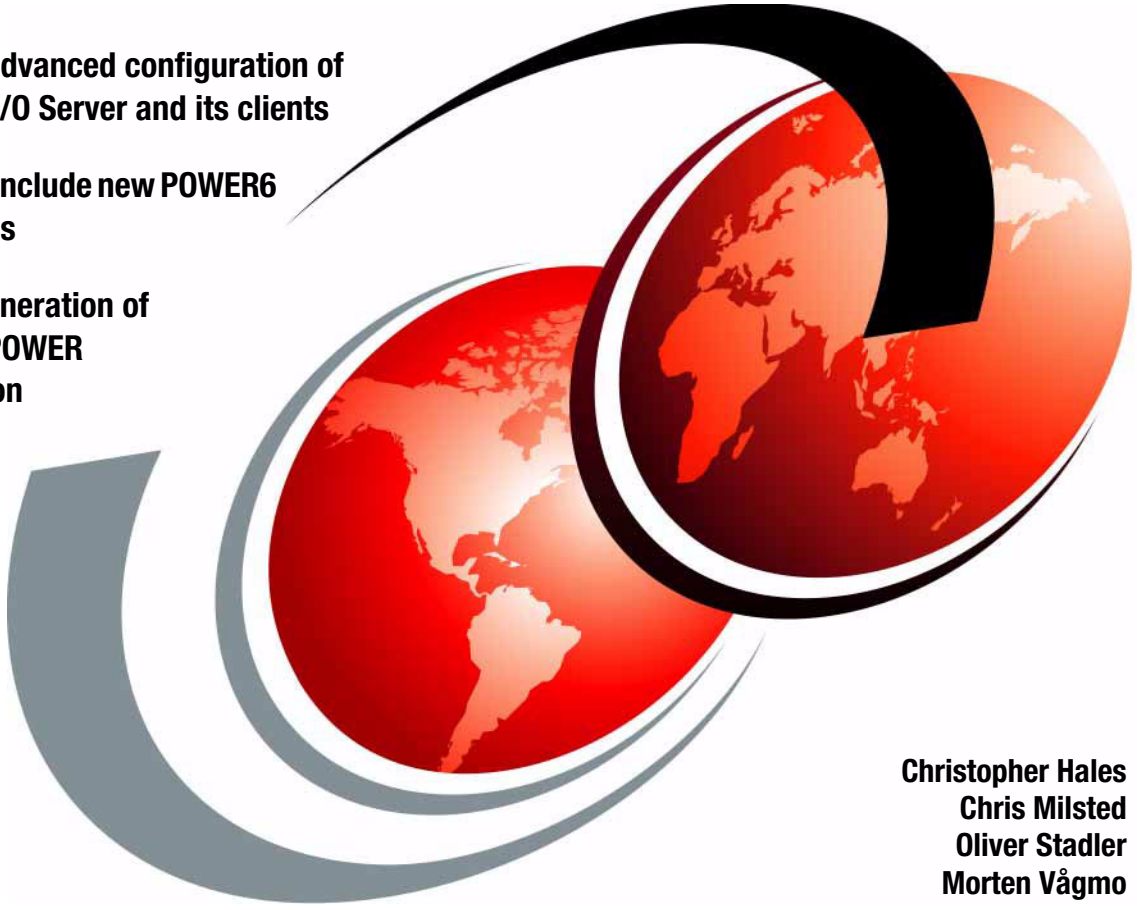
IBM

PowerVM Virtualization on IBM System p:
Introduction and Configuration
Fourth Edition

- Basic and advanced configuration of the Virtual I/O Server and its clients
- Updated to include new POWER6 technologies
- The next generation of Advanced POWER Virtualization



Christopher Hales
Chris Milsted
Oliver Stadler
Morten Vågmo

Redbooks

ibm.com/redbooks



International Technical Support Organization

PowerVM Virtualization on IBM System p: Introduction and Configuration

May 2008

SG24-7940-03

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

Fourth Edition (May 2008)

This edition applies to:
Version 6, Release 1, service pack 1 of AIX (product number 5765-G62)
Version 1, Release 5, (fixpack 10.1) of the Virtual I/O Server (product number 5765-G34)
Version 7, Release 3, Modification 2 (plus fix MH1062) of the HMC
Version 2, Release 7, Modification 320 of the IBM System Planning Tool
Version SF240, release 332 of the POWER5 System Firmware
Version EM320, release 031 of the POWER6 System Firmware
The hardware was POWER5 and POWER6 IBM System p servers.

© Copyright International Business Machines Corporation 2004, 2005, 2007, 2008. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xiii

Notices xv
Trademarks xvi

Preface xvii
The team that wrote this book xvii
Become a published author xix
Comments welcome xix

Summary of changes xxi
May 2008, Fourth Edition xxi

Chapter 1. Introduction 1
1.1 The value of virtualization on System p 2
1.2 PowerVM on IBM System p servers 3
1.2.1 What is PowerVM? 3
1.2.2 PowerVM Editions 4
1.2.3 POWER Hypervisor 5
1.2.4 Simultaneous multithreading 6
1.2.5 Logical partitions and processor capacity 6
1.2.6 LPAR and dynamic reconfiguration 9
1.2.7 Virtual Ethernet 9
1.2.8 Virtual SCSI 9
1.2.9 Virtual I/O Server 10
1.2.10 Integrated Virtualization Manager 10
1.2.11 Live Partition Mobility 11
1.2.12 Multiple operating system support 11
1.3 Capacity Upgrade on Demand for IBM System p 14
1.4 Availability of virtualized systems 15
1.4.1 Serviceability in virtualized environments 16
1.5 Security in a virtualized environment 18
1.6 Summary of PowerVM 18

Chapter 2. Virtualization technologies on System p servers 19
2.1 Editions of the PowerVM feature 20
2.1.1 PowerVM Standard Edition feature 20

