sp
LV IDENTIFIER:      00cddeec00004c000000000c4a8b3d81.1 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            32512                  PP SIZE:        32 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                320                    PPs:            320
STALE PPs:          0                      BB POLICY:      non-relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    1024
MOUNT POINT:        N/A                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
DEVICESUBTYPE : DS_LVZ

$ extendlv db_lv 5G

$ lslv db_lv
LOGICAL VOLUME:     db_lv                  VOLUME GROUP:   db_sp
LV IDENTIFIER:      00cddeec00004c000000000c4a8b3d81.1 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            32512                  PP SIZE:        32 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                480                    PPs:            480
STALE PPs:          0                      BB POLICY:      non-relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    1024
MOUNT POINT:        N/A                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
DEVICESUBTYPE :
DS_LVZ
```

2. Example 4-2 shows how to recognize the change on the virtual I/O client using the **chvg** command as follows:

   ```
   chvg -g Volumegroup_name
   ```

Example 4-2 The recognition of LV change at VIOC

```
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                   VOLUME GROUP:     oravg
PV IDENTIFIER:      00cddeec6828645d VG IDENTIFIER
00cddeec00004c000000000c4a8d4346
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)           LOGICAL VOLUMES:  0
TOTAL PPs:          639 (10224 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           639 (10224 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  128..128..127..128..128
```

```
USED DISTRIBUTION:  00..00..00..00..00

# chvg -g oravg

# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                   VOLUME GROUP:     oravg
PV IDENTIFIER:      00cddeec6828645d VG IDENTIFIER
00cddeec00004c000000000c4a8d4346
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)           LOGICAL VOLUMES:  0
TOTAL PPs:          959 (15344 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           959 (15344 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  192..192..191..192..192
USED DISTRIBUTION:
00..00..00..00..00
```

4.3 Managing the mapping of LUNs to VSCSI to hdisks

One of the keys to managing a virtual environment is keeping track of what virtual objects correspond to what physical objects. This is particularly challenging in the storage arena where individual LPARs can have hundreds of virtual disks. This mapping is critical to manage performance and to understand what systems will be affected by hardware maintenance.

Virtual disks can be mapped to physical disks in one of two different ways (Figure 4-1 on page 95):

- Physical volumes
- Logical volumes

Logical volumes can be mapped from volume groups or storage pools. For more information about which method is right for your environment, see 4.1, “Managing and exporting physical storage on the VIOS” on page 90.

For information about mapping network devices and virtual LANs, see 3.3, “Managing the mapping of network devices” on page 58.

The `lshwres` HMC command can help clients map the virtual I/O client, vhost, and other information if they do not have it mapped correctly.

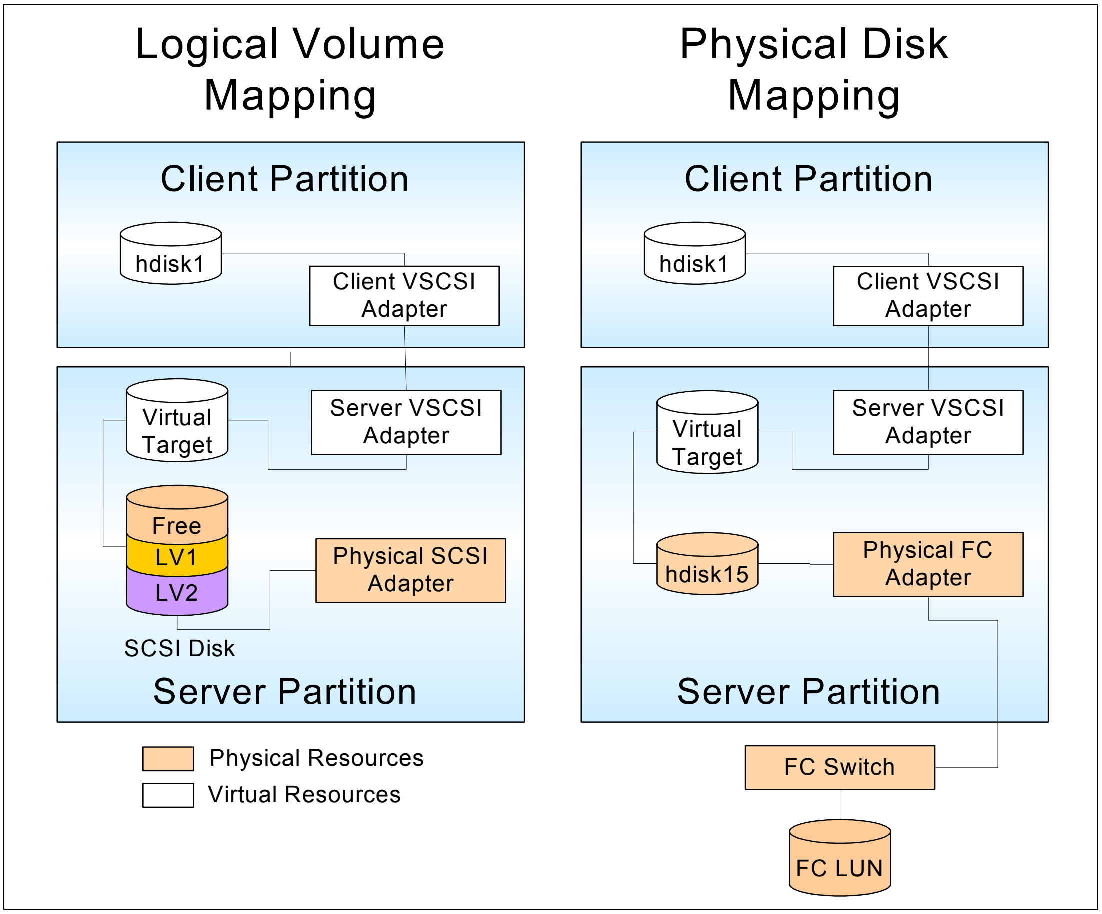


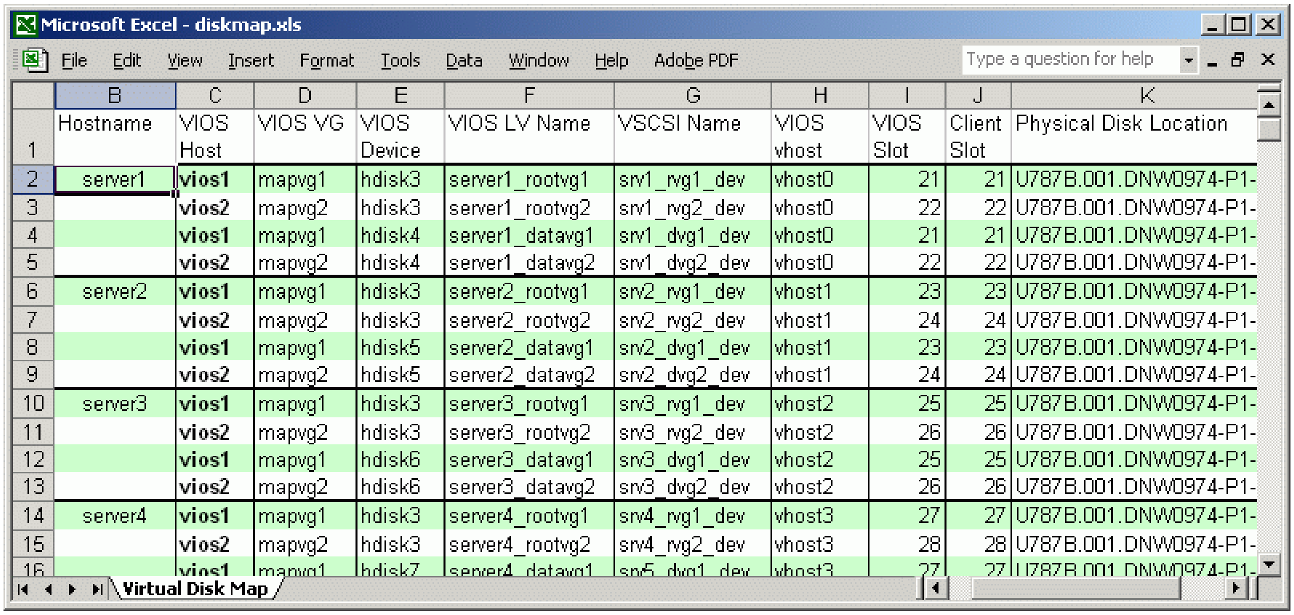
Figure 4-1 Logical versus physical drive mapping

Depending on which method you choose, you might need to track the following information:

- Virtual I/O Server
  - Server host name
  - Physical disk location
  - Physical adapter device name
  - Physical hdisk device name
  - Volume group or storage pool name[1]
  - Logical volume or storage pool backing device name[1]
  - VSCSI adapter slot
  - VSCSI adapter device name
  - Virtual target device
- Virtual I/O client
  - Client host name
  - VSCSI adapter slot
  - VSCSI adapter device name
  - Virtual hdisk device name

Because of the number of fields to be tracked, we recommend using a spreadsheet or database program, such as shown Figure 4-2 on page 96, to track this information. Record the data when the system is installed, and track it over time as the configuration changes.

[1] For logical volume or storage pool export only



	B	C	D	E	F	G	H	I	J	K
1	Hostname	VIOS Host	VIOS VG	VIOS Device	VIOS LV Name	VSCSI Name	VIOS vhost	VIOS Slot	Client Slot	Physical Disk Location
2	server1	vios1	mapvg1	hdisk3	server1_rootvg1	srv1_rvg1_dev	vhost0	21	21	U787B.001.DNW0974-P1-
3		vios2	mapvg2	hdisk3	server1_rootvg2	srv1_rvg2_dev	vhost0	22	22	U787B.001.DNW0974-P1-
4		vios1	mapvg1	hdisk4	server1_datavg1	srv1_dvg1_dev	vhost0	21	21	U787B.001.DNW0974-P1-
5		vios2	mapvg2	hdisk4	server1_datavg2	srv1_dvg2_dev	vhost0	22	22	U787B.001.DNW0974-P1-
6	server2	vios1	mapvg1	hdisk3	server2_rootvg1	srv2_rvg1_dev	vhost1	23	23	U787B.001.DNW0974-P1-
7		vios2	mapvg2	hdisk3	server2_rootvg2	srv2_rvg2_dev	vhost1	24	24	U787B.001.DNW0974-P1-
8		vios1	mapvg1	hdisk5	server2_datavg1	srv2_dvg1_dev	vhost1	23	23	U787B.001.DNW0974-P1-
9		vios2	mapvg2	hdisk5	server2_datavg2	srv2_dvg2_dev	vhost1	24	24	U787B.001.DNW0974-P1-
10	server3	vios1	mapvg1	hdisk3	server3_rootvg1	srv3_rvg1_dev	vhost2	25	25	U787B.001.DNW0974-P1-
11		vios2	mapvg2	hdisk3	server3_rootvg2	srv3_rvg2_dev	vhost2	26	26	U787B.001.DNW0974-P1-
12		vios1	mapvg1	hdisk6	server3_datavg1	srv3_dvg1_dev	vhost2	25	25	U787B.001.DNW0974-P1-
13		vios2	mapvg2	hdisk6	server3_datavg2	srv3_dvg2_dev	vhost2	26	26	U787B.001.DNW0974-P1-
14	server4	vios1	mapvg1	hdisk3	server4_rootvg1	srv4_rvg1_dev	vhost3	27	27	U787B.001.DNW0974-P1-
15		vios2	mapvg2	hdisk3	server4_rootvg2	srv4_rvg2_dev	vhost3	28	28	U787B.001.DNW0974-P1-
16		vios1	mapvg1	hdisk7	server4_datavg1	srv5_dvg1_dev	vhost3	27	27	U787B.001.DNW0974-P1-

Figure 4-2 Disk tracking spreadsheet

4.3.1 Naming conventions

Develop naming conventions for your enterprise.

In addition to a tracking tool such as a spreadsheet, a good naming convention is key to managing this information. One strategy for reducing the amount of data that must be tracked is to make device names and slots match on the virtual I/O client and server wherever possible.

This can include corresponding volume group, logical volume, and virtual target device names. Integrating the virtual I/O client host name into the virtual target device name can simplify tracking on the server.

When using fibre channel disks on a storage server that supports LUN naming, this feature can be used to make it easier to identify LUNs. Commands such as **lssdd** for the IBM System Storage™ DS8000™ and DS6000™ series storage servers, and the **fget_config** command for the DS4000™ series can be used to match hdisk devices with LUN names as in the following example. Because the **fget_config** command is part of a storage device driver, you must use the **oem_setup_env** command to access it on the Virtual I/O Server.