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

2.1.2 PowerVM Enterprise Edition feature 24
2.1.3 Summary of PowerVM feature codes 26
2.2 Introduction to the POWER Hypervisor 27
2.2.1 POWER Hypervisor virtual processor dispatch 28
2.2.2 POWER Hypervisor and virtual I/O 32
2.2.3 System port (virtual TTY/console support) 33
2.3 Overview of Micro-Partitioning technologies 33
2.3.1 Micro-partitions 34
2.3.2 POWER5 physical shared-processor pool 40
2.3.3 POWER6 Multiple Shared-Processor Pools 44
2.3.4 Examples of Multiple Shared-Processor Pools 57
2.3.5 Shared dedicated capacity 63
2.4 Virtual I/O Server 63
2.4.1 Supported platforms 64
2.4.2 Storage virtualization 64
2.4.3 Shared Ethernet Adapter 66
2.4.4 Network security 66
2.4.5 Command line interface 67
2.4.6 Hardware Management Console integration 67
2.4.7 System Planning Tool support 67
2.4.8 Performance Toolbox support 68
2.4.9 Integrated Virtualization Manager 68
2.4.10 Tivoli support 68
2.5 Integrated Virtualization Manager 70
2.5.1 IVM setup guidelines 70
2.5.2 Partition configuration with IVM 72
2.6 Virtual SCSI introduction 75
2.6.1 Partition access to virtual SCSI devices 76
2.6.2 General virtual SCSI considerations 81
2.7 Virtual and Shared Ethernet introduction 83
2.7.1 Virtual LAN 84
2.7.2 Inter-partition networking with virtual Ethernet 91
2.7.3 Sharing physical Ethernet adapters 92
2.7.4 Virtual and Shared Ethernet configuration example 97
2.7.5 PowerVM Enterprise Edition and virtual Ethernet 102
2.8 Linux virtual I/O concepts 103
2.8.1 Linux device drivers for IBM System p virtual devices 104
2.8.2 Linux as a VIO client 104
2.8.3 Considerations 109
2.8.4 Further reading 109
2.9 Software licensing in a virtualized environment 109
2.9.1 Software licensing methods for UNIX operating systems 110
2.9.2 Licensing factors in a virtualized system 110

2.9.3 Capacity capping of LPARs 111
2.9.4 License planning and license provisioning of IBM software 114
2.9.5 Sub-capacity licensing for IBM software 116
2.9.6 Linux operating system licensing 118
2.10 Introduction to simultaneous multithreading 118
2.10.1 POWER processor SMT 119
2.10.2 SMT and AIX 119
2.10.3 SMT control in Linux 122
2.10.4 Capacity Upgrade on Demand 123
2.10.5 Instruction retry recovery 126
2.10.6 Dynamic processor de-allocation and processor sparing 126
2.10.7 Dynamic resources 126
2.10.8 Shared processor considerations 128
2.11 Integrated Virtual Ethernet overview 130

Chapter 3. Setting up virtualization: the basics 133
3.1 Getting started 134
3.1.1 Command line interface 134
3.1.2 Hardware resources managed 138
3.1.3 Software packaging and support 139
3.2 Virtual I/O Server enhancements and updates 141
3.2.1 Tracking the latest virtualization enhancements 141
3.2.2 Updating the Virtual I/O Server using fix packs 152
3.3 Creating a Virtual I/O Server partition 153
3.3.1 Defining the Virtual I/O Server partition 153
3.4 Virtual I/O Server software installation 173
3.5 Basic Virtual I/O Server scenario 178
3.5.1 Creating a Shared Ethernet Adapter 179
3.5.2 Defining virtual disks 182
3.5.3 Virtual SCSI optical devices 190
3.5.4 Mirroring the Virtual I/O Server rootvg 195
3.5.5 Creating client partitions 197
3.5.6 AIX client partition installation 207
3.5.7 Dynamic partitioning for virtual SCSI devices 214
3.5.8 Dedicated donating processors 215
3.6 Interaction with UNIX client partitions 218
3.6.1 Virtual SCSI resources 218
3.6.2 Virtual Ethernet resources 221
3.6.3 Simplification by using system plans and System Planning Tool . . 222

Chapter 4. Advanced virtualization configurations 243
4.1 Providing continuous availability for virtual I/O 244
4.2 Virtual SCSI redundancy 245
4.2.1 MPIO in the client partition 246
4.2.2 MPIO in the Virtual I/O Server 249
4.2.3 LVM mirroring in the client partition 251
4.3 Shared Ethernet redundancy 251
4.3.1 Shared Ethernet Adapter failover 252
4.3.2 Network Interface Backup in the client partition 255
4.3.3 When to use SEA failover or Network Interface Backup 256
4.3.4 Using Link Aggregation on the Virtual I/O Server 258
4.4 System maintenance with dual Virtual I/O Servers 260
4.5 Separating and balancing disk and network traffic 262
4.6 Scenario 1: Shared Ethernet Adapter failover 264
4.6.1 Configuring Shared Ethernet Adapter failover 265
4.7 Scenario 2: Network Interface Backup in the client 270
4.7.1 Configuring Network Interface Backup 271
4.7.2 Testing Network Interface Backup 273
4.8 Scenario 3: MPIO in the client with SAN 275
4.8.1 Configuring MPIO 277
4.8.2 Configuring MPIO in the client partitions 282
4.8.3 Testing MPIO 286
4.9 Scenario 4: LVM mirroring 288
4.9.1 Configuring LVM mirroring 289
4.9.2 Testing LVM mirroring 294
4.10 Linux on POWER network installation 296
4.10.1 Using an AIX server for Linux installation 297
4.10.2 Using a Linux installation server for installation 298
4.11 Using Linux on POWER with two Virtual I/O Servers 299
4.11.1 Ethernet connection bonding 299
4.11.2 Software RAID mirroring 303
4.11.3 Device mapper multipath 306
4.12 Configuring Multiple Shared-Processor Pools 308
4.12.1 Shared-Processor Pool management using the HMC GUI 310
4.12.2 Shared-Processor Pool management using the command line . . 315
4.13 Extending VLANs into virtual networks 317
4.13.1 POWER Hypervisor switch implementation 318
4.13.2 Scenario for setting up a VLAN 320
4.13.3 Configuring the client partitions 322
4.13.4 Configuring the Virtual I/O Server 323
4.13.5 Ensuring VLAN tags are not stripped on the Virtual I/O Server . . 325
4.13.6 Configuring the Shared Ethernet Adapter for VLAN use 325
4.13.7 Extending multiple VLANs into client partitions 327

4.13.8 Virtual Ethernet and SEA considerations 329
4.14 Supported configurations 330
4.14.1 Supported virtual SCSI configurations 331
4.14.2 HACMP for virtual I/O clients 341
4.14.3 Concurrent disks in client partitions 345
4.14.4 General Parallel Filesystem 348