```
$ oem_setup_env
# fget_config -Av

---dar0---

User array name = 'FAST200'
dac0 ACTIVE dac1 ACTIVE

Disk     DAC   LUN Logical Drive
utm             31
hdisk4   dac1    0 Server1_LUN1
hdisk5   dac1    1 Server1_LUN2
hdisk6   dac1    2 Server-520-2-LUN1
hdisk7   dac1    3 Server-520-2-LUN2
```

In many cases, using LUN names can be simpler than tracing devices using Fibre Channel world wide port names and numeric LUN identifiers.

4.3.2 Virtual device slot numbers

Develop a slot numbering policy for your enterprise.

After establishing the naming conventions, also establish slot numbering conventions for the virtual I/O adapters.

Slot numbers are shared between virtual storage and virtual network devices. In complex systems, there will tend to be far more storage devices than network devices because each virtual SCSI device can only communicate with one server or client. We recommend reserving the slot numbers through 20 for network devices on all LPARs in order to keep the network and storage devices grouped together.

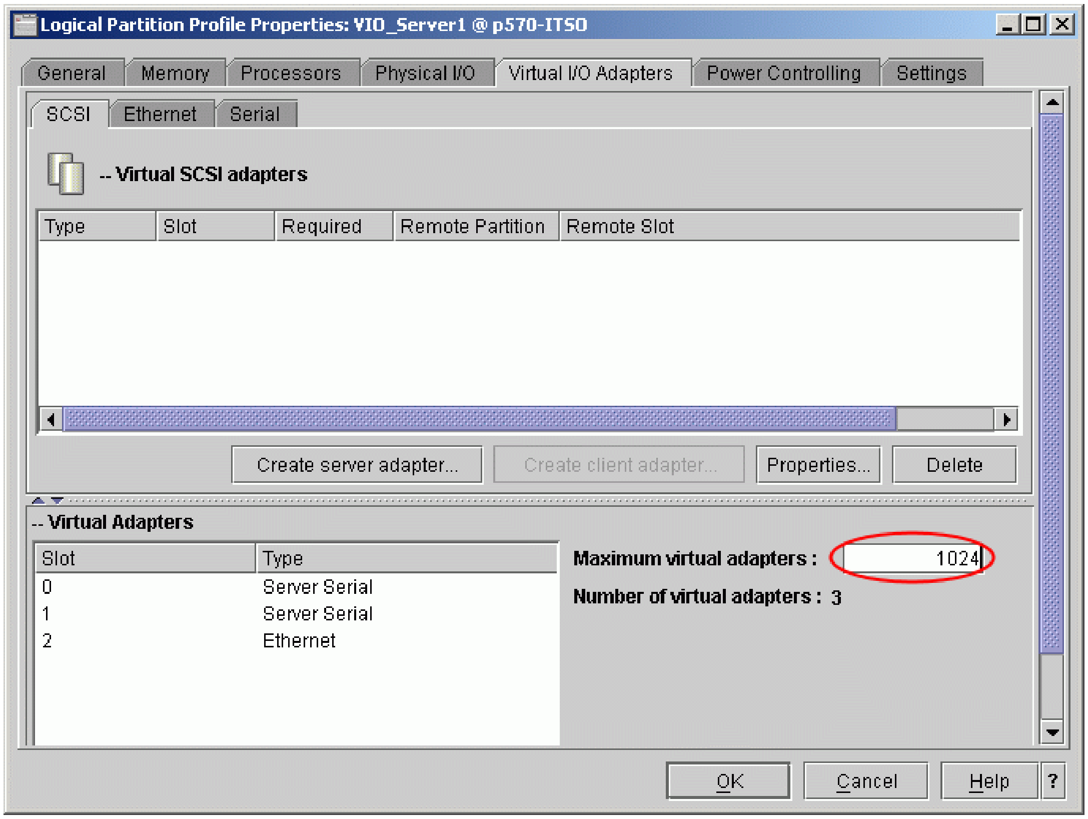
Management can be simplified by keeping slot numbers consistent between the virtual I/O client and server. However, when partitions are moved from one server to another, this might not be possible. For more information, see 4.6.3, “Storage planning” on page 108.

In environments with only one virtual I/O server, add storage adapters incrementally starting with slot 21 and higher. When clients are attached to two Virtual I/O Servers, the adapter slot numbers should be alternated from one VIOS to the other. The first VIOS should use odd numbered slots starting at 21, and the second should use even numbered slots starting at 22. In a two server scenario, allocate slots in pairs, with each client using two adjacent slots such as 21 and 22, or 33 and 34.

Increase the maximum number of virtual adapter slots (Figure 4-3 on page 98) per LPAR above the default value of 10 when you create an LPAR. The appropriate number for your environment depends on the number of LPARs and adapters expected on each system. Each unused virtual adapter slot consumes a small amount of memory, so the allocation should be balanced. Use the System Planning Tool available from the following URL to plan memory requirements for your system configuration:

`http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html`

> **Important:** When planning for the number of virtual I/O slots on your LPAR, the maximum number of virtual adapter slots available on a partition is set by the partition’s profile. To change this profile, shut down the LPAR. We recommend leaving plenty of room for expansion when setting the maximum number of slots so that new virtual I/O clients can be added without shutting down the LPAR or Virtual I/O Server partition.



-- Virtual SCSI adapters

Type	Slot	Required	Remote Partition	Remote Slot

-- Virtual Adapters

Slot	Type
0	Server Serial
1	Server Serial
2	Ethernet

Maximum virtual adapters : 1024

Number of virtual adapters : 3

Figure 4-3 Maximum virtual adapters

Because VSCSI connections operate at memory speed, there is generally no performance gain from adding multiple adapters between a Virtual I/O Server and client. Each adapter pair can handle up to 85 virtual devices with the default queue depth of three. In situations where virtual devices per partition are expected to exceed that number, or where the queue depth on some devices might be increased above the default, reserve an additional adapter slot per server, one additional slot per VIOS with which the client will communicate.

4.3.3 Tracing a configuration

Despite the best intentions in record keeping, it sometimes becomes necessary to manually trace a client virtual disk back to the physical hardware. The IBM Systems Hardware Information Center contains a guide to tracing virtual disks, available at:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1_vios_managing_mapping.htm

4.4 Replacing a disk on the Virtual I/O Server

If it becomes necessary to replace a disk on the Virtual I/O Server, you must first identify the virtual I/O clients affected and the target disk drive.

> **Tip:** Before replacing a disk device using this procedure, check that the disk can be hot swapped.

Hot swapping goes beyond the hardware. It requires coordination with active systems.

If you run in a single Virtual I/O Server environment without disk mirroring on the virtual I/O clients, the disk replacement requires data to be restored. This also applies if you have the same disk exported through two Virtual I/O Servers using MPIO. MPIO by itself does not protect against outages due to disk replacement. You should protect the data on a disk or LUN using either mirroring or RAID technology, regardless of whether single or dual Virtual I/O Servers or MPIO are used.

There are three ways to export disk storage to virtual I/O clients. These are described in 4.1, “Managing and exporting physical storage on the VIOS” on page 90. This section outlines the procedure for replacing a disk device in the logical volume and storage pool environments.

In order to check the state of a physical volume, use the **lsvg -pv** command for both the logical volume and storage pool environments, as shown in the following example:

```
#lsvg -pv vioc_rootvg
vioc_rootvg:
PV_NAME           PV STATE          TOTAL PPs    FREE PPs     FREE DISTRIBUTION
hdisk3            active            1092         580          219..00..00..142..219
```

Note: Before replacing the disk, document the virtual I/O client, logical volume (LV), backing devices, vhost and vtscsi associated, and the size of the vtscsi device. See 4.3, “Managing the mapping of LUNs to VSCSI to hdisks” on page 94 for more information about managing this.

4.4.1 Replacing a disk in the LVM environment

In the LVM scenario, we want to replace hdisk2 in the volume group vioc_rootvg_1, which contains the LVs vioc_1_rootvg associated to vtscsi0. It has the following attributes:

- The size is 32 GB.
- The virtual disk is LVM mirrored on the virtual I/O client.
- The failing disk on the virtual I/O client is hdisk1.
- The virtual SCSI adapter on the virtual I/O client is vscsi1.
- The volume group on the virtual I/O client is rootvg.

Figure 4-4 shows the setup.

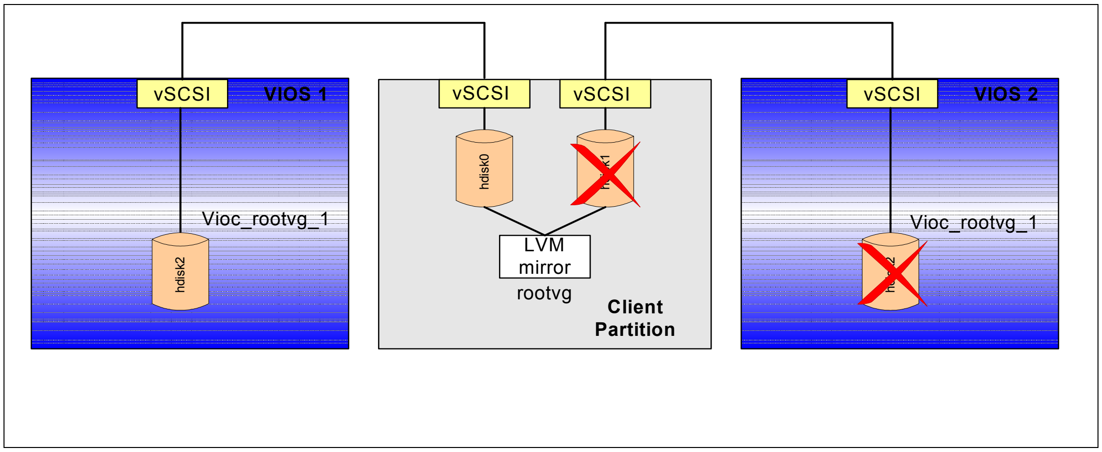
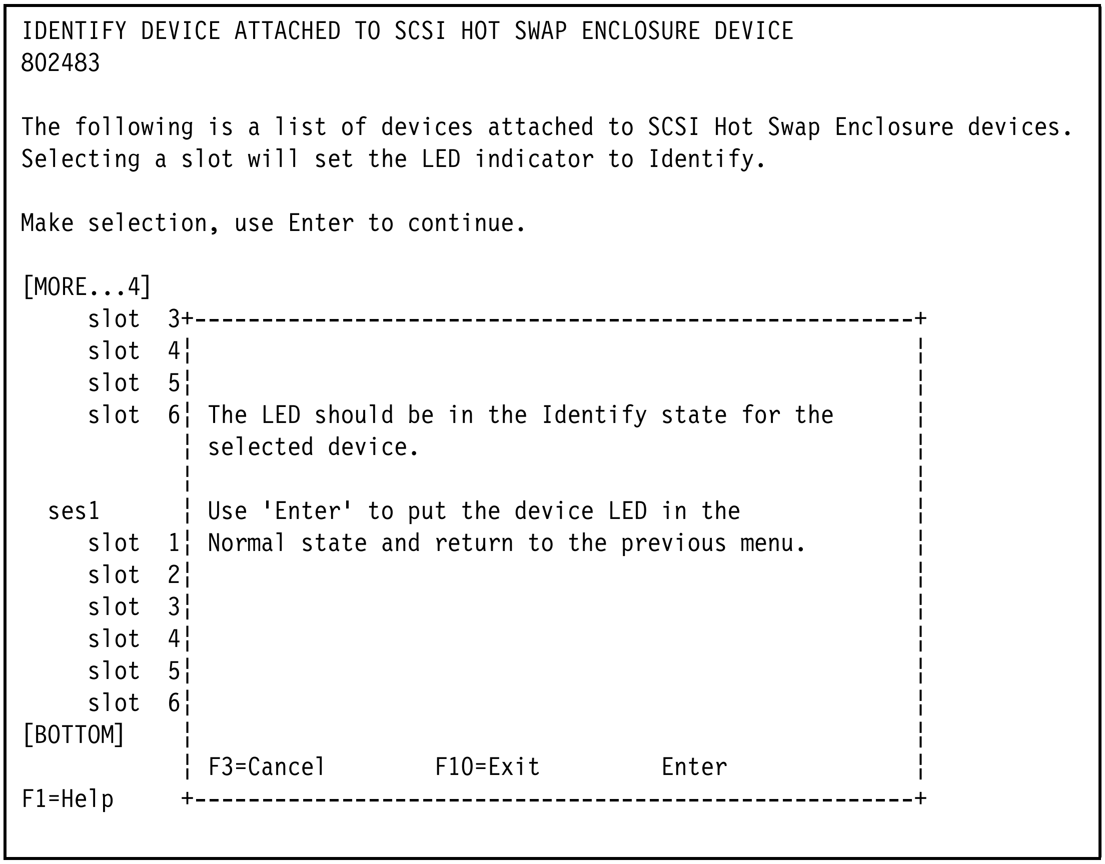


Figure 4-4 The LVM environment

To replace a disk:

1. Identify the physical disk drive with the **`diagmenu`** command.
2. Then, select **Task Selection (Diagnostics, Advanced Diagnostics, Service Aids, etc.)**. In the next list, select **Hot Plug Task**.
3. In this list, select **SCSI and SCSI RAID Hot Plug Manager** and select **Identify a Device Attached to a SCSI Hot Swap Enclosure Device**.
4. In the next list, find the hdisk and press Enter. A screen similar to the one in Figure 4-5 opens. Note that this is an example from an p5-570 with internal disk.



```
IDENTIFY DEVICE ATTACHED TO SCSI HOT SWAP ENCLOSURE DEVICE
802483

The following is a list of devices attached to SCSI Hot Swap Enclosure devices.
Selecting a slot will set the LED indicator to Identify.

Make selection, use Enter to continue.

[MORE...4]
     slot  3+-------------------------------------------------------+
     slot  4¦                                                       ¦
     slot  5¦                                                       ¦
     slot  6¦ The LED should be in the Identify state for the       ¦
            ¦ selected device.                                      ¦
            ¦                                                       ¦
  ses1      ¦ Use 'Enter' to put the device LED in the              ¦
     slot  1¦ Normal state and return to the previous menu.         ¦
     slot  2¦                                                       ¦
     slot  3¦                                                       ¦
     slot  4¦                                                       ¦
     slot  5¦                                                       ¦
     slot  6¦                                                       ¦
[BOTTOM]    ¦                                                       ¦
            ¦ F3=Cancel        F10=Exit         Enter               ¦
F1=Help     +-------------------------------------------------------+
```

Figure 4-5 Find the disk to remove

5. On the virtual I/O client, unmirror the rootvg, as follows:

```
# unmirrorvg -c 1 rootvg hdisk1
0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv -c
<diskname>'
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.

0301-108 mkboot: Unable to read file blocks. Return code: -1
0516-1132 unmirrorvg: Quorum requirement turned on, reboot system for this
        to take effect for rootvg.
0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
        bosboot of system to reinitialize boot records.  Then, user must modify
        bootlist to just include:  hdisk0.
```

6. On the virtual I/O client, reduce the rootvg:

```
# reducevg rootvg hdisk1
#
```

7. On the virtual I/O client , remove the hdisk1 device:

```
# rmdev -l hdisk1 -d
hdisk1 deleted
#
```

8. On the Virtual I/O Server, remove the vtscsi/vhost association:

```
$ rmdev -dev vtscsi0
vtscsi0 deleted
$
```

9. On the Virtual I/O Server, reduce the volume group. If you get an error, as in the following example, and you are sure that you have only one hdisk per volume group, you can use the **deactivatevg** and **exportvg** commands.

> **Note:** If you use the **exportvg** command, it will delete all the logical volumes inside the volume group, and if your volume group contains more than one hdisk, the logical volumes on this hdisk are also affected. Use the **lspv** command to check. In this case, it is safe to use the **exportvg vioc_rootvg_1** command:
>
> ```
> $ lspv
> NAME          PVID                                VG              STATUS
> hdisk0        00c478de00655246                    rootvg          active
> hdisk1        00c478de008a399b                    rootvg          active
> hdisk2        00c478de008a3ba1                    vioc_rootvg_1   active
> hdisk3        00c478deb4b0d4b0                    None
> $
> ```

```
$ reducevg -rmlv -f vioc_rootvg_1 hdisk2


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
0516-882 reducevg: Unable to reduce volume group.
$ deactivatevg vioc_rootvg_1


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
$ exportvg vioc_rootvg_1
$
```

10. On the Virtual I/O Server, remove the hdisk device:

```
$ rmdev -dev hdisk2
hdisk2 deleted
$
```

11. Replace the physical disk drive.
12. On the Virtual I/O Server, configure the new hdisk device with the **cfgdev** command and check that the new disk is configured:

```
$ cfgdev
$ lspv
NAME            PVID                                VG              STATUS
hdisk2          none                                None
hdisk0          00c478de00655246                    rootvg          active
hdisk1          00c478de008a399b                    rootvg          active
hdisk3          00c478deb4b0d4b0                    None
$
```

13. On the Virtual I/O Server, extend the volume group with the new hdisk using the **mkvg** command if you only have one disk per volume group. Use the **extendvg** command if you have more disks per volume group. In this case, we have only one volume group per disk, which is recommended. If the disk has a PVID, use the **-f** flag on the **mkvg** command.

```
$ mkvg -vg vioc_rootvg_1 hdisk2
vioc_rootvg_1
0516-1254 mkvg: Changing the PVID in the ODM.
$
```

14. On the Virtual I/O Server, make the logical volume for the vtscsi device:

```
$ mklv -lv vioc_1_rootvg vioc_rootvg_1 32G
vioc_1_rootvg
$
```

15. On the Virtual I/O Server, check that the LV does not span disks:

```
$ lslv -pv vioc_1_rootvg
vioc_1_rootvg:N/A
PV                COPIES        IN BAND       DISTRIBUTION
hdisk2            512:000:000   42%           000:218:218:076:000
$
```

16. On the Virtual I/O Server, make the virtual device:

```
$ mkvdev -vdev vioc_1_rootvg -vadapter vhost0
vtscsi0 Available
$
```

17. On the virtual I/O client, reconfigure the new hdisk1:

```
# cfgmgr -l vscsi1
#
Or if you do not know the parent device
# cfgmgr
#
```

18. On the virtual I/O client, extend the rootvg:

```
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
#
```

19. On the virtual I/O client, mirror the rootvg:

```
# mirrorvg -c 2 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:  hdisk0 hdisk1.
#
```

20. On the virtual I/O client, initialize boot records and set the bootlist:

```
# bosboot -a

bosboot: Boot image is 18036 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
#
```

4.4.2 Replacing a disk in the storage pool environment

In the storage pool scenario, we want to replace hdisk2 in the storage pool vioc_rootvg_1. It has the following attributes:

- It contains backing device vioc_1_rootvg associated to vhost0.
- The size is 32 GB.
- The virtual disk is mirrored on the virtual I/O client.
- The volume group on the virtual I/O client is rootvg.

To replace the disk, perform the following steps:

1. Identify the physical disk drive; see step 1 on page 100.
2. On the virtual I/O client, un-mirror the rootvg:

```
# unmirrorvg -c 1 rootvg hdisk1
0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv -c
<diskname>'
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.

0301-108 mkboot: Unable to read file blocks. Return code: -1
0516-1132 unmirrorvg: Quorum requirement turned on, reboot system for this
        to take effect for rootvg.
0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
        bosboot of system to reinitialize boot records.  Then, user must modify
        bootlist to just include:  hdisk0.
```

3. On the virtual I/O client, reduce the rootvg:

```
# reducevg rootvg hdisk1
#
```

4. On the virtual I/O client , remove the hdisk1 device:

```
# rmdev -l hdisk1 -d
hdisk1 deleted
#
```

5. On the Virtual I/O Server, remove the backing device:

```
$ rmbdsp -bd vioc_1_rootvg
vtscsi0 deleted
$
```

6. Remove the disk from the disk pool. If you receive an error message, such as in the following example, and you are sure that you have only one hdisk per storage pool, you can use the **deactivatevg** and **exportvg** commands.

> **Note:** If you use the **exportvg** command, it will delete all the logical volumes inside the volume group, and if your volume group contains more than one hdisk, the logical volumes on this hdisk are also affected. Use the **lspv** command to check. In this case, it is safe to use the **exportvg vioc_rootvg_1** command:
>
> ```
> $ lspv
> NAME            PVID                                VG              STATUS
> hdisk0          00c478de00655246                    rootvg          active
> hdisk1          00c478de008a399b                    rootvg          active
> hdisk2          00c478de008a3ba1                    vioc_rootvg_1   active
> hdisk3          00c478deb4b0d4b0                    None
> $
> ```

```
$ chsp -rm -f -sp vioc_rootvg_1 hdisk2


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
0516-882 reducevg: Unable to reduce volume group.
$ deactivatevg vioc_rootvg_1


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
$ exportvg vioc_rootvg_1
$
```

7. On the Virtual I/O Server, remove the hdisk device:

```
$ rmdev -dev hdisk2
hdisk2 deleted
$
```

8. Replace the physical disk drive.

9. On the Virtual I/O Server, configure the new hdisk device using the **cfgdev** command and check the configuration using the **lspv** command to determine that the new disk is configured:

   ```
   $ cfgdev
   $ lspv
   NAME           PVID                                VG              STATUS
   hdisk2         none                                None
   hdisk0         00c478de00655246                    rootvg          active
   hdisk1         00c478de008a399b                    rootvg          active
   hdisk3         00c478deb4b0d4b0                    None
   $
   ```

10. On the Virtual I/O Server, add the hdisk to the storage pool using the **chsp** command when you have more than one disk per storage pool. If you only have one storage pool per hdisk, use the **mksp** command:

    ```
    $ mksp vioc_rootvg_1 hdisk2
    vioc_rootvg_1
    0516-1254 mkvg: Changing the PVID in the ODM.
    $
    ```

11. On the Virtual I/O Server, make the backing device and attach it to the virtual device:

    ```
    $ mkbdsp -sp vioc_rootvg_1 32G -bd vioc_1_rootvg -vadapter vhost0
    vtscsi0 Available
    $
    ```

12. On the Virtual I/O Server, check that the backing device does not span a disk in the storage pool. In this case, we have only have one hdisk per storage pool.

13. On the virtual I/O client, reconfigure the new hdisk1:

    ```
    # cfgmgr -l vscsi1
    #
    Or if you do not know the parent device
    # cfgmgr
    #
    ```

14. On the virtual I/O client, extend the rootvg:

    ```
    # extendvg rootvg hdisk1
    0516-1254 extendvg: Changing the PVID in the ODM.
    #
    ```

15. On the virtual I/O client, mirror the rootvg:

    ```
    # mirrorvg -c 2 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:  hdisk0 hdisk1.
    #
    ```

16. On the virtual I/O client, initialize the boot record and set the bootlist:

    ```
    # bosboot -a

    bosboot: Boot image is 18036 512 byte blocks.
    # bootlist -m normal hdisk0 hdisk1
    ```

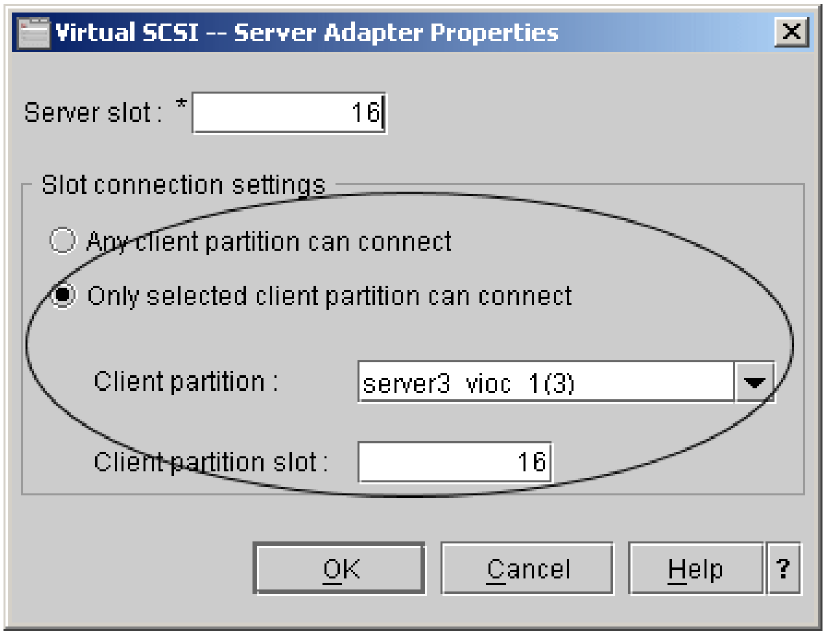
4.5 Managing multiple storage security zones

> **Note:** Security in a virtual environment depends on the integrity of the Hardware Management Console and the Virtual I/O Server. Access to the HMC and Virtual I/O Server must be closely monitored because they are able to modify existing storage assignments and establish new storage assignments on LPARs within the managed systems.

When planning for multiple storage security zones in a SAN environment, study the enterprise security policy for the SAN environment and the current SAN configuration.

If different security zones or disk subsystems share SAN switches, the virtual SCSI devices can share the HBAs, because the hypervisor firmware acts in a similar manner a SAN switch. If a LUN is assigned to a partition by the Virtual I/O Server, it cannot be used or seen by any other partitions. The hypervisor is designed in a way that no operation within a client partition can gain control of or use a shared resource that is not assigned to the client partition.

When you assign a LUN in the SAN environment for the different partitions, remember that the zoning is done by the Virtual I/O Server. Therefore in the SAN environment, assign all the LUNs to the HBAs used by the Virtual I/O Server. The Virtual I/O Server assigns the LUNs (hdisk) to the virtual SCSI server adapters (vhost) that are associated to the virtual SCSI client adapters (vscsi) used by the partitions. See Figure 4-6.



Virtual SCSI -- Server Adapter Properties

Server slot : * 16

Slot connection settings

○ Any client partition can connect

● Only selected client partition can connect

Client partition : server3 vioc 1(3)

Client partition slot : 16

Figure 4-6 Create virtual SCSI

If accounting or security audits are made from the LUN assignment list, you will not see the true owners of LUNs, because all the LUNs are assigned to the same HBA. This might cause audit remarks.

You can produce the same kind of list from the Virtual I/O Server by using the **`lsmap`** command, but if it is a business requirement that the LUN mapping is at the storage list, you must use different HBA pairs for each account and security zone. You can still use the same Virtual I/O Server, because this will not affect the security policy.

If it is a security requirement, and not an hardware issue, that security zones or disk subsystems do not share SAN switches, you cannot share an HBA. In this case, you cannot use multiple Virtual I/O Servers to virtualize the LUN in one managed system, because the hypervisor firmware will act as one SAN switch.

4.6 Moving an AIX 5L LPAR from one server to another

Best practices for moving an LPAR from one CEC to another

The following section outlines steps used during the development of this publication. We recommend that you verify these steps in a test environment prior to working with them in a production environment.

With careful planning, you can move an LPAR from one server CEC to another by shutting it down on the source system and activating it on the target. The key to moving a partition successfully is ensuring that the virtual device environment is exactly the same on both systems.

This capability can be used to balance load across servers and perform hardware maintenance with only minimal downtime.

In order to keep the virtual environment manageable, we make the following recommendations for moving partitions:

- Only use virtual devices in LPARs that will be moved.
- For virtual devices, use matching slot numbers on the source and target.
- Attach all disk storage to a shared fibre channel SAN.
- Make all FC LUNs visible to VIOS partitions in both servers.
- If using multipath drivers on the source VIOS partitions, use the same drivers on the target Virtual I/O Servers at the same software levels.
- For all VSCSI disks, use physical volumes as backing devices rather than logical volumes.

Note: Assign virtual devices for the client LPAR to be moved to the same slot numbers on both the source and target system. If the devices are not in the same slots on the virtual I/O client, manual intervention might be required to boot the LPAR on the target system and configure it in the network.

Following these recommendations will help keep the system manageable as the number of LPARs grows.

4.6.1 CPU and memory planning

Ideally, the CPU and memory resources would be identical on the source and target systems. Keeping these variables constant will make troubleshooting easier in the event that application behavior changes following the move. However, there might be situations where it is advantageous to configure the target system differently from the source.

For example, when planning for routine maintenance, it might make sense to configure a target LPAR with less CPU or memory in order to provide degraded service, or to service an off-peak load when some servers are temporarily offline.

When moving a partition to a server with a different processor speed, it might not be possible to configure exactly the same amount of resource. You must consider the resources you have available and the reason for the move when planning CPU and memory requirements on the target system.

4.6.2 Network planning

In order to move an LPAR without involving DNS or network routing changes, you must ensure that both the source and target LPAR have the same number of virtual Ethernet adapters, in the same slots, with access to the same virtual and physical IP address ranges and VLANs (when using 802.1Q).

> **Note:** If the virtual Ethernet adapters are not in the same slots in the source and target LPARs, new Ethernet devices will be created on the target, and the LPAR might require manual intervention to access the network.

If you require additional availability, there are two methods you can use, SEA failover or Network Interface Backup.

- Network Interface Backup was the first supported method and requires a configuration change on each AIX LPAR using the virtual networks.
- A new SEA failover method was introduced from Virtual I/O Server Version 1.2 that simplifies the configuration process as it is controlled from the Virtual I/O Servers. In addition, the SEA failover method supports the extension of 802.1Q VLAN tags from the physical network into the virtual network, as well as providing an option for clients that do not support Network Interface Backup, such as Linux LPARs. For this reason we would recommend the SEA failover method as the preferred choice.

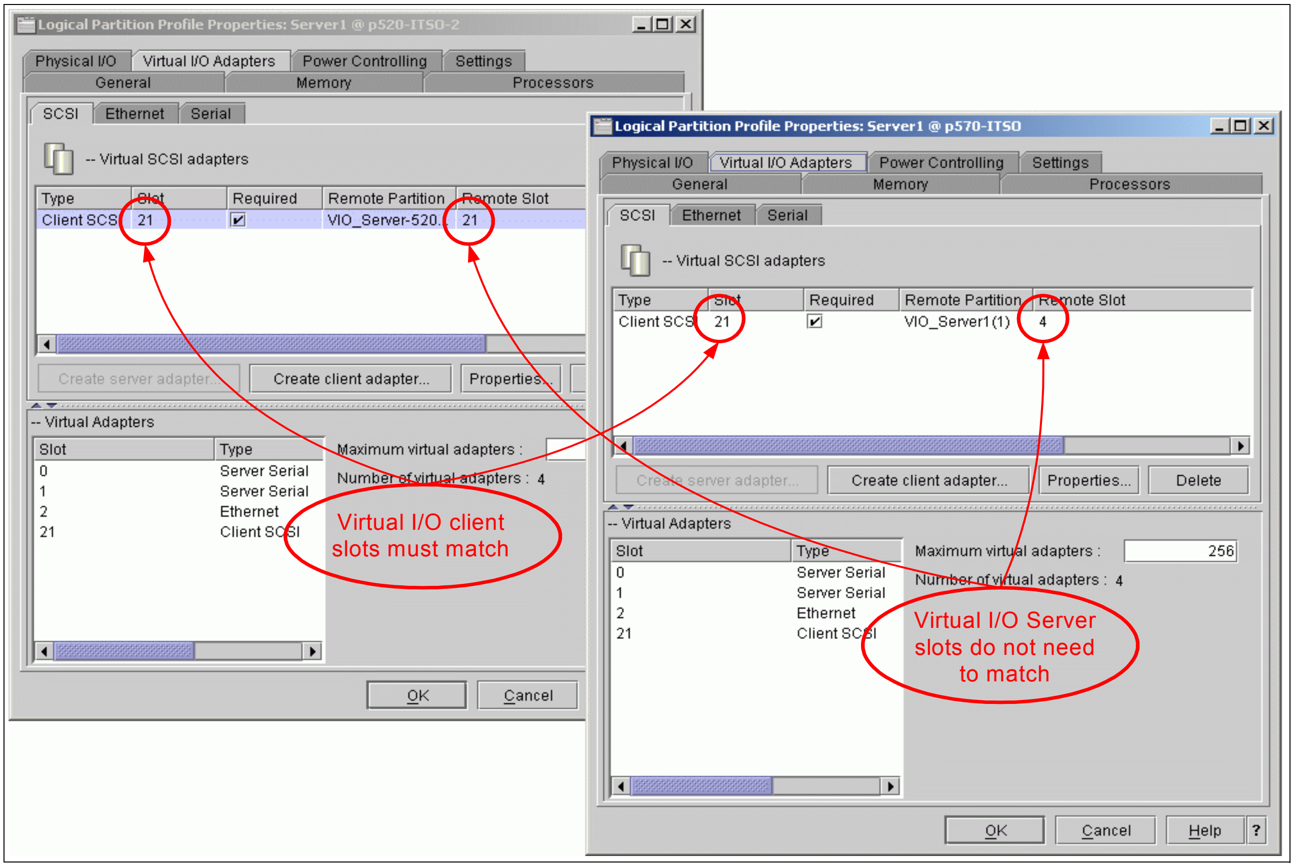
The MAC addresses of the virtual Ethernet adapters will change from the source to target LPARs. As a result, a gratuitous Address Resolution Protocol (ARP) is necessary to update the ARP caches of other hosts on the same subnet that were communicating with the LPAR prior to the move. The AIX 5L operating system generates gratuitous ARPs automatically as network addresses are configured to adapters on the target system. Because these are broadcast packets, it is possible for another host to drop them if the network or the host is extremely busy. In that case, pings to an affected host should update the cache.

4.6.3 Storage planning

Managing storage resources during an LPAR move can be more complex than managing network resources. Careful planning is required to ensure that the storage resources belonging to an LPAR are in place on the target system.

Virtual adapter slot numbers

Virtual SCSI adapters are tracked by slot number and partition ID on both the Virtual I/O Server and client. The source and target LPARs must have identical virtual SCSI adapter configurations, with the adapters in the same slot numbers. Although the VSCSI client slot numbers within the source and target client LPARs must be the same, the corresponding VSCSI server slot numbers in the Virtual I/O Server partitions on the source and target systems are not required to match. See Figure 4-7 on page 109 for an example of the HMC settings for example partitions. Note that the source and destination profiles are on two separate systems (CECs).



Type	Slot	Required	Remote Partition	Remote Slot
Client SCSI	21	☑	VIO_Server-520...	21

Slot	Type
0	Server Serial
1	Server Serial
2	Ethernet
21	Client SCSI

Maximum virtual adapters :

Number of virtual adapters : 4

Type	Slot	Required	Remote Partition	Remote Slot
Client SCSI	21	☑	VIO_Server1(1)	4

Slot	Type
0	Server Serial
1	Server Serial
2	Ethernet
21	Client SCSI

Maximum virtual adapters : 256

Number of virtual adapters : 4

Virtual I/O client slots must match

Virtual I/O Server slots do not need to match

Figure 4-7 HMC Partition Profile Properties Settings for source and target partitions

We recommend that you configure the Virtual I/O Server slot numbers identically on the source and target systems where possible. However, in a large environment where LPARs might be moved between many different servers, this can require tracking unique slot numbers across a large number of servers. As the number of servers increases, this approach becomes less practical. It is also impractical in preexisting environments where slot numbers overlap between systems. In those environments, we recommend an alternate approach:

- One alternate approach is to allocate the next available slot number on the target Virtual I/O Server, and track this in a spreadsheet, such as the one in Figure 4-2 on page 96.
- Another approach is to assign ranges of slot numbers to each server and increment the client slot number to compute the server slot, as shown in Figure 4-8 on page 110.

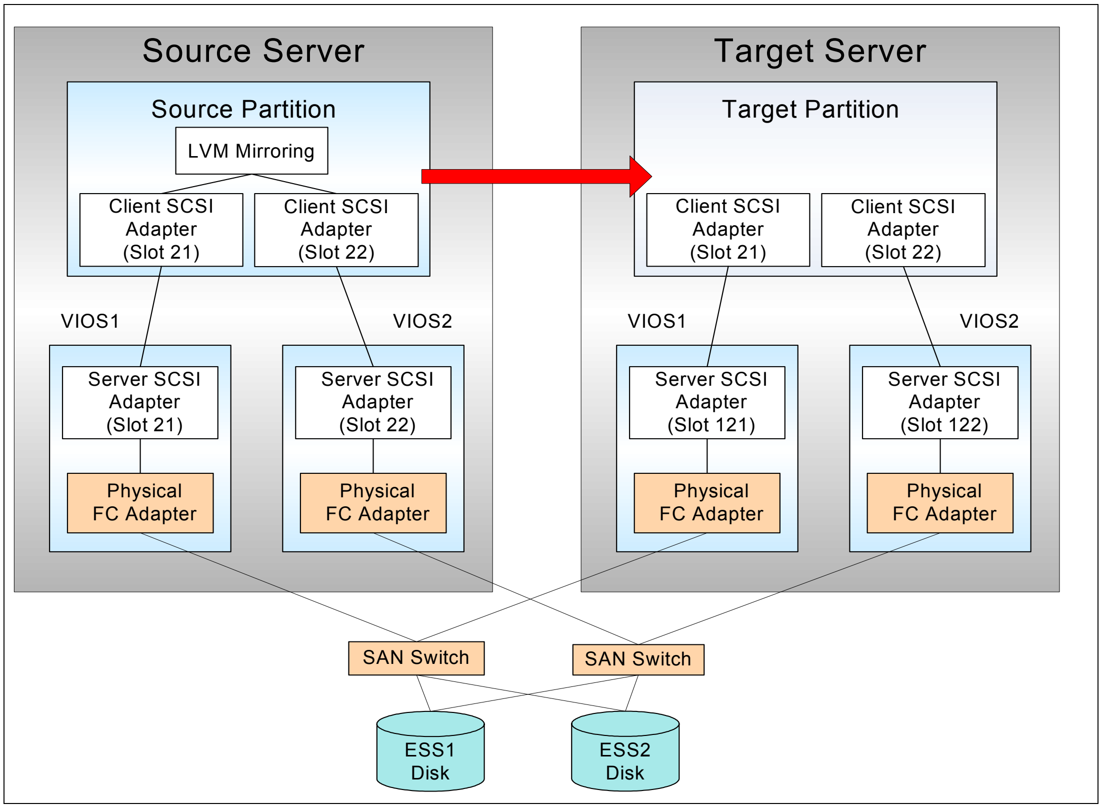

Source Server
Source Partition
LVM Mirroring
Client SCSI Adapter (Slot 21)
Client SCSI Adapter (Slot 22)
VIOS1
VIOS2
Server SCSI Adapter (Slot 21)
Server SCSI Adapter (Slot 22)
Physical FC Adapter
Physical FC Adapter
Target Server
Target Partition
Client SCSI Adapter (Slot 21)
Client SCSI Adapter (Slot 22)
VIOS1
VIOS2
Server SCSI Adapter (Slot 121)
Server SCSI Adapter (Slot 122)
Physical FC Adapter
Physical FC Adapter
SAN Switch
SAN Switch
ESS1 Disk
ESS2 Disk

Figure 4-8 Virtual storage configuration for a movable LPAR with dual VIOS and LVM mirroring

SAN considerations

All storage associated with an LPAR to be moved should be hosted on LUNs in a fibre channel SAN and exported as physical volumes within the Virtual I/O Server. The LUNs used by the LPAR to be moved must be visible to both the source and target Virtual I/O Servers. This can involve zoning, LUN masking, or other configuration of the SAN fabric and storage devices, which is beyond the scope of this document.

If multipath software is employed in the Virtual I/O Servers on the source system, the same multipath software must be in place on the target Virtual I/O Servers. For example, if SDDPCM (SDD or Powerpath) is in use on the source server, the same level must also be in use on the target server. Storage should not be migrated between different multipath environments during an LPAR move because this might affect the visibility of the unique tag on the disk devices.

Another important consideration is whether to allow concurrent access to the LUNs used by the LPAR. By default, the Virtual I/O Server acquires a SCSI reserve on a LUN when its hdisk is configured as a virtual SCSI target device. This means that only one VIOS can export the LUN to an LPAR at a time. The SCSI reserve prevents the LPAR from being started on multiple systems at once, which could lead to data corruption.

The SCSI reserve does make the move process more complicated, because configuration is required on both the source and target Virtual I/O Servers between the time that the LPAR is shut down on the source and activated on the target server.

Turning off the SCSI reserve on the hdisk devices associated with the LUNs makes it possible to move the LPAR with no configuration changes on the Virtual I/O Servers during the move. However, it raises the possibility of data corruption if the LPAR is accidentally activated on both servers concurrently.

Important: We recommend leaving the SCSI reserve active on the hdisks within rootvg to eliminate the possibility of booting the operating system on two servers concurrently, which could result in data corruption.

Query the SCSI reserve setting with the **lsdev** command and modify it with the **chdev** command on the Virtual I/O Server. The exact setting can differ for different types of storage. The setting for LUNs on IBM storage servers should not be `no_reserve`.