Abbreviations and acronyms 349

Related publications 353
IBM Redbooks 353
Other publications 354
Online resources 354
How to get Redbooks 356
Help from IBM 356

Index 357

Figures

1-1 Architectural overview of POWER6-based server with MSPPs defined . . . 8
1-2 Redundant components in a virtualized system 17
2-1 Example of virtualization activation codes Web site 22
2-2 HMC window to activate PowerVM Standard Edition 23
2-3 ASMI menu to enable the Virtualization Engine Technologies 24
2-4 HMC window to enter the PowerVM Enterprise Edition activation code . . 26
2-5 The POWER Hypervisor abstracts the physical server hardware. 28
2-6 Virtual processor to physical processor mapping: pass 1 and pass 2 . . . 30
2-7 Micro-Partitioning processor dispatch 31
2-8 POWER5 physical shared processor pool and micro-partitions 41
2-9 Distribution of processor capacity entitlement on virtual processors 42
2-10 Example of capacity distribution of a capped micro-partition 43
2-11 Example of capacity distribution of an uncapped micro-partition 44
2-12 Overview of the architecture of Multiple Shared-Processor Pools 45
2-13 Redistribution of ceded capacity within Shared-Processor Pool$_1$ 48
2-14 Example of Multiple Shared-Processor Pools. 49
2-15 POWER6-based server with two Shared-Processor Pools defined 52
2-16 The two levels of unused capacity redistribution. 53
2-17 Example of a micro-partition moving between Shared-Processor Pools 56
2-18 Example of a Web-facing deployment mapped onto a set of micro-partitions 58
2-19 Web deployment using within a Shared-Processor Pool 59
2-20 Capped Shared-Processor Pool offering database services 61
2-21 Example of a system with Multiple Shared-Processor Pools 62
2-22 Simple Virtual I/O Server configuration 64
2-23 Integrated Virtualization Manager configuration on a POWER6 server . 71
2-24 Virtual SCSI architecture overview 78
2-25 Queue depths and virtual SCSI considerations 79
2-26 Logical Remote Direct Memory Access 80
2-27 Example of VLANs 85
2-28 The VID is placed in the extended Ethernet header 87
2-29 adapters and interfaces with VLANs (left) and LA (right) 91
2-30 Connection to external network using routing 93
2-31 Shared Ethernet Adapter 95
2-32 VLAN configuration example. 98
2-33 Adding virtual Ethernet adapters on the Virtual I/O Server for VLANs . 100
2-34 MPIO Single Client with Single VIOS with dual paths to the same disk 106
2-35 MPIO Single Client with Dual VIOS accessing the same disk 107

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

2-36 Implementing Mirroring in the VIO client or VIO server 108
2-37 License boundaries with different processor and pool modes 114
2-38 Licensing requirements for a non-partitioned server 117
2-39 Licensing requirements in a micro-partitioned server 117
2-40 Physical, virtual, and logical processors 120
2-41 SMIT SMT panel with options 122
2-42 Virtual I/O Server SEA comparison with Integrated Virtual Ethernet . . 131
3-1 The Config Assist for VIOS menu 134
3-2 Hardware Management Console Server view 154
3-3 Starting the Create Logical Partition Wizard 155
3-4 Using the pull-down menu in the HMC Contents area 156
3-5 Defining the partition ID and partition name 157
3-6 Naming the partition profile 158
3-7 Select whether processors are to be shared or dedicated 159
3-8 Processing Settings for a micro-partition 160
3-9 Memory setting 161
3-10 Selecting physical adapters for the partition 163
3-11 The start menu for creating virtual adpters. 164
3-12 Selecting to create a virtual Ethernet adapter 165
3-13 Creating Virtual Ethernet adapter 166
3-14 Creating the SCSI server adapter for the DVD 167
3-15 SCSI server adapter for the NIM_server 168
3-16 List of created virtual adapters 169
3-17 Menu for creating Logical Host Ethernet Adapters 170
3-18 Optional Settings menu 171
3-19 The Profile Summary menu 172
3-20 The created partition VIO_Server1 173
3-21 Activating a partition 175
3-22 The Activate Logical Partition submenu 176
3-23 Selecting the SMS menu for startup 176
3-24 The SMS startup menu 177
3-25 Basic Virtual I/O Server scenario 179
3-26 Setting TCP/IP parameters using the cfgassist command 182
3-27 Creating a client partition 198
3-28 The start menu for creating virtual adapters window 199
3-29 Creating a client Ethernet adapter 200
3-30 Creating the client SCSI disk adapter 201
3-31 Creating the client SCSI DVD adapter 201
3-32 List of created virtual adapters 202
3-33 The Logical Host Ethernet Adapters menu 203
3-34 The Optional Settings menu 204
3-35 The Profile Summary menu 205
3-36 The list of partitions for the basic setup 206

3-37 Backing up the profile definitions 207
3-38 Activating the DB_server partition 209
3-39 The SMS menu 210
3-40 Selecting adapter 211
3-41 IP settings 212
3-42 Ping test 213
3-43 Setting the install device 214
3-44 The edit Managed Profile window 216
3-45 Setting the Processor Sharing options 217
3-46 Basic configuration flow of virtual SCSI resources 219
3-47 Basic configuration flow of virtual SCSI resources 220
3-48 Edit Virtual Slots in SPT 223
3-49 Selecting to work with System Plans 224
3-50 Importing a system plan 225
3-51 Deploying a system plan 226
3-52 Opening the Deploy System Plan Wizard 226
3-53 System plan validation 227
3-54 The Partition Deployment window 228
3-55 Operating Environment installation window 229
3-56 The Customize Operating Environment menu 230
3-57 The New Install Image Resource menu 231
3-58 Virtual I/O Server installation setup 231
3-59 The Summary window 234
3-60 Deployment complete 235
3-61 Basic scenario deployed from the system plan created in SPT 236
3-62 Creating an HMC system plan for documentation 237
3-63 Giving a name to the system plan being created 238
3-64 The created system plan 238
3-65 The back of the server and its installed adapters 239
3-66 Options for the HMC system plan 240
3-67 Added lpar using the system plan 242
4-1 Virtual SCSI redundancy using MPIO and LVM mirroring 246
4-2 MPIO attributes 249
4-3 Basic SEA failover configuration 253
4-4 Alternative configuration for SEA failover 255
4-5 Network redundancy using two Virtual I/O Servers and NIB 256
4-6 Link Aggregation (EtherChannel) on the Virtual I/O Server 260
4-7 Redundant Virtual I/O Servers before maintenance 261
4-8 Redundant Virtual I/O Servers during maintenance 262
4-9 Separating disk and network traffic 264
4-10 Highly available Shared Ethernet Adapter setup 265
4-11 Create an IP address on the Shared Ethernet Adapter using cfgassist 268
4-12 NIB configuration 271