```
$ lsdev -dev hdisk7 -attr reserve_policy
value

no_reserve
$ chdev -dev hdisk7 -attr reserve_policy=single_path
hdisk7 changed
$ lsdev -dev hdisk7 -attr reserve_policy
value

single_path
```

Consult the documentation from your storage vendor for the reserve setting on other types of storage.

In situations where the LPAR normally participates in concurrent data access, such as a GPFS cluster, the SCSI reserve should remain deactivated on hdisks that are concurrently accessed. These hdisks should be in separate volume groups, and the reserve should be active on all hdisks in rootvg to prevent concurrent booting of the partition.

Backing devices and virtual target devices

The source and destination partitions must have access to the same backing devices from the Virtual I/O Servers on the source and destination system. Each backing device must have a corresponding virtual target device. The virtual target device refers to a SCSI target for the backing disk or LUN, while the destination server is the system to which the partition moving.

Important: Fibre channel LUNs might have different hdisk device numbers on the source and destination VIOS. The hdisk device numbers increment as new devices are discovered, so the order of attachment and number of other devices can influence the hdisk numbers assigned. Use the WWPN and LUN number in the device physical location to map corresponding hdisk numbers on the source and destination partitions.

Use the **lsmap** command on the source VIOS to list the virtual target devices that must be created on the destination VIOS and corresponding backing devices. If the vhost adapter numbers for the source VIOS are known, run the **lsmap** command with the **-vadapter** flag for the adapter or adapters. Otherwise, run the **lsmap** command with the **-all** flag, and any virtual adapters attached to the source partition should be noted. The following listing is for a DS4000 series device:

```
$ lsmap -all
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.570.107CD9E-V1-C4                      0x00000007

VTD                   lpar07_rootvg
LUN                   0x8100000000000000
Backing device        hdisk5
Physloc               U7879.001.DQD186N-P1-C3-T1-W200400A0B8110D0F-L0
```

The Physloc identifier for each backing device on the source VIOS can be used to identify the appropriate hdisk device on the destination VIOS from the output of the **lsdev -vpd** command. In some cases, with multipath I/O and multicontroller storage servers, the Physloc string can vary by a few characters, depending on which path or controller is in use on the source and destination VIOS.

```
$ lsdev -vpd -dev hdisk4
  hdisk4          U787A.001.DNZ00XY-P1-C5-T1-W200500A0B8110D0F-L0  3542     (20
0) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:  block
```

Make a note of the hdisk device on the destination VIOS that corresponds to each backing device on the source VIOS. This mapping will be required while the LPAR is inactive during the move process, as described in the following section, to create virtual target devices on the destination VIOS. An incomplete or incorrect mapping might extend the time the LPAR is inactive during the move.

4.6.4 Moving an LPAR

Before attempting to move an LPAR, it is important to verify that the configuration is in place and the environment is ready for the move. Perform the following steps:

1. Verify that the source and destination configurations are consistent using the HMC interface:
   - The destination CPU and memory should be appropriate for the expected workload, if not exactly the same as the source settings.
   - The destination virtual adapters and slot numbers within the LPAR must match the source. Slot numbers in the VIOS partitions are not required to match.
2. Verify that the same networks are available on the destination VIOS so that the LPAR can continue to use the same IP addresses.
3. Verify that all required hdisk devices are visible on the destination VIOS and that you have a mapping of which hdisk corresponds to each virtual target device.
4. Make a backup of the source LPAR prior to the move.

After completing the initial planning and setup, the process of moving an LPAR is relatively simple. To move an LPAR, perform the following steps:

1. Shut down the LPAR on the source system using either the operating system **shutdown** command or the HMC interface.
2. After verifying that the source LPAR is Not Activated, remove the virtual SCSI target devices from the source VIOS using the **rmdev** command:

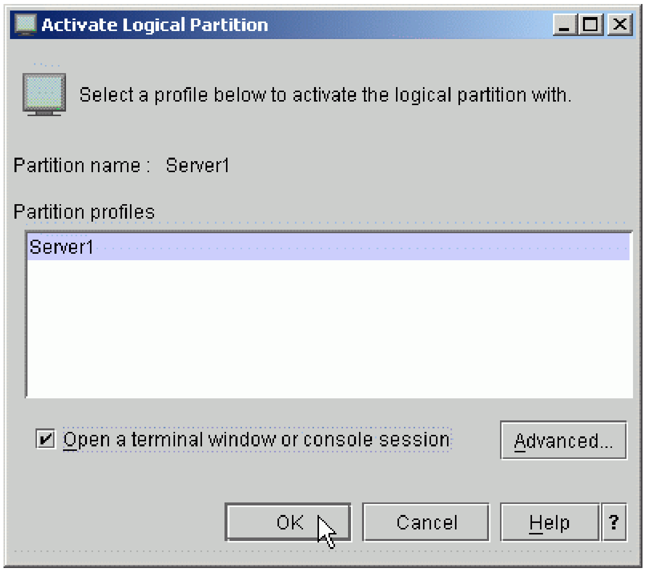
   ```
   $ rmdev -dev lpar07_rootvg
   lpar07_rootvg deleted
   ```

3. Using the just created list created, configure virtual SCSI target devices on the destination VIOS using the **mkvdev** command. Use the **lsmap** command to verify that all target devices are created successfully.

```
$ mkvdev -vdev hdisk4 -vadapter vhost0 -dev lpar07_rootvg
lpar07_rootvg Available
$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9111.520.10DDEDC-V1-C21                     0x00000000