4-13 SAN attachment with MPIO with multiple Virtual I/O Server 276
4-14 LVM mirroring scenario 289
4-15 VIO_Server2 physical adapter selection 290
4-16 Virtual SCSI adapters for VIO_Server2 291
4-17 Linux client partition using mirroring with mdadm 303
4-18 Linux partitioning layout for mdadm mirroring (Red Hat) 304
4-19 Linux client partition using MPIO to access SAN storage 306
4-20 Starting Shared-Processor Pool configuration 311
4-21 Virtual Shared-Processor Pool selection 312
4-22 Shared-Processor Pool configuration 313
4-23 Virtual Shared-Processor Pool partition tab 314
4-24 Shared-Processor Pool partition assignment 314
4-25 Overview of Shared-Processor Pool assignments 315
4-26 Flow chart of virtual Ethernet 319
4-27 VLAN configuration scenario 321
4-28 Virtual Ethernet configuration for the client partition using the HMC . . . 323
4-29 Virtual Ethernet configuration for Virtual I/O Server using the HMC . . . 324
4-30 HMC in a VLAN tagged environment 328
4-31 Cross-network VLAN tagging with a single HMC 329
4-32 Supported and recommended ways to mirror virtual disks 332
4-33 RAID5 configuration using a RAID adapter on the Virtual I/O Server . . 333
4-34 Recommended way to mirror virtual disks with two Virtual I/O Server . 334
4-35 Using MPIO on the Virtual I/O Server with IBM TotalStorage 337
4-36 Using RDAC on the Virtual I/O Server with IBM TotalStorage 338
4-37 Configuration for multiple Virtual I/O Server and IBM ESS 339
4-38 Configuration for multiple Virtual I/O Servers and IBM FAStT 340
4-39 Basic issues for storage of AIX client partitions and HACMP 342
4-40 Example of HACMP cluster between two AIX client partitions 344
4-41 Example of an AIX client partition with HACMP using two VIOSs 345

Tables

1-1 The PowerVM umbrella . . . 3
1-2 PowerVM Editions . . . 4
1-3 CUoD options for POWER5-based systems . . . 14
1-4 CUoD options for POWER6-based systems . . . 15
2-1 Overview of PowerVM capabilities by edition . . . 20
2-2 PowerVM feature code overview . . . 26
2-3 Entitled capacities for micro-partitions in a Shared-Processor Pool . . . 47
2-4 Attribute values for the default Shared-Processor Pool (SPP0) . . . 50
2-5 Suggested maximum number of devices per virtual SCSI link . . . 79
2-6 Inter-partition VLAN communication . . . 99
2-7 VLAN communication to external network . . . 101
2-8 Kernel modules for IBM System p5 virtual devices . . . 104
2-9 Reasonable settings for shared processor partitions . . . 129
3-1 Network settings . . . 181
4-1 Main differences between EC and LA aggregation . . . 259
4-2 Virtual Ethernet adapter overview for Virtual I/O Servers . . . 266
4-3 Network interface backup configuration examples . . . 271
4-4 Virtual SCSI adapter configuration for MPIO . . . 278
4-5 Virtual SCSI adapter configuration for LVM mirroring . . . 290
4-6 Micro-partition configuration and Shared-Processor Pool assignments . 309
4-7 Shared-Processor Pool attributes . . . 310

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

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

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

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

Trademarks

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

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

AIX 5L™
AIX®
BladeCenter®
DB2®
DS4000™
DS6000™
DS8000™
Enterprise Storage Server®
GDPS®
General Parallel File System™
Geographically Dispersed Parallel Sysplex™
GPFS™
HACMP™
i5/OS®
IBM®
Micro-Partitioning™
Parallel Sysplex®
Passport Advantage®
POWER™
Power Architecture®
POWER Hypervisor™
POWER4™
POWER5™
POWER6™
PowerVM™
pSeries®
PTX®
Redbooks®
Redbooks (logo) ®
System i™
System p™
System p5™
System Storage™
System z™
Tivoli®
TotalStorage®
Virtualization Engine™
Workload Partitions Manager™

The following terms are trademarks of other companies:

NetApp, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

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

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

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

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

Preface

This IBM® Redbooks® publication provides an introduction to PowerVM™ virtualization technologies on IBM System p™ servers. The Advanced POWER™ Virtualization features and partitioning and virtualization capabilities of IBM Systems based on the Power Architecture® have been renamed to PowerVM.

PowerVM is a combination of hardware, firmware and software that provides CPU, network and disk virtualization. The main virtualization technologies are:

- POWER6™ and POWER5™ hardware
- POWER Hypervisor™
- Virtual I/O Server

Though the PowerVM brand includes partitioning, software Linux® emulation, management software, and other offerings, this publication focuses on the virtualization technologies that are part of the PowerVM Standard and Enterprise Editions.

This publication is also designed to be an introduction guide for system administrators, providing instructions for:

- Configuration and creation of partitions and resources on the HMC
- Installation and configuration of the Virtual I/O Server
- Creation and installation of virtualized partitions

While discussion is focussed on IBM System p hardware and AIX®, the basic concepts can be extended to the i5/OS® and Linux operating systems as well as the IBM System i™ hardware.

This edition has been updated with the new features available with the IBM POWER6 hardware and firmware.

The team that wrote this book

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

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

Christopher Hales is a Systems Architect and Consulting IT Specialist based in the UK and currently works in the IBM STG Major Deal Team. Chris has been designing and implementing IT solutions on behalf of customers for over 22 years. He specializes in virtualization technologies and has assisted several large IBM customers to deploy virtualized infrastructures across a number of platforms. Chris holds an Honors degree in Computer Science, and is a co-inventor of the Multiple Shared-Processor Pools technology.

Chris Milsted is a Senior IT Specialist based in the UK working on the System p services team. He has three years of experience in Advanced POWER Virtualization and PowerVM. Chris has worked for IBM UK for seven years. His areas of expertise include System p, AIX, Linux and High Availability. Chris holds a Masters in Chemistry from the University of Nottingham.