VTD                   lpar07_rootvg
LUN                   0x8100000000000000
Backing device        hdisk4
Physloc               U787A.001.DNZ00XY-P1-C5-T1-W200500A0B8110D0F-L0
```

4. Activate the LPAR on the destination system using the HMC, and open a terminal window to the LPAR if one is not already open (Figure 4-9).



Select a profile below to activate the logical partition with.

Partition name : Server1

Partition profiles

Server1

☑ Open a terminal window or console session

Figure 4-9 Activate the partition and open a terminal window

5. Using the terminal window, verify that the LPAR boots to the operating system on the destination. The active console and boot list might need to be updated using the SMS menu on the first boot.

4.7 Planning and deploying MPIO

In a virtualized environment, the deployment of MPIO provides flexibility and maximizes system availability in an effective and efficient manner. This section discusses the general considerations and recommended parameters that need to be configured when planning a dual Virtual I/O Servers solution based on the deployment of MPIO. For the detailed step-by-step process of configuring dual Virtual I/O Servers and MPIO, refer to *Advanced POWER Virtualization on IBM System p5*, SG24-7940.

We base this section on the deployment of a recommended dual Virtual I/O Servers configuration, as shown in Figure 4-10, using vendor-specific device drivers on the Virtual I/O Servers and default AIX 5L MPIO on the virtual I/O client partitions.

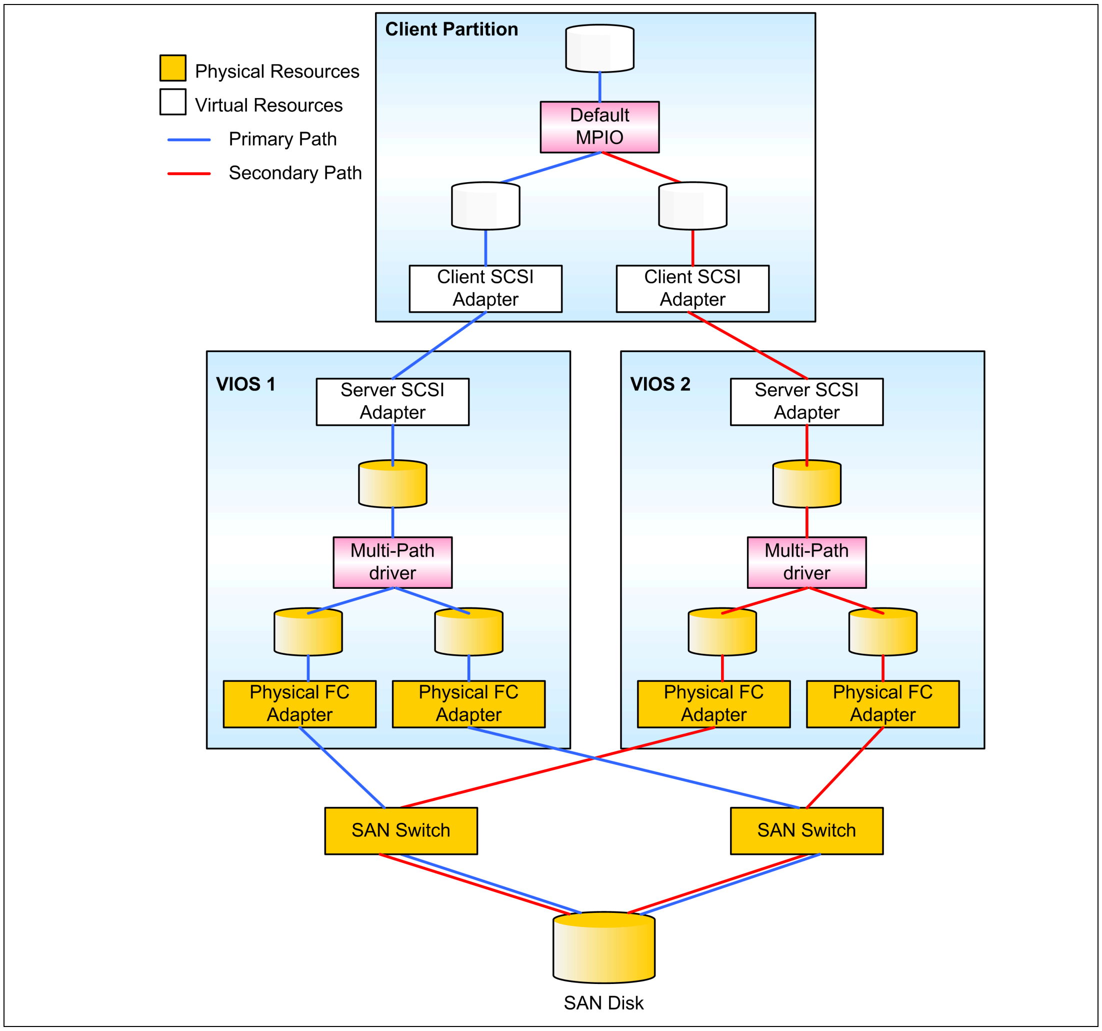


Figure 4-10 Recommended MPIO deployment using dual Virtual I/O Servers

4.7.1 Planning for a dual VIOS environment (MPIO)

MPIO best practices for dual Virtual I/O Servers

Note the following considerations when planning an MPIO-based environment using dual Virtual I/O Servers:

- The virtual adapters can be defined in the initial profile for the Virtual I/O Server or added dynamically as the client partitions are created. If the virtual I/O resources are being predefined on the Virtual I/O Server, set the virtual SCSI Server adapter to allow any remote partition and slot connection. After the client partition is created, you can update the Connection Information for the virtual SCSI Server adapter in the Virtual I/O Server profile to reflect the specific client partition and virtual slot number.
- If the virtual SCSI server adapters are added dynamically, the profile for the Virtual I/O Server must also be updated with the new adapters the next time the Virtual I/O Server is shut down and activated again.

- When using dynamic LPAR, virtual adapters can only be added or removed. When defining the partitions for the Virtual I/O Server, we recommend selecting the virtual adapters as Desired. This enables the adapter to be removed and added while the Virtual I/O Server is operational using dynamic LPAR.
- The physical adapters that are necessary for booting the Virtual I/O Server should be Required. All the other physical adapters can be Desired so that they can be removed or moved to other partitions dynamically.
- When creating the virtual adapters, it is also possible to set the virtual slot number rather than defaulting to the next available slot number. Refer to 4.3.2, "Virtual device slot numbers" on page 97 for the recommended method for virtual slot numbering.

4.7.2 Configuring the Virtual I/O Server

This section describes the configuration parameters required on the Virtual I/O Server when deploying MPIO in a dual Virtual I/O Servers environment.

Fibre channel SCSI device configuration

The fscsi devices include specific attributes that must be changed on both Virtual I/O Servers. These attributes are fc_err_recov and the dyntrk. Both attributes can be changed using the **chdev** command as follows:

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
```

Changing the fc_err_recov attribute to `fast_fail` will fail any I/Os immediately if the adapter detects a link event, such as a lost link between a storage device and a switch. The `fast_fail` setting is only recommended for dual Virtual I/O Servers configurations. Setting the dyntrk attribute to `yes` allows the Virtual I/O Server to tolerate cabling changes in the SAN. Both Virtual I/O Servers need to be rebooted for these changed attributes to take effect. Alternatively, the fscsi devices can be unconfigured and reconfigured for the settings to take effect.

SDDPCM and third-party multipathing software

For Virtual I/O Servers attached to IBM SAN storage (such as SVC, DS8000, DS6000, and ESS), Subsystem Device Driver Path Control Module (SDDPCM) multipathing software is installed on the Virtual I/O Servers to provide the load balancing and redundancy across multiple fibre channel connections between the Virtual I/O Servers and the IBM SAN Storage. Download the SDDPCM software from the following IBM Storage Web site:

http://www.ibm.com/support/docview.wss?uid=ssg1S4000201

In addition to IBM SAN storage, the Virtual I/O Server also supports many third-party SAN storage solutions and allows the installation of third-party, vendor-specific multipathing software. For the support matrix for all IBM and third-party storage solutions for the Virtual I/O Server, see the following Web site:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/VIOS_datasheet_063006.html

> **Important:** Third-party storage vendors provide their own recommendations relating to the installation of their storage devices on AIX 5L and the Virtual I/O Server. Refer to the vendor-specific installation guides for configuration instructions.

AIX 5L hdisk device configuration

Each disk must have the no_reserve setting.

To correctly enable the presentation of a physical drive to a client partition by dual Virtual I/O Servers, the reserve_policy attribute on each disk must be set to `no_reserve`. Using hdisk1 as an example, use the **chdev** command to change both the reserve policy and algorithm on the hdisk:

```
$ chdev -dev hdisk1 -attr reserve_policy=no_reserve
hdisk1 changed
```

In addition, to enable load balancing across multiple HBAs within the Virtual I/O Servers when using the base AIX 5L MPIO support, set the algorithm to `round_robin` for each physical drive:

```
$ chdev -dev hdisk1 -attr algorithm=round_robin
hdisk1 changed
```

The **lsdev** command confirms the changes made to the hdisk device:

```
$ lsdev -dev hdisk1 -attr
   attribute       value                            description
   user_settable

   PCM             PCM/friend/scsiscsd              Path Control Module          False
   algorithm       round_robin                      Algorithm                    True
   dist_err_pcnt   0                                Distributed Error Percentage True
   dist_tw_width   50                               Distributed Error Sample Time True
   hcheck_interval 0                                Health Check Interval        True
   hcheck_mode     nonactive                        Health Check Mode            True
   max_transfer    0x40000                          Maximum TRANSFER Size        True
   pvid            0021768a0151feb40000000000000000 Physical volume identifier   False
   queue_depth     3                                Queue DEPTH                  False
   reserve_policy  no_reserve                       Reserve Policy               True
   size_in_mb      18200                            Size in Megabytes            False
```

Virtual SCSI server support

For multipathing support of virtual SCSI devices in the AIX 5L client partition, the SAN LUN must be presented as a physical drive (hdiskx) from the Virtual I/O Server to the client partition. It is not possible to provide a large SAN LUN and then further subdivide it into logical volumes at the Virtual I/O Server level when using dual Virtual I/O Servers. The storage management for this configuration is performed in the SAN, so there is a one-to-one mapping of SAN LUNs on the Virtual I/O Servers to virtual SCSI drives on the client partition. For more information, see 4.1, “Managing and exporting physical storage on the VIOS” on page 90.

> **Important:** If each Virtual I/O Server has a different number of drives or the drives were zoned at different times, the device names (hdiskx) might be different between Virtual I/O Servers. Always check that the LUN IDs match when presenting a drive to the same client partition using dual Virtual I/O Servers. It is useful from an administration point of view to have the same device names on both Virtual I/O Servers.

There should not be a volume group created on the SAN LUNs on the Virtual I/O Server. To map a physical disk to the virtual SCSI server adapter (for example, vhost0), use the **mkvdev** command and specify the physical disk (for example, hdisk1) as the target device:

```
$ mkvdev -vdev hdisk1 -vadapter vhost0 -dev vtscsi0
vtscsi0 Available
```

4.7.3 Configuring the virtual I/O client

After installing both Virtual I/O Servers and mapping the virtual SCSI resources, install the operating system on the client partitions. The MPIO implementation requires additional configuration steps.

MPIO path management and load balancing

The MPIO support of virtual SCSI between client partitions and dual Virtual I/O Servers only supports failover mode. For any given virtual SCSI disk, a client partition will use a primary path to one Virtual I/O Server and fail over to the secondary path to use the other Virtual I/O Server. Only one path is used at a given time even though both paths can be enabled.

To balance the load of multiple client partitions across dual Virtual I/O Servers, the priority on each virtual SCSI disk on the client partition can be set to select the primary path and, therefore, a specific Virtual I/O Server. The priority is set on a per virtual SCSI disk basis, allowing increased flexibility in deciding the primary path for each disk. Due to this granularity, a system administrator can specify whether all the disks or alternate disks on a client partition use one of the Virtual I/O Servers as the primary path. The recommended method is to divide the client partitions between the two Virtual I/O Servers.

> **Important:** The priority path can be set per VSCSI LUN, enabling fine granularity of load balancing in dual VIOS configurations.

You can adjust the priority of individual paths.

Using an example where vscsi0 and vscsi1 are the virtual I/O client SCSI adapters for the client partition, use the **chpath** command on the client partition to set vscsi1 as the primary path by lowering the priority of the vscsi0 path:

```
# chpath -l hdisk1 -a priority=2 -p vscsi0
path Changed
```

To check the priority for each path, use the **lspath** command as follows:

```
# lspath -E -l hdisk1 -p vscsi0
priority 2 Priority True

# lspath -E -l hdisk1 -p vscsi1
priority 1 Priority True
```

The health check interval attribute for each virtual SCSI disk in the client partition that is being used with MPIO using dual Virtual I/O Servers must also be changed to enable automatic path failure detection. The default value for the hcheck_interval attribute is 0 (disabled). Using the **chdev** command on the client partition, change the health_interval attribute to 20 seconds for each virtual SCSI disk. The client partition must be rebooted for the attribute to take effect if the disk is part of a volume group or the volume group is varied offline.

```
# chdev -l hdisk1 -a hcheck_interval=20 -P
hdisk1 changed
```

Use the **lsattr** command to list attributes for the disk:

```
# lsattr -El hdisk1
PCM             PCM/friend/vscsi                  Path Control Module       False
algorithm       fail_over                         Algorithm                 True
hcheck_cmd      test_unit_rdy                     Health Check Command      True
hcheck_interval 20                                Health Check Interval     True
hcheck_mode     nonactive                         Health Check Mode         True
max_transfer    0x40000                           Maximum TRANSFER Size     True
```

```
pvid           00c5e9de252ef77f0000000000000000 Physical volume identifier False
queue_depth    100                              Queue DEPTH                True
reserve_policy no_reserve                       Reserve Policy             True
```

4.8 SCSI queue depth

Increasing the value of queue_depth, as shown in the previous section, to the default value of three might improve the throughput of the disk in some configurations. However, there are several other factors that must be taken into consideration. These factors include the value of the queue_depth attribute for all of the physical storage devices on the Virtual I/O Server being used as a virtual target device by the disk instance on the client partition. It also includes the maximum transfer size for the virtual SCSI client adapter instance that is the parent device for the disk instance.

The maximum transfer size for virtual SCSI client adapters is set by the Virtual I/O Server, which determines that value based on the resources available on that server and the maximum transfer size for the physical storage devices on that server. Other factors include the queue depth and maximum transfer size of other devices involved in mirrored volume group or MPIO configurations. Increasing the queue depth for some devices might reduce the resources available for other devices on that same parent adapter and decrease throughput for those devices.

The most straightforward configuration is when there is a physical LUN used as the virtual target device. In order for the virtual SCSI client device queue depth to be used effectively, it should not be any larger than the queue depth on the physical LUN. A larger value wastes resources without additional performance. If the virtual target device is a logical volume, the queue depth on all disks included in that logical volume must be considered. If the logical volume is being mirrored, the virtual SCSI client queue depth should not be larger than the smallest queue depth of any physical device being used in a mirror. When mirroring, the LVM writes the data to all devices in the mirror, and does not report a write as completed until all writes have completed; therefore, throughput is effectively throttled to the device with the smallest queue depth. This applies to mirroring on the Virtual I/O Server and the client.

Keep the same queue depth for all virtual disks on a volume group.

We recommend that you have the same queue depth on the virtual disk as the physical disk. If you have a volume group on the client that spans virtual disks, keep the same queue depth on all the virtual disks in that volume group. This is most important if you have mirrored logical volumes in that volume group, because the write will not complete before the data is written to the last disk.

In MPIO configurations on the client, if the primary path has a much greater queue depth than the secondary, there might be a sudden loss of performance as the result of a failover.

The virtual SCSI client driver allocates 512 command elements for each virtual I/O client adapter instance. Two command elements are reserved for the adapter to use during error recovery, and three command elements are reserved for each device that is open to be used in error recovery. The rest are left in a common pool for use in I/O requests. As new devices are opened, command elements are removed from the common pool. Each I/O request requires one command element for the time that it is active on the Virtual I/O Server.

Increasing the queue depth for one virtual device reduces the number of devices that can be open at one time on that adapter. It also reduces the number of I/O requests that other devices can have active on the Virtual I/O Server.

As an example, consider the case shown in Figure 4-11. In this scenario, we map a physical disk to a virtual disk.