Oliver Stadler is a Senior IT Specialist working in the Integrated Technology Delivery division in IBM Switzerland. He has 18 years of experience in the IT industry. In his current job he is responsible for the design and implementation of System p and AIX based solutions. He has written extensively on System p virtualization.

Morten Vågmo is a Consulting IT Specialist in IBM Norway with 19 years of AIX and System p experience. He is working in technical pre-sales support and is now focusing on Advanced POWER Virtualization and PowerVM implementations. Morton co-authored the previous update of the *Advanced POWER Virtualization on IBM System p5, Introduction and Configuration* book. Morten holds a degree in Marine Engineering from the Technical University of Norway.

The project that produced this publication was managed by:

Scott Vetter, PMP. Scott is a Certified Executive Project Manager at the International Technical Support Organization, Austin Center. He has enjoyed 23 years of rich and diverse experience working for IBM in a variety of challenging roles. His latest efforts are directed at providing world-class Power Systems Redbooks, white papers, and workshop collateral.

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

John Banchy, Gail Belli, Herman Dierks, Ron Gordon, Eric Haase, Bob Kovacs, Chris Liebl, Naresh Nayar, Jorge Nogueras, Jim Pafumi, Amartey Pearson, Michael Reed, Sergio Reyes,
IBM US

Nigel Griffiths, Sam Moseley, Dai Williams
IBM UK

Thanks to the authors of the previous editions of this book.

- Authors of the first edition, *Advanced POWER Virtualization on IBM eServer p5 Servers: Introduction and Basic Configuration*, published in October 2004, were:

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

- Authors of the second edition, *Advanced POWER Virtualization on IBM System p5*, December 2005, were:

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

- Authors of the third edition, *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration*, February 2007, were:

  Morten Vågmo, Peter Wüstefeld

Become a published author

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

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

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

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

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

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

  `ibm.com/redbooks`

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Summary of changes

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

May 2008, Fourth Edition

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

New information

- Added the new Enterprise edition of PowerVM, see 1.2.2, “PowerVM Editions” on page 4
- Added new POWER6 capabilities including
  - Live Partition Mobility, see 1.2.11, “Live Partition Mobility” on page 11
  - Multiple Shared Processor Pools, see “Multiple Shared-Processor Pools” on page 7
  - Shared Dedicated Capacity, see “Shared dedicated capacity” on page 8

Changed information

- Updated many areas to change from Advanced POWER Virtualization to PowerVM
- Large sections rewritten to reflect the new technologies above.
- Sections five onwards have been moved into the *PowerVM virtualization on IBM System p Managing and Monitoring*, SG24-7590 publication..
- Best practises from the *Advanced POWER Virtualization Best Practises*, REDP-4194 Redpaper, have been incorporated into this book where appropriate.
- Updates covering the Virtual I/O Server Version 1.5
- Screenshots updated for HMC version 7 and AIX 6.1
- Minor extension of the MPIO topic indicated by change bars in the left margin.
- Notes added regarding VIOS installation target disk size minimum of 30 GB.
- Added a recomendation to handle the case when the optical install media is sharing a disk resource controller, and two VIOS partitions are needed on internal storage.

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

1

Introduction

The first edition of this IBM Redbooks publication was published over three years ago. Since then the number of customers using Advanced POWER Virtualization (currently named PowerVM) editions on IBM System p servers has grown rapidly. Customers use PowerVM in a variety of environments including business-critical production systems, development, and business continuity. This fourth edition includes best practices learned over the past years to build on the foundation work of the previous versions of this book.

This book targets customers new to virtualization as well as more experienced virtualization professionals. It is split into four chapters, each with a different target audience in mind.

Chapter 1 provides a short overview of the key virtualization technologies. An understanding of this chapter is required for the remainder of the book.

Chapter 2 is a slightly more in-depth discussion of the technology aimed more at the estate- or project-architect for deployments.

Chapters 3 and 4 are aimed at professionals who are deploying the technology. Chapter 3 works through a simple scenario and Chapter 4 introduces the more advanced topics such as VLANs, Multiple Shared Processor Pools, and Linux. Additionally we introduce the techniques that can be used to provide the periods of continuous availability required in production systems.

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

1.1 The value of virtualization on System p

One of the main objectives of any IT department is to be responsive to business demand for new or changing system deployments. Demand for:

- New applications or systems
- Changes in the capacity or service standards for existing systems

needs to be met ever more rapidly and as cost-effectively as possible.

Classic IT deployments (non-virtualized) involved the specification, sizing, and configuration of many individual elements:

- Servers
- Networks and switches
- SAN
- Storage

This approach to the selection and purchase of the new hardware and the time taken to deploy was often lengthy and costly. Using IBM System p virtualization it is possible to provide rapid, cost-effective response to business need. In addition, PowerVM allows you to increase the utilization of servers, which reduces;

- Server management and administration costs, because there are fewer physical servers
- Power and cooling costs with increased utilization of existing servers
- Time to market, virtual resources can be deployed immediately

With the escalating cost of powering and cooling servers, virtualization capabilities are becoming a very important element of server and estate design and planning. With commercial and environmental concerns growing, there is pressure to reduce the power foot-print within the server estate. IBM System p virtualization can help to meet growing demand for IT services by maximizing the utilization of server resources, helping to mitigate the impact of any future government regulations.

It is no exaggeration to say that many projects succeed because of IBM System p virtualization. The ability to provision a new micro-partition using available processor capacity on a server in a matter of minutes has allowed projects to react to changes in their architecture rapidly, without lengthy procurement cycles. Coupled with complementary technologies such as Capacity Upgrade on Demand and PowerVM AIX 6 Workload Partitions, the case for PowerVM on IBM System p is compelling.

1.2 PowerVM on IBM System p servers

The next sections provide an overview of the features that combine to form PowerVM.

1.2.1 What is PowerVM?

PowerVM is a brand that includes a range of hardware and software features that allow a system to be more useful in adapting various workloads.

PowerVM is the new brand for Power Systems virtualization. It includes elements such as Micro-Partitioning™, logical partitioning, Virtual I/O, hypervisor, and so on. Table 1-1 shows how the previous functions fit under the PowerVM brand.

Table 1-1 The PowerVM umbrella

Old name	New name	Enabler
Power Systems Virtualization	PowerVM	N/A - Umbrella name
Advanced POWER Virtualization (APV)	PowerVM Editions	POWER hardware, (POWER5/POWER6)
	PowerVM Express Edition	PowerVM Express Edition
APV Standard Edition	PowerVM Standard Edition	POWER5 or POWER6 hardware
APV Enterprise Edition	PowerVM Enterprise Edition	POWER6 hardware
System p Application Virtual Environment (p AVE)	PowerVM Lx86	POWER5 or POWER6
Live Partition Mobility	PowerVM Live Partition Mobility	POWER6 hardware
AIX 6 Workload Partitions Manager™	PowerVM AIX 6 Workload Partitions Manager	AIX 6.1
AIX 6 Workload Partitions	PowerVM AIX 6 Workload Partitions	AIX 6.1

1.2.2 PowerVM Editions

The PowerVM Editions are optional hardware features available on IBM System p servers based on POWER5 or POWER6 processors.

The PowerVM Express Edition is available on the POWER6 technology-based System p550 and System p520 Express servers and includes:

- Up to three partitions per server
- Shared dedicated capacity
- PowerVM Lx86, which enables x86-based Linux applications to run on these servers
- The Virtual I/O Server
- Integrated Virtualization Manager
- Power Hypervisor

PowerVM Standard Edition supports POWER5-based systems, allows a greater number of LPARs, and adds Multiple Shared Processor pools for POWER6-based systems.

PowerVM Enterprise Edition is only available on the new POWER6-based systems, and adds PowerVM Live Partition Mobility to the suite of functions.

Table 1-2 describes the features included in the Editions of PowerVM.

Table 1-2 PowerVM Editions

Feature	Licensed by	Processor support	Operating System support[a]
LPAR[b]	All Editions (limited on Express)	POWER4™, 5 and 6	AIX 5.3 and 6.1 SLES 9, 10 for POWER (not POWER4) RHEL 4, 5 for POWER (not POWER4)
Capacity Upgrade on Demand	All Editions (specific models)	POWER4, 5 and 6	AIX 5.3 and 6.1 SLES 9, 10 for POWER RHEL 4, 5 for POWER (5.1 for POWER6)
Micro-Partitioning	All Editions	POWER5 and 6	AIX 5.3 and 6.1 SLES 9, 10 for POWER RHEL 4, 5 for POWER
Shared Dedicated Capacity	All Editions	POWER6	AIX 5.3 and 6.1 SLES 10 update 1 for POWER RHEL 5.1 for POWER

Feature	Licensed by	Processor support	Operating System support[a]
Multiple Shared Processor Pools	PowerVM SE and EE	POWER6 with HMC	AIX 5.3 and 6.1 SLES 10 for POWER6 RHEL 5.1 for POWER6
Virtual I/O Server	All Editions	POWER5 and 6	Virtual I/O Server
IVM	PowerVM Express, SE and EE	POWER5 and 6	Integrated Virtualization Manager
Virtual SCSI	All Editions	POWER5 and 6	AIX 5.3 and 6.1 SLES 10 for POWER RHEL 4, 5, 5.1 for POWER
Virtual Ethernet	All Editions	POWER5 and 6	AIX 5.3 and 6.1 SLES 9 for POWER (10 for POWER6) RHEL 4, 5 for POWER (5.1 for POWER6)
Live Partition Mobility	PowerVM Enterprise Edition	POWER6	AIX 5.3 and 6.1 SLES 10 update 1 for POWER RHEL 5.1 for POWER

a. Contact your sales representative for specific technology level or service pack requirements.
b. Including DLPAR operations, except dynamic memory removal for all Linux versions.

1.2.3 POWER Hypervisor

The POWER Hypervisor is the foundation for virtualization on IBM System p servers, enabling the hardware to be divided into multiple partitions, and ensuring isolation between them.

Always active on POWER5- and POWER6-based servers, the POWER Hypervisor is responsible for dispatching the logical partition workload across the shared physical processors. The POWER Hypervisor also enforces partition security, and can provide inter-partition communication that enables the Virtual I/O Server's virtual SCSI and virtual Ethernet function.

1.2.4 Simultaneous multithreading

Simultaneous multithreading (SMT) is an IBM microprocessor technology that allows two separate hardware instruction streams (threads) to run concurrently on the same physical processor. SMT significantly improves overall processor and system throughput. SMT was first introduced on POWER5-based servers and has been further enhanced in POWER6-based servers with corresponding gains in efficiency.

1.2.5 Logical partitions and processor capacity

A logical partition can be regarded as a logical server, and is capable of booting an operating system and running a workload.

Logical partitions

LPAR technology has been available in IBM POWER-based systems since 2001 with the introduction of the IBM pSeries® 690, a POWER4-based server. A set of system resources: whole POWER microprocessors, memory, and I/O resources are logically grouped into a partition—an LPAR. Because resource allocation is a logical function, the amount of resources can vary according to need and availability within the physical server.

The physical shared-processor pool and micro-partitioning

Shared-processor partition capability was introduced with IBM POWER5-based servers.

These shared-processor partitions, known as micro-partitions, are allocated processor resources from a single pool of physical processors.

The amount of processor capacity that is allocated to a micro-partition—its entitled capacity—may range from ten percent (10%) of a physical processor up to the entire capacity of the physical shared processor pool. The physical shared processor pool can have from one physical processor up to the total installed processor capacity of the system.

Changes to the entitled capacity of a micro-partition can be as granular as one percent (1%) of a physical processor.

Micro-partitions can be designated as either *capped* or *uncapped*. These modes are defined as follows:

- *Capped*

  A capped micro-partition has a defined processor entitled capacity which it is guaranteed to receive. There are no circumstances under which a capped micro-partition will receive any more processor resource than its entitled capacity.

- *Uncapped*

  An uncapped micro-partition has a defined processor entitled capacity which it is guaranteed to receive. However, under some circumstances it can receive additional processor resources. Typically, if the uncapped micro-partition is runnable (has real work to execute) and there are unused processor cycles within the physical shared processor pool, then additional cycles can be allocated to the micro-partition on a weighted basis.

Capped and uncapped micro-partitions can coexist and both receive the processor resources from the physical shared-processor pool.

Multiple Shared-Processor Pools

On POWER6-based systems there is support for Multiple Shared-Processor Pools. This new capability allows a systems administrator to create a set of micro-partitions with the purpose of controlling the processor capacity that they can consume from the physical shared-processor pool.