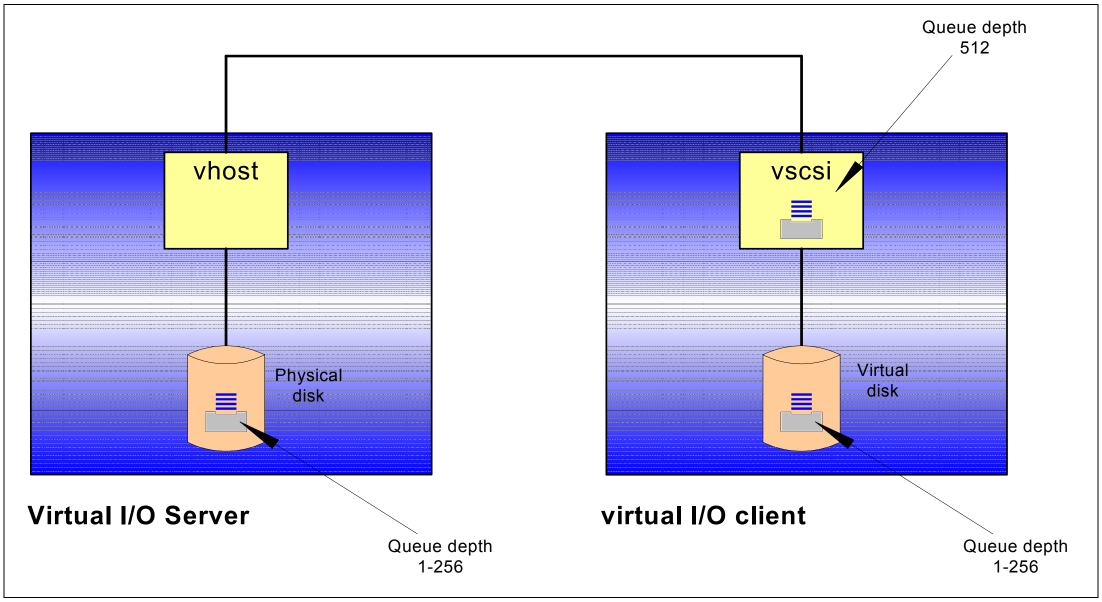


Figure 4-11 Physical disk mapped to a virtual disk

On the Virtual I/O Server, you can run the **lsdev -dev hdisk*N* -attr** command specifying the physical disk that you virtualized to virtual I/O client. Record the queue depth as shown in bold in Example 4-3.

Example 4-3 Using the lsdev command on the Virtual I/O Server

```
$ lsdev -dev hdisk2 -attr
attribute       value                            description
user_settable

PCM             PCM/friend/scsiscsd              Path Control Module           False
algorithm       fail_over                        Algorithm                     True
dist_err_pcnt   0                                Distributed Error Percentage  True
dist_tw_width   50                               Distributed Error Sample Time True
hcheck_interval 0                                Health Check Interval         True
hcheck_mode     nonactive                        Health Check Mode             True
max_transfer    0x40000                          Maximum TRANSFER Size         True
pvid            00cddeec68220f190000000000000000 Physical volume identifier    False
queue_depth     3                                Queue DEPTH                   False
reserve_policy  single_path                      Reserve Policy                True
size_in_mb      36400                            Size in Megabytes             False
$
```

On the virtual I/O client, run the **chdev -l hdiskN -a queue_depth=*x*** command, where ***x*** is the number that you recorded previously on the Virtual I/O Server. Using this approach, you will have a balanced the physical and virtual disk queue depth. Remember that the size of the queue depth will limit the number of devices that you can have on the virtual I/O client SCSI adapter because it partitions the available storage to the assigned resources.

The default queue depth of 3 might require tuning.

Figure 4-12 on page 120 shows another case. A logical volume on the Virtual I/O Server is mapped to a virtual disk on the virtual I/O client. Note that every logical volume on the Virtual I/O Server shares the same disk queue. The goal in fine-grained performance tuning is to avoid flooding the disk queue. Consider the example where you have a physical disk with a default queue depth of three. You also have three logical volumes virtualized on that disk, and all the virtual disks on the virtual I/O clients have a default queue depth of three. In this case, you might end up with nine pending I/Os on a queue depth of three. One of the solutions is to increase the queue depth of the physical disk to match the total of the virtual disk queue depth.

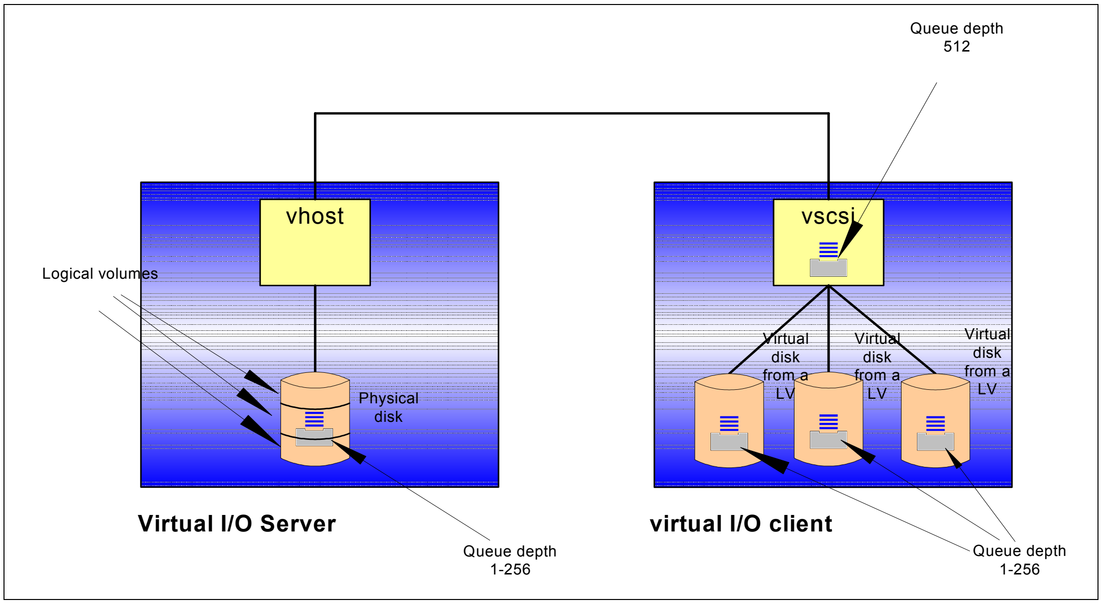


Figure 4-12 LV mapped to virtual disk

Command queuing best practices

Increasing the VSCSI client queuing can be a useful optimization when:

- The storage is fibre channel attached.

  SCSI queue depth is already a limiting factor using the default setting of three.

- The VSCSI device is attached using LUNs:
  - The Virtual I/O Server does not allow striping LVs across multiple disks/LUNs.
  - LVs could benefit when on a LUN with multiple disks and there is not great contention on the LUN from requests from multiple clients.
- The LUN contains multiple physical disks.

  The more disks in the LUN, the better the possibility of more I/Os in flight at the same time.

- The workload has enough processes or asynchronous I/O to drive a lot of outstanding I/O requests.

5

Performance and planning

Partitioning and the virtual I/O environment provide many ways to access system resources such as CPU, memory, networks, and storage. Deployment decisions can improve or impact the performance of the virtual I/O clients. This chapter highlights best practices for planning and configuring the virtual environment for performance.

5.1 Virtual processor configuration

In a micropartitioned environment, the configuration of virtual processors for any given partition can be controlled by the system administrator using the HMC. As such, deciding on the appropriate number of virtual processors to assign to a partition requires some planning and testing because this can affect the performance for both capped and uncapped partitions.

Processor folding helps use idle virtual processors.

There is processing in the hypervisor associated with the maintenance of online virtual processors, so consider their capacity requirements before choosing values for these attributes. However, AIX 5L Version 5.3 ML3 introduces the processor folding feature to help manage idle virtual processors. The kernel scheduler has been enhanced to dynamically increase and decrease the use of virtual processors in conjunction with the instantaneous load of the partition, as measured by the physical utilization of the partition.

Essentially, the kernel measures the load on the system every second and uses that value to determine whether virtual processors need to be enabled or disabled. The number of enabled virtual processors, and the number that can be enabled in a second, can be tuned using the **schedo** command setting, vpm_xvcpus.

A more detailed explanation is that once every second, AIX 5L calculates the CPU utilization consumed in the last one-second interval. This metric is called p_util. If p_util is greater than 80 percent of the capacity provided by the number of virtual CPUs currently enabled in the partition, then the value 1 + vpm_xvcpus additional virtual CPUs are enabled. If ceiling value (p_util + vpm_xvcpus) is less than the number of currently enabled virtual CPUs in the partition, then one virtual CPU is disabled. Therefore, each second, the kernel can disable, at most, one virtual CPU, and it can enable at most the value 1 + vpm_xvcpus virtual CPUs.

When virtual processors are deactivated, they are not dynamically removed from the partition as with dynamic LPAR. The virtual processor is no longer a candidate to run or receive unbound work. However, it can still run bound jobs. The number of online logical processors and online virtual processors that are visible to the user or applications does not change. There is no impact to the middleware or the applications running on the system because the active and inactive virtual processors are internal to the system.

The default value of the vpm_xvcpus tunable is 0, which signifies that folding is enabled. This means that the virtual processors are being managed. You can use the **schedo** command to modify the vpm_xvcpus tunable.

The following example disables the virtual processor management feature:

```
# schedo -o vpm_xvcpus=-1
Setting vpm_xvcpus to -1
```

To determine if the virtual processor management feature is enabled, use the following command:

```
# schedo -a | grep vpm_xvcpus
            vpm_xvcpus = -1
```

To increase the number of virtual processors in use by 1, use the following command:

```
# schedo -o vpm_xvcpus=1
Setting vpm_xvcpus to 1
```

Each virtual processor can consume a maximum of one physical processor. The p_util + vpm_xvcpus value is, therefore, rounded up to the next integer.

The following example describes how to calculate the number of virtual processors to use.

Over the last interval, partition A uses two and a half processors. The vpm_xvcpus tunable is set to 1. Using the previous equation:

- Physical CPU utilization = 2.5
- Number of additional virtual processors to enable (vpm_xvcpus) = 1
- Number of virtual processors needed = 2.5 + 1 = 3.5

Rounding up the value that was calculated to the next integer equals 4. Therefore, the number of virtual processors needed on the system is four. So, if partition A was running with eight virtual processors and the load remains constant, four virtual processors are disabled (over the next four seconds) and four virtual processors remain enabled. If simultaneous multithreading is enabled, each virtual processor yields two logical processors. So, eight logical processors are disabled and eight logical processors are enabled.

In the following example, a modest workload that is running without the folding feature enabled consumes a minimal amount of each virtual processor that is allocated to the partition. The following output using the **mpstat -s** command on a system with four virtual processors indicates the utilization for the virtual processor and the two logical processors that are associated with it:

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0           Proc2           Proc4           Proc6
     19.15%          18.94%          18.87%          19.09%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 11.09%   0.07%  10.97%   7.98%  10.93%   7.93%  11.08%   8.00%
```

When the folding feature is enabled, the system calculates the number of virtual processors needed with the preceding equation. The calculated value is then used to decrease the number of virtual processors to what is needed to run the modest workload without degrading performance.

The following output using the **mpstat -s** command on a system with four virtual processors indicates the utilization for the virtual processor and the two logical processors that are associated with it:

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0           Proc2           Proc4           Proc6
     54.63%           0.01%           0.00%           0.08%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 38.89%  15.75%   0.00%   0.00%   0.00%   0.00%   0.03%   0.05%
```

As you can determine from this data, the workload benefits from a decrease in utilization and maintenance of ancillary processors, and increased affinity when the work is concentrated on one virtual processor. When the workload is heavy, however, the folding feature does not interfere with the ability to use all the virtual processors, if needed.

Configure virtual processors for peak loads.

Enabling this feature can be very useful for uncapped partitions because it allows the partition to be configured with a larger number of virtual processors without significant performance issues. As a general rule, we recommend configuring a number of virtual processors that is reasonable to accommodate immediate spikes or rapid short-term growth requirements.

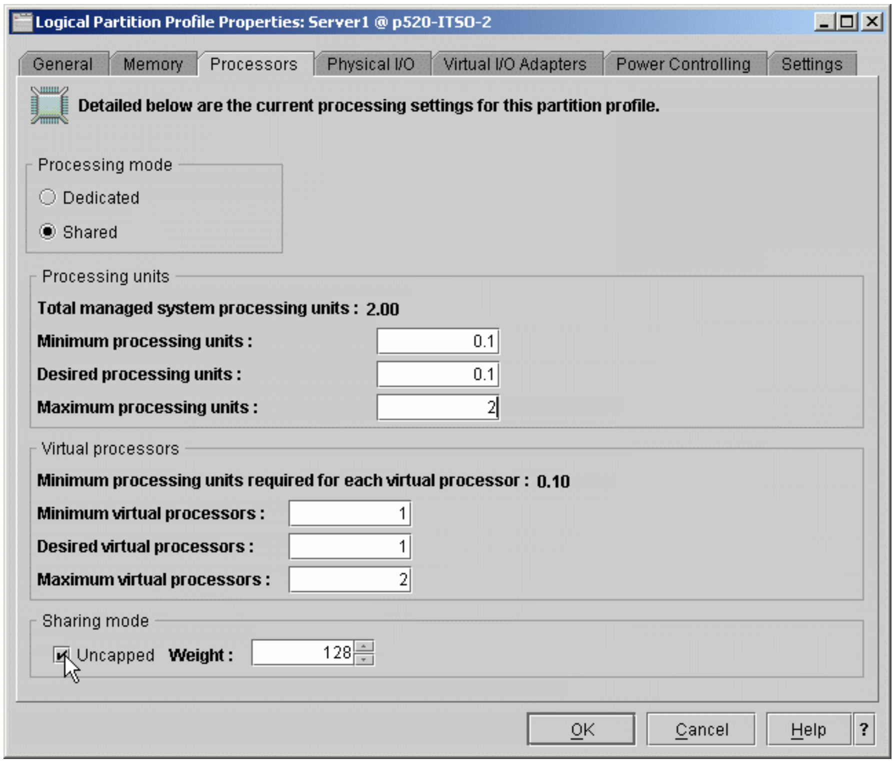
5.2 CPU sharing and weighting

The POWER Hypervisor manages the allocation of CPU resources to partitions within a server. Each partition is allocated CPU resources according to its system profile. All partitions that are not using dedicated processors share the same processor pool within a server. Partitions within the server can be configured in either a capped or uncapped mode. Capped partitions have a preset amount of maximum CPU resource.

Use the correct CPU settings in a partition profile.

When partitions are configured with uncapped CPU usage, they are able to consume all of their CPU allocation, plus any unused CPU cycles in the pool. Uncapping can provide a significant benefit to applications that have spikes in utilization.

When only one uncapped partition is exceeding its allocation, it can use all of the unused cycles. Uncapped partitions are assigned a uncapped weight in the HMC, as shown in Figure 5-1 on page 124. When multiple uncapped partitions exceed their allocation, the unused CPU cycles are allocated by the hypervisor according to the weights.



Detailed below are the current processing settings for this partition profile.

Processing mode

○ Dedicated

● Shared

Processing units

Total managed system processing units : 2.00

Minimum processing units : 0.1

Desired processing units : 0.1

Maximum processing units : 2

Virtual processors

Minimum processing units required for each virtual processor : 0.10

Minimum virtual processors : 1

Desired virtual processors : 1

Maximum virtual processors : 2

Sharing mode

☑ Uncapped Weight : 128

Figure 5-1 Uncapped weight for a logical partition

The hypervisor uses the weights of the partitions competing for unused CPU cycles to assign a proportion of the available cycles. If two partitions, both with a weight of 128, are competing for cycles, each partition will be given 128/(128+128)=1/2 of the available cycles.

Varying the weights alters the proportions accordingly. If one partition with a weight of 64 is competing with another partition with a weight of 128, the available cycles are assigned in proportion. The partition with weight 64 will receive 64/(64+128)=1/3 cycles, while the partition with weight 128 will receive 128/(64+128)=2/3 of the cycles.

When adding new partitions to an existing system or tuning the weight of existing partitions, it is important to consider the effects on other partitions on the system. If a partition belonging to an application consistently exceeds its guaranteed processor allocation, changing the weight of a different partition or adding another LPAR might reduce the amount of resource available for the application and affect performance.