Multiple Shared-Processor Pools use processor capacity from the physical shared processor pool—there can only be one physical shared processor pool in the system. Each Shared-Processor Pool has an associated Entitled Pool Capacity which is consumed by the set of micro-partitions in that Shared-Processor Pool.

The micro-partitions within a Shared-Processor Pool are guaranteed to receive their entitled capacity. In addition, unused processor cycles within their Shared-Processor Pool are harvested and are redistributed to eligible micro-partitions within the same Shared-Processor Pool.

The source of additional processor cycles can be the Reserved Pool Capacity, which is processor capacity specifically reserved for a Shared-Processor Pool but not assigned to any of the micro-partitions in the Shared-Processor Pool.

When the set of micro-partitions in a Shared-Processor Pool are heavily loaded, they can consume additional processor capacity (assuming they are uncapped) from outside their Shared-Processor Pool up to a defined maximum, Maximum

Pool Capacity. Processor capacity distributed in this way has been ceded by underutilized sets of micro-partitions in their Shared-Processor Pools.

There is a default Shared-Processor Pool (Shared-Processor $Pool_0$) that is used for all micro-partitions if the system administrator has not created other Shared-Processor Pools. The Shared-Processor $Pool_0$ has some special properties that ensure operational consistency with POWER5-based micro-partitioning and the physical shared processor pool.

Figure 1-1 shows an architectural overview of a POWER6-based server with two Shared-Processor Pools defined.

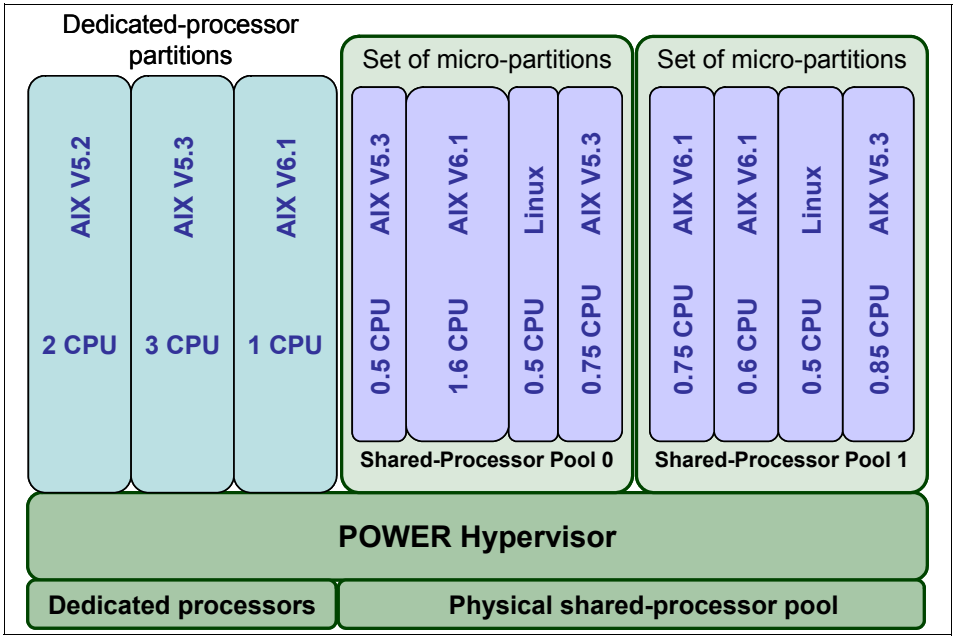


Figure 1-1 Architectural overview of POWER6-based server with MSPPs defined

There can be up to 64 MSPPs in total. With the default MSPP, there can be an additional 63 MSPPs that are user-defined.

Shared dedicated capacity

POWER6-based servers offer the capability of harvesting unused processor cycles from dedicated-processor partitions. These unused cycles are then donated to the physical shared-processor pool associated with micro-partitioning. This ensures the opportunity for maximum processor utilization throughout the system.

The systems administrator can control which LPARs can donate unused cycles. The LPAR must be identified in the LPAR profile.

1.2.6 LPAR and dynamic reconfiguration

The POWER Hypervisor allows you to define logical partitions, which can be a subset of the overall system resources.

It is possible to dynamically move system resources between partitions without rebooting. These resources can be:

- Physical processors (for dedicated processor LPARs)
- Virtual processors or entitled capacity (for micro-partitions)
- Memory
- Physical I/O adapters
- Virtual I/O adapters

This is known as dynamic reconfiguration or dynamic LPAR.

1.2.7 Virtual Ethernet

The virtual Ethernet function is provided by the POWER Hypervisor. The POWER Hypervisor implements the Ethernet transport mechanism as well as an Ethernet switch that supports VLAN capability. Virtual LAN allows secure communication between logical partitions without the need for a physical I/O adapter or cabling. The ability to securely share Ethernet bandwidth across multiple partitions increases hardware utilization.

1.2.8 Virtual SCSI

The functionality for virtual SCSI is provided by the POWER Hypervisor. Virtual SCSI allows secure communications between partitions and a Virtual I/O Server that provides storage backing devices.

The combination of virtual SCSI and the Virtual I/O Server capabilities allows you to share storage adapter bandwidth and (optionally) to subdivide single large disks into smaller segments. The adapters and disks can then be shared across multiple partitions, increasing utilization.

The virtual disk I/O capability offered by the combination of Virtual SCSI and the Virtual I/O Server provides the opportunity to share physical disk I/O adapters in a flexible and reliable manner. A single physical disk I/O adapter and associated

disk subsystem can be used by many logical partitions on the same server. This facilitates the consolidation of disk I/O resources and minimizes the number of disk I/O adapters required.

1.2.9 Virtual I/O Server

As part of PowerVM there is an appliance server with which you can associate physical resources and that allows you to share these resources amongst a group of logical partitions. The Virtual I/O Server can use both virtualized storage and network adapters, making use of the virtual SCSI and virtual Ethernet facilities.

For storage virtualization, these backing devices can be used:

- Direct-attached entire disks from the Virtual I/O Server
- SAN disks attached to the Virtual I/O Server
- Logical volumes defined on either of the previous disks
- File-backed storage, with the files residing on either of the first two disks
- Optical storage devices

For virtual Ethernet we can define Shared Ethernet Adapters on the Virtual I/O Server, bridging network traffic from the virtual Ethernet networks out to physical Ethernet networks.