To begin, set weights to 128 and then adjust in small increments.

When you first begin to deploy uncapped partitions, we recommend setting all weights to 128. Adjust the weights over time in small increments until you gain experience with the effects of the weights and uncapping to avoid side effects on other partitions.

Monitor system performance and note those partitions that consistently exceed their guaranteed CPU allocation. If any of those partitions host performance-sensitive applications, it might be necessary to increase their guaranteed allocation or their weights to avoid affects from other partitions.

5.3 Uncapped partitions and licensing

An uncapped partition enables the processing capacity of that partition to exceed its entitled capacity when the shared processing pool has available resources. This means that idle processor resource within a server can be used by any uncapped partition, resulting in an overall increase of the physical processor resource utilization. However, in an uncapped partition, the total number of virtual processors configured limit the total amount of physical processor resource that the partition can potentially consume. Using an example, a server has eight physical processors. The uncapped partition is configured with two processing units (equivalent of two physical processors) as its entitled capacity and four virtual processors. In this example, the partition is only ever able to use a maximum of four physical processors. This is because a single virtual processor can only ever consume a maximum equivalent of one physical processor. A dynamic LPAR operation to add more virtual processors would be required to enable the partition to potentially use more physical processor resource.

Consider software licensing when using uncapped partitions.

From a software licensing perspective, different vendors have different pricing structures on which they license their applications running in an uncapped partition. Because an application has the potential of using more processor resource than the partition’s entitled capacity, many software vendors that charge on a processor basis require additional processor licenses to be purchased simply based on the possibility that the application might consume more processor resource than it is entitled. When deciding to implement an uncapped partition, check with your software vendor for more information about their licensing terms.

5.4 Sizing your Virtual I/O Server

The only certain way to size any server, it can be argued, is to run it with the real workload, monitor, and then tune.

The following sections provide a starting point for tuning.

The IBM Systems Hardware Information Center has some detailed planning calculations to help with CPU and memory planning:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1_vios_planning.htm

The following guidelines are intended to help you started and might need additional adjustments after they are run and monitored in production, but they will enable you to reasonably accurately plan how much CPU and memory to use.

Most network and disk usage within a server is not constant. You will have bursts of network traffic and disk activity when actions are performed. For this reason, the Virtual I/O Servers are perfect candidates to use the shared CPU micropartition model and we recommend that you set up them as such.

To use the IBM Systems Workload Estimator tool to help size your server, see:

http://www-912.ibm.com/supporthome.nsf/document/16533356

5.4.1 Virtual I/O Server memory planning

The Virtual I/O Server, similar to other workloads, requires system memory to operate. Some of this memory is used to support network communications. This network memory is used to buffer network traffic, so detailed planning will involve knowing such factors as the messages sizes (MTU or jumbo frames) and also how many Shared Ethernet Adapters to use.

Simple guidelines exist for Virtual I/O Server memory usage.

The easiest rule to follow is that if you have a simple Virtual I/O Server that will only be bridging a couple of networks and will never be using jumbo frames or a larger MTU, 512 MB will be needed by the Virtual I/O Server. System configurations with 20-30 clients, many LUNs, and a generous vhost configurations might need increased memory to suit performance expectations.

If there is a possibility that you might start bridging more networks and use jumbo frames, use 1 GB of RAM for the Virtual I/O Servers. If you have enough memory to spare, the best situation is to use 1 GB for the Virtual I/O Servers to allow the maximum flexibility, but this is generally the maximum. When 10 Gbps Ethernet adapters are commonly implemented, this memory requirement can require revision.

5.4.2 Virtual I/O Server CPU planning

The CPU planning has two major parts:

- The CPU required to support the virtual network
- The CPU required to support the virtual disk

CPU requirements for network use in the Virtual I/O Server

The best way to plan any server CPU sizing is to start with a value, run, monitor, and then tune after observing a production workload. In this section, we provide a simple way to arrive at an appropriate starting value.

The first question to ask is how much network traffic will there be running through your Virtual I/O Server. We are not concerned about the number network adapters, just how busy these will be, because we need minimal CPU to support a network adapter card; but for every network packet, the CPU will have to calculate things such as the network checksum.

We summarize the starting point to use in Table 5-1.

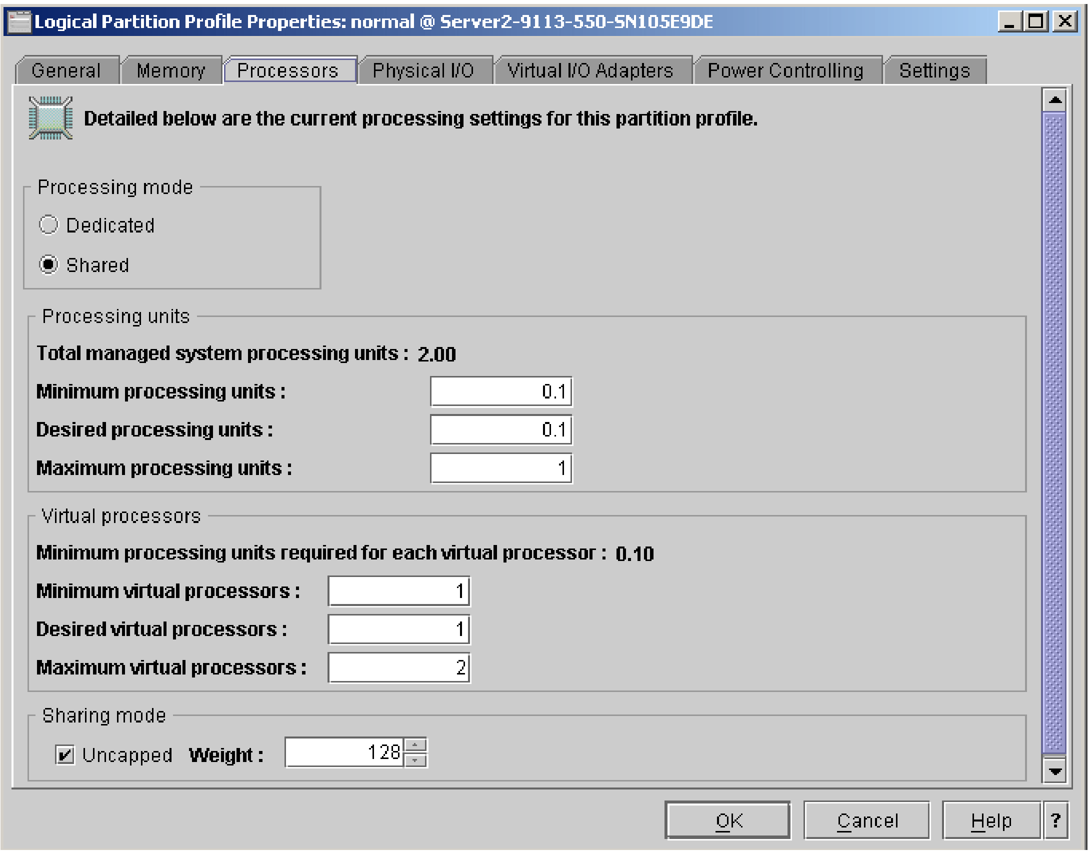
Table 5-1 For 1 GB of network traffic, an approximate amount of CPU needed in the VIOS

MTU (bytes)	1500	9000 or jumbo frames
CPU speed (GHz)		
1.5	1.1	0.55
1.65	1.0	0.5
1.9	0.87	0.44
2.2	0.75	0.38

For example, if you have a Virtual I/O Server with four gigabit network adapters and you know that during normal operations you have about 100 Mb of traffic overall. However, you also know that during a backup window you use a full gigabit of network traffic for two hours at night.

Examine the effects of network traffic in a partition.

These values can be translated into a CPU requirement to support the network. As discussed previously, this can be best done by using the shared CPU model. If we assume that your server has 1.65 GHz CPUs and you are using an MTU of 1500, we can calculate that during normal operation you only need 0.1 of a CPU. During peak loads, you would need a 1.0 CPU. If we assume that your user base is not using the system at night (thus the backup), there should be plenty of unused CPU in the free pool that can be used for the CPU requirement here, with a setup similar to that shown in Figure 5-2 on page 127.



Detailed below are the current processing settings for this partition profile.

Processing mode

○ Dedicated

● Shared

Processing units

Total managed system processing units : 2.00

Minimum processing units : 0.1

Desired processing units : 0.1

Maximum processing units : 1

Virtual processors

Minimum processing units required for each virtual processor : 0.10

Minimum virtual processors : 1

Desired virtual processors : 1

Maximum virtual processors : 2

Sharing mode

☑ Uncapped Weight : 128

Figure 5-2 Capacity planning: Network example

In this example, we guaranteed 0.1 of a CPU to sustain the daily network usage, but by using the uncapped CPU resources, we can allow the Virtual I/O Server to grow to 1.0 CPUs if required using spare CPU cycles from the CPU pool.

It is important to remember that you need CPU to support network traffic and that adding additional network cards (providing the same network traffic) does not require additional CPU. Using Table 5-1 on page 127, estimate a value for the required network bandwidth to support normal operations. Guarantee this amount (plus the disk value from the following section) to the logical partition as the processing units. You can then use uncapped CPUs on the servers profile to allow the Virtual I/O Server to use spare processing from the free pool to handle any spikes that might occur.

The Virtual I/O Server is designed to handle your most demanding enterprise computing networking needs.

CPU requirements for disk in the Virtual I/O Server

The disk CPU requirement is more difficult to work out accurately because it involves knowing detailed information about your I/O, such as block sizes and number of I/Os per second. If you know this information, the IBM Systems Hardware Information Center has the detailed planning calculations.

Disk I/O has a low impact on CPU usage in a virtualized environment.

For a rough guideline, it is probably easier to work out what the disks you are using can provide and make an estimate as to how busy you think these disks will be. For example, consider a simple Virtual I/O Server that has a basic internal set of four SCSI disks. These disks will be used to provide all of the I/O for the clients and are 10 K rpm disks. We will use a typical workload of 8 KB blocks. For these disks, a typical maximum is around 150 I/Os per second, so this works out to be about 0.02 of a CPU—a very small amount.

If we plot the amount of CPU required against the number of I/Os per second for the 1.65 GHz CPUs for the various I/O block sizes, we get a graph, as shown in Figure 5-3.

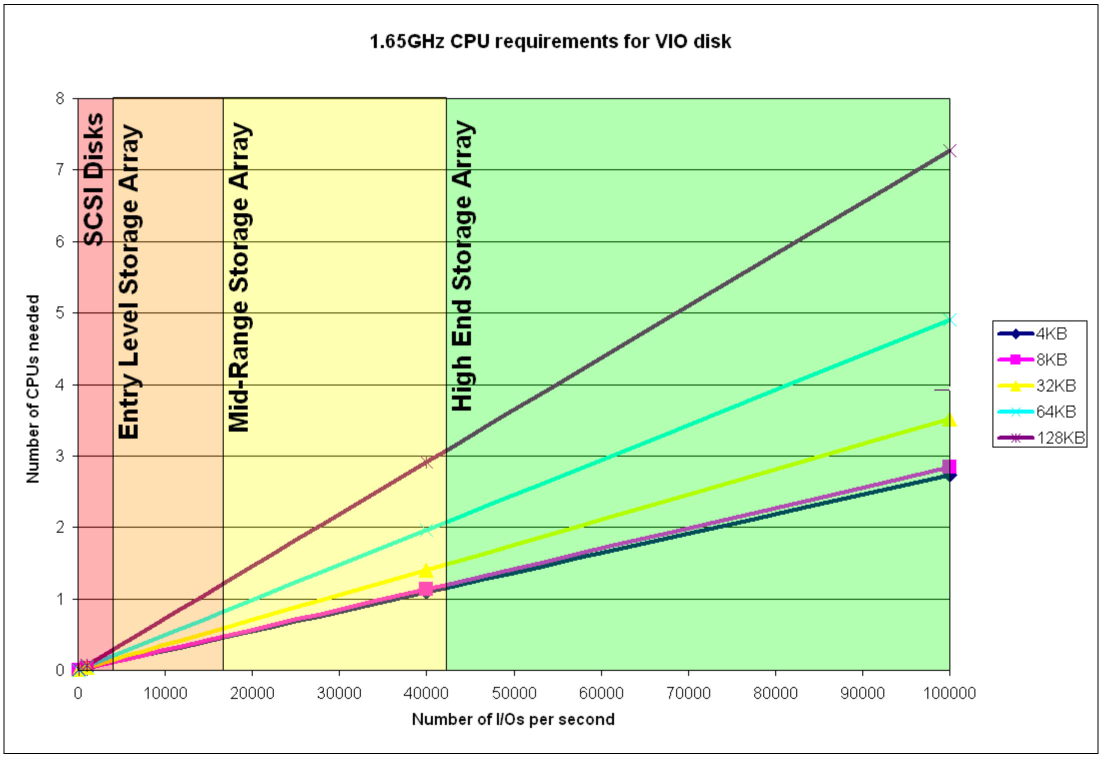


Figure 5-3 Estimated size of storage array needed to drive I/O versus Virtual I/O Server CPU

These I/O numbers and the storage subsystem sizing assume that the storage subsystem is being driven at 100% by the virtual I/O clients only (so every I/O is virtual disk) and the disk subsystems have only been placed on the graph to indicate the performance you would need to generate this sort of load. It is important to remember that we are not saying the bigger the disk subsystem, the more CPU power you need. What we actually find is that we need the most powerful storage subsystem offered to require any significant amount of CPU in the Virtual I/O Server from a disk usage perspective. As an example, a mid-range disk subsystem running at full speed will be in the region of the yellow/green boundary (mid-range to high end) on the graph when configured with more than 250 drives. In most cases, this storage will not be 100% dedicated to the Virtual I/O Server. However, if this was the case, even when running at full speed, a Virtual I/O will only need 1 CPU for most block sizes.

Most I/O requests for systems will be in the lower left section, so a starting figure of 0.1 to 0.2 CPU to cater for disk is a good starting point in most cases.

We always recommend testing a configuration before putting it into production.

5.4.3 Sizing dual Virtual I/O Servers

If you are planning a system with dual Virtual I/O Servers, you can size each Virtual I/O Server to support the workload of one half of the clients. Use uncapped CPU to provide enough reserve CPU in case a Virtual I/O Server is removed from the configuration for service.

For more detailed information about how to do this, see 4.7, “Planning and deploying MPIO” on page 113 and 3.8.2, “Dual Virtual I/O Servers enhanced availability options” on page 75.

5.4.4 Sizing the Virtual I/O Server summary

With most workloads, you expect the I/O to come in bursts. For example, when a client accesses an application, there will be a burst of disk and network activity while the data is sent to the client. When they submit a change to a Web site, for example, some network use will be generated with a burst of disk activity to update a backend database.

With this sporadic activity, making full use of the Micro-Partitioning and uncapped CPU features within the Virtual I/O Server makes the most sense. A guideline is to use 1 GB as a starting point for the Virtual I/O Server, and scale down or up from there. The CPU can take a bit more thought, but the only way to guarantee accuracy will be running the system under real life loads and then tuning. The previous estimates should help you find a sensible starting point for this tuning to begin.

If you plan dual Virtual I/O Servers, a bit of planning can make sure that you can size two smaller Virtual I/O Servers that support half of the virtual I/O clients each. Additional capacity to allow for Virtual I/O resilience can be provided through the uncapping facility of the servers.

5.5 Measuring VIOS performance

Measuring performance can be generally divided into:

- Short-term measuring used in testing, sizing, or troubleshooting scenarios, often initialized by a long-term measuring
- Long-term measuring used as capacity management input, because it includes performance trends or change in workload

5.5.1 Measuring short-term performance

Use these tools as starting points for short-term monitoring.

On the Virtual I/O Server, use the following tools for short-term performance measurements:

viostat Reports CPU statistics and I/O statistics for the entire system, adapters, tty devices, disks, and CD-ROM.

netstat Reports network statistics.

topas Reports selected local system statistics, such as CPU, network, process, and memory.

If you start short-term performance measuring on the Virtual I/O Server and you do not have a specific target, such as network degradation, start with the **topas** command. This tool provides an overall performance status of the Virtual I/O Server. The **topas** command provides you system statistics that highlight general issues that you can further examine with the **viostat** or **netstat** command. These commands provide more detailed output. Plan your performance measuring and document the time and status of your system and your assumptions, because it will be valuable information if you need to research a performance bottleneck.

In Example 5-1, you will see the output from the **topas** command.

Example 5-1 The topas command output with a 10 second interval

```
$ topas -interval 10
Topas Monitor for host:    lpar01               EVENTS/QUEUES    FILE/TTY
Mon Jul 10 08:40:37 2006   Interval: 10         Cswitch      83  Readch     2722
                                                Syscall      95  Writech      39
Kernel    2.6   |#                           |  Reads         4  Rawin         0
User      1.6   |#                           |  Writes        0  Ttyout       39
Wait      0.0   |                            |  Forks         0  Igets         0
Idle     95.9   |########################### |  Execs         0  Namei         5
Physc =  0.01                     %Entc=   6.6  Runqueue    0.1  Dirblk        0
                                                Waitqueue   0.0
Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out
en2       0.4      3.6     0.2     0.4     0.1  PAGING           MEMORY
lo0       0.0      0.2     0.2     0.0     0.0  Faults        8  Real,MB    1024
                                                Steals        0  % Comp     33.0
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  PgspIn        0  % Noncomp   6.9
hdisk3    0.0      0.0     0.0     0.0     0.0  PgspOut       0  % Client    6.9
hdisk2    0.0      0.0     0.0     0.0     0.0  PageIn        0
                                                PageOut       0  PAGING SPACE
WLM-Class (Active)     CPU%    Mem%  Disk-I/O%  Sios          0  Size,MB    1536
System                    1      17         0                    % Used      0.6
Unmanaged                 1      19         0   NFS (calls/sec)  % Free     99.3
                                                ServerV2       0
Name            PID CPU% PgSp Class             ClientV2       0   Press:
xmwlm        368860  0.2  0.8 System            ServerV3       0   "h" for help
topas        360476  0.1  0.9 System            ClientV3       0   "q" to quit
```

The output from the **viostat** command is similar to the output from the AIX 5L version of the **iostat** command. Example 5-2 shows an example of the output from the **viostat** command. This output can be interpreted similarly to the output from the AIX 5L version of **iostat**. For additional information, refer to the AIX 5L product documentation.

Example 5-2 The viostat command without parameters

```
$ viostat
System configuration: lcpu=2 drives=4 ent=0.10 paths=4 vdisks=3

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait physc % entc
          0.1          2.0                0.3   0.8   98.9      0.0   0.0    1.9

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk2           0.0       0.0       0.0          9     14428
hdisk3           0.0       0.2       0.0      18945    108860
hdisk0           0.0       0.4       0.1      21574    230561
hdisk1           0.0       0.3       0.1          6    230561
$
```

You can get more specialized output from the **viostat** command, similar to the output shown in Example 5-3 on page 131. Do not use this output for long-term measuring because it will require a lot of disk space to record. The output can be limited by putting the hdisk number in the parameter list, as we did in Example 5-3 on page 131.

Example 5-3 Extended disk output of hdisk2 and hdisk3 using viostat

```
$ viostat -extdisk hdisk2 hdisk3
System configuration: lcpu=2 drives=4 paths=4 vdisks=3

hdisk2         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0     22.2      0.0        0.1       22.2
               read:        rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.0      0.3   12.5             0          0
              write:        wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      9.4      1.6   18.1             0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0    0.0          0.0           7
hdisk3         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0    196.0      0.0       29.0      167.1
               read:        rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.6      0.2   11.5             0          0
              write:        wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.9      1.2   20.1             0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.8      0.0     30.0    0.0          0.0        5724
--------------------------------------------------------------------------------
$
```

The output from Virtual I/O Server version of the **netstat** command is very similar to the AIX 5L version of the **netstat** command. The Virtual I/O Server **netstat** command not only provides performance data, it also provides network information, such as routing table and network data. To show performance-related data, you must use the command with an interval, as shown in Example 5-4. Note that you must stop the command with a Ctrl+C, making it more difficult to use the **netstat** command in a long-term measuring script, because you only put in a interval and not a count, as you can with the **viostat** command.

Example 5-4 Output from the netstat command

```
$ netstat 1
    input   (en2)      output           input   (Total)    output
 packets  errs  packets  errs colls  packets  errs  packets  errs colls
 1038255     0    45442     0     0  1075627     0    83225      0     0
       9     0        1     0     0        9     0        1      0     0
       2     0        1     0     0        2     0        1      0     0
       3     0        1     0     0        3     0        1     0     0
       3     0        1     0     0        3     0        1     0     0
       8     0        1     0     0        8     0        1     0     0
^C$
```

5.5.2 Measuring long-term performance

Long-term performance monitoring can include Workload Manager.

The primary tools used in long-term performance measurement are:

wkldmgr The Workload Manager control command. It starts and stops the workload manager.

wkldagent The recording agent. It records system performance and Workload Manager metrics in a binary file in /home/ios/perf/wlm.

wkldout This formats the output from the **wkldagent** into a text format to the terminal.

You can use commands such as **viostat** and **netstat** as long-term performance tools on the Virtual I/O Server. They can benefit from being put in a script or started using the **crontab** command to form customized reports.

The tools provided by the Virtual I/O Server to perform long-term performance measuring are a subset from Workload Manager (WLM). When the Workload Manager components run in passive mode, they do not regulate the CPU, memory, or disk I/O utilization, but monitor them. The Virtual I/O Server resource allocation mechanisms are not affected.

To start performance monitoring using the Workload Manager-based commands, you must first start the Workload Manager with the **wkldmgr** command and then check the status, as shown in Example 5-5. This command does not start any recording or performance monitoring.

Example 5-5 Start wkldmgr and check the status

```
$ wkldmgr -start
$ wkldmgr -status
WorkLoad Manager is running
$
```

To start recording system performance data, use the **wkldagent** command. Example 5-5 shows the start of a recording session. You can check the status of the Workload Manager agent by using the **wkldagent** command or the **cat** command on the file /home/ios/perf/wlm/xmwlm.log1.

Example 5-6 Start the agent using wkldagent and check the status

```
$ wkldagent -start
$ wkldagent -status
WorkLoad Manager Agent is running
$ cat /home/ios/perf/wlm/xmwlm.log1

=====  xmtrend scheduled at Jul 10 10:50:42 =====
Initially logging to "/home/ios/perf/wlm/xmwlm.log1"
Initializing WLM and checking the state
WLM is turned on, activating the recording...
$
```

The output data is collected in a binary file, which is located in /home/ios/perf/wlm, in the format xmwlm.*yymmdd*, for example, xmwlm.060710. This file has to be converted into text format before it is readable by using the **wkldout** command. You can run this command while the **wkldagent** command is recording. For an example, see Example 5-7. Remember to specify the full path for the input and output file.

Example 5-7 Output from the wkldout command

```
$ ls -l /home/ios/perf/wlm
total 160
-rw-r--r--   1 root     staff         30872 Jul 10 11:09 xmwlm.060710
-rw-r--r--   1 root     staff           190 Jul 10 10:50 xmwlm.log1
$ wkldout -filename /home/ios/perf/wlm/xmwlm.060710
Time="2006/07/10 11:06:03", WLM/System/DISK/hardmax=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/softmax=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/min=0.00
Time="2006/07/10 11:06:03", WLM/System/DISK/shares=1.00
Time="2006/07/10 11:06:03", WLM/System/DISK/desired=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/consumed=0.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/hardmax=100.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/softmax=100.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/min=0.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/shares=1.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/desired=100.00
.
. (Lines omitted for clarity)
.
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/hardmax=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/softmax=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/min=0.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/shares=1.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/desired=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/consumed=0.00
Time="2006/07/10 11:09:06", WLM/System/tier=0.00
Time="2006/07/10 11:09:06", WLM/Shared/tier=0.00
Time="2006/07/10 11:09:06", WLM/Default/tier=0.00
Time="2006/07/10 11:09:06", WLM/Unmanaged/tier=0.00
Time="2006/07/10 11:09:06", WLM/Unclassified/tier=0.00
Time="2006/07/10 11:09:06", WLM/wlmstate=1.00
Time="2006/07/10 11:09:06", PagSp/%totalused=0.70
Time="2006/07/10 11:09:06", Proc/swpque=0.00
Time="2006/07/10 11:09:06", Proc/runque=0.93
Time="2006/07/10 11:09:06", CPU/glwait=0.00
Time="2006/07/10 11:09:06", CPU/gluser=1.56
Time="2006/07/10 11:09:06", CPU/glkern=2.28
$ ls -l /home/ios/perf/wlm
total 168
-rw-r--r--   1 root     staff         32924 Jul 10 11:12 xmwlm.060710
-rw-r--r--   1 root     staff         41015 Jul 10 11:11 xmwlm.060710_01
-rw-r--r--   1 root     staff           190 Jul 10 10:50 xmwlm.log1
$
```

If you run the **ls** command as shown in Example 5-7 on page 133, you find a xmwlm.yymmdd_01 file as the text output from the **wkldout** command. You can use the **more** command to show it in pages or use the **ftp** command to move it to a other server. Note that the binary files created by the **wkldagent** command are saved and each file will contain one day of data. If you want to save the files, you must copy them from the /home/ios/perf/wlm directory.

5.6 Simultaneous multithreading on a virtual I/O client

There are several different forms of multithreading. The form of multithreading implemented in the POWER5 architecture is named simultaneous multithreading and is a hardware design enhancement that enables two separate threads to execute on one processor simultaneously.

5.6.1 When simultaneous multithreading is beneficial

Simultaneous multithreading is available in a virtualized environment.

Simultaneous multithreading can be a good choice when the overall throughput is most important for a general workload. If most of the transactions are similar in execution needs, this workload is a good candidate for simultaneous multithreading. For example, a Web server, online transaction application severs, and database servers are all good candidates because:

- Most of the threads are similar in execution needs.
- In random data access, because threads must sleep for data to be loaded into cache or from disk.
- The overall throughput is the most important performance aspect.

Transaction types that have a very high cycles per instruction (CPI) count tend to use processor and memory resource poorly and usually see the greatest simultaneous multithreading benefit. These high CPIs are usually caused by high cache miss rates from a very large working set. Although most commercial transactions typically have this characteristic, they are dependent on whether the two simultaneous threads are shared instructions and data or are completely distinct. Transactions that share instructions or data, which include those that run mainly in the operation system or within a single application, tend to have a better simultaneous multithreading benefit.

Workloads with a low CPI and low cache miss rates see a smaller benefit. For high performance computing, try enabling simultaneous multithreading and monitor the performance. If the transaction type is data intensive with tight loops, you might see more contention for cache and memory, which can reduce performance.

5.6.2 When simultaneous multithreading might not be beneficial

Sometimes, simultaneous multithreading might not be the best choice for very specific workloads. Some transactions where the majority of individual software threads highly utilize resources in the processor or memory will benefit very little from simultaneous multithreading. For example, the transactions that are heavily floating-point intensive and have long execute times might perform better with simultaneous multithreading disabled.

Most commercial transaction workloads will benefit in performance and throughput using simultaneous multithreading.

5.6.3 Deciding when to use simultaneous multithreading on virtual I/O clients

The performance gain observed from simultaneous multithreading depends on the type of application and thread number. In most cases, the performance dramatically improves with simultaneous multithreading turned on.

We recommend running a simple test and measuring your production workloads.

In general, based on IBM tests, consider the following summary of rules for application performance in simultaneous multithreading environments:

- Transactions executed in typical commercial environments show a high performance gain with simultaneous multithreading enabled.
- The simultaneous multithreading improves performance when the number of transactions is increased.

Experiments on different transaction types have shown varying degrees of simultaneous multithreading gains, ranging from 19% to 35%. On average, most of the workloads showed a positive gain running in simultaneous multithreading mode.

> **Important:** System p5 servers provide a feature for AIX 5L V5.3 to dynamically switch between simultaneous multithreading and single threaded mode for transactions that might benefit from simultaneous multithreading.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 138. Note that some of the documents referenced here may be available in softcopy only.

- *Advanced POWER Virtualization on IBM System p5*, SG24-7940
- *Advanced POWER Virtualization on IBM @server p5 Servers: Architecture and Performance Considerations*, SG24-5768
- *NIM: From A to Z in AIX 4.3*, SG24-5524
- *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061
- *Hardware Management Console (HMC) Case Configuration Study for LPAR Management*, REDP-3999

Online resources

These Web sites are also relevant as further information sources:

- 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

- Latest *Multipath Subsystem Device Driver User's Guide*

  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303

- IBM System Planning Tool

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/

- 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

- Capacity on Demand

  http://www.ibm.com/systems/p/cod/

- 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

- Virtual I/O Server supported hardware

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/VIOS_datasheet_063006.html

- Virtual I/O Server documentation

  http://techsupport.services.ibm.com/server/vios/documentation/home.html

- IBM Systems Workload Estimator

  http://www-912.ibm.com/supporthome.nsf/document/16533356

- IBM Systems Hardware Information Center

  http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp

- IBM eServer pSeries and AIX Information Center

  http://publib16.boulder.ibm.com/pseries/index.htm

- Virtualization wiki

  http://www-941.ibm.com/collaboration/wiki/display/virtualization/Home

- IBM Advanced POWER Virtualization on IBM System p Web page

  http://www.ibm.com/systems/p/apv/

How to get IBM Redbooks

You can search for, view, or download Redbooks, 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

IBM System p Advanced POWER Virtualization Best Practices

Redpaper

A collection of recommended practices created to enhance your use of the Advanced POWER Virtualization feature

Builds on the knowledge found in existing IBM System p publications

A valuable refereence for experienced system administrators and architects

This IBM Redpaper provides best practices for planning, installing, maintaining, and operating the functions available using the Advanced POWER Virtualization feature 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 IBM POWER5 and POWER5+ processor-based systems. The main technologies are:

- Virtual Ethernet
- Shared Ethernet Adapter
- Virtual SCSI server
- Micro-Partitioning

This Redpaper can be read from start to finish, but it is meant to be read as a notebook where you access the topics that pertain best to you. This paper begins where *Advanced POWER Virtualization on IBM System p5*, SG24-7940, ends by adding additional samples and scenarios harvested by a select team that works at client and outsourcing sites, running both small and large installations. The experiences contained within are select best practices from real-life experience.

A working understanding of the Advanced POWER Virtualization feature and logical partitioning and IBM AIX 5L is required, as well as a basic understanding of network and VLAN tagging.