The Virtual I/O Server technology facilitates the consolidation of LAN and disk I/O resources and minimizes the number of physical adapters that are required, while meeting the non-functional requirements of the server. To understand the support for storage devices, refer to the following Web site:

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

The Virtual I/O Server can run in either a dedicated processor partition or a micro-partition. The different configurations for the Virtual I/O Server and associated I/O subsystems can be seen in *Advanced POWER Virtualization on IBM System p: Virtual I/O Server Deployment Examples*, REDP-4224.

1.2.10 Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) is a Web-browser based management interface that provides support for certain models of POWER5- or POWER6-based systems. The IVM is hosted by the Virtual I/O Server.

IVM systems do not require an HMC but each system requires a separate instance of IVM. For this reason IVM is recommended for smaller systems in stand-alone deployments.

1.2.11 Live Partition Mobility

Live Partition Mobility, licensed through PowerVM Enterprise Edition, is a feature that relies on a number of different components, including:

- POWER Hypervisor
- Virtual I/O Server (or IVM)
- Hardware Management Console (or IVM)

Live Partition Mobility allows you to move running AIX or Linux partitions from one physical POWER6 server to another without disruption. The movement of the partition includes everything that partition is running, that is, all hosted applications. Some possible applications and advantages are:

- Moving partitions from servers to allow planned maintenance of the server without disruption to the service and users
- Moving heavily used partitions to larger machines without interruption to the service or disruption to users
- Moving partitions to appropriate servers depending on workload demands; adjusting the utilization of the server-estate to maintain an optimal level of service to users at the optimal cost
- Consolidation of underutilized partitions out-of-hours to enable unused servers to be shut down, saving power and cooling expenditure

For more information on Live Partition Mobility see, *IBM System p Live Partition Mobility*, SG24-7940.

1.2.12 Multiple operating system support

The POWER5-based System p5™ servers support IBM AIX 5L Version 5.2, IBM AIX 5L™ Version 5.3, IBM AIX Version 6.1, i5/OS, and Linux distributions from SUSE and Red Hat.

POWER6-based System p servers support IBM AIX 5L Version 5.2 (selected systems only), IBM AIX 5L Version 5.3, IBM AIX Version 6.1, and Linux distributions from SUSE and Red Hat.

Important: For specific technical support details, refer to the IBM System p Web site:

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

IBM AIX 5L

The AIX operating system is an open standards based UNIX® operating system that provides the enterprise-class IT infrastructure for thousands of clients around the world.

On System p servers, mixed environments are supported: AIX 5L V5.2 ML2 and AIX 5L V5.3 partitions with dedicated processors and adapters, and of AIX 5L V5.3 partitions using micro-partitioning and virtual devices. AIX 5L V5.3 partitions can use physical and virtual resources at the same time.

AIX 5L is supported on the System p servers in partitions with dedicated processors (LPARs), and shared-processor partitions (micro-partitions). See Table 1-2 on page 4 for more information.

For System p servers configured with one of the PowerVM features, AIX 5L Version 5.3 or later is required for micro-partitions, virtual I/O, and virtual LAN.

IBM AIX V6.1

IBM is making available a new version of AIX, AIX V6.1, which will include significant new capabilities for virtualization, security features, continuous availability features and manageability. AIX V6.1 is the first generally available version of AIX V6.

AIX V6.1 features include support for:

- PowerVM AIX 6 Workload Partitions (WPAR), software-based virtualization
- Live Application Mobility - with the IBM PowerVM AIX 6 Workload Partitions Manager for AIX (5765-WPM)
- 64-bit kernel for higher scalability and performance
- Dynamic logical partitioning and micro-partitioning support
- Support for Multiple Shared-Processor Pools
- Trusted AIX - Multilevel, compartmentalized security
- Integrated Role Based Access Control
- Encrypting JFS2 file system
- Kernel exploitation of POWER6 Storage Keys for greater reliability

- Robust journaled file system and Logical Volume Manager (LVM) software including integrated file system snapshot
- Tools for managing the systems environment—System Management
- Interface Tool (SMIT) and the IBM Systems Director Console for AIX

Linux for System p systems

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

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

- Novell SUSE Linux Enterprise Server V9 and V10
- Red Hat Enterprise Linux Advanced Server V3, V4, V4.6 and V5

The PowerVM features are supported in Version 2.6.9 and above of the Linux kernel. The commercially available latest distributions from Red Hat, Inc. (RHEL AS 5) and Novell SUSE LINUX (SLES 9 and SLES 10) support the IBM System p 64-bit architectures and are based on this 2.6 kernel series. Also supported is the modified 2.4 kernel version of Red Hat Enterprise Server AS 3 (RHEL AS 3) with update 8.

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

- Not all devices and features supported by the AIX operating system are supported in logical partitions running the Linux operating system.
- Linux operating system licenses are ordered separately from the hardware. Clients can acquire Linux operating system licenses from IBM, to be included with their System p5 systems, or from other Linux distributors.
- Clients or authorized business partners are responsible for the installation and support of the Linux operating system on the System p5 systems.
- Regardless of how a Linux distribution is ordered, the distributors offer maintenance and support. IBM also has support offerings from IBM Global Services for these distributions.
- While Linux can be run successfully on partitions with more than eight processors (the Linux kernel 2.6 may scale up to 16 or even 24 processors for certain workloads) typical Linux workloads will only effectively utilize up to four or eight processors.

Supported virtualization features

SLES 9, SLES 10, RHEL AS 4 RHEL AS 4.5, and RHEL AS 5 support the following virtualization features:

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

SLES 9, SLES 10, RHEL AS 4, RHEL AS 4.5, and RHEL AS 5 do not support the following:

- Dynamic reconfiguration of memory
- Dynamic reconfiguration of I/O slots

1.3 Capacity Upgrade on Demand for IBM System p

There are several Capacity Upgrade on Demand (CUoD) possibilities offered on selected System p servers. Table 1-3 shows the CUoD options for POWER5-based System p servers.

Table 1-3 CUoD options for POWER5-based systems

CUoD For POWER5	Description
Permanent Capacity Upgrade on Demand	Enables permanent system upgrades by activating processors or memory.
On/Off Capacity Upgrade on Demand	Usage-based billing that allows for activation and deactivation of both processors and memory as required.
Reserve Capacity Upgrade on Demand	Prepaid agreement that adds reserve processor capacity to the shared processor pool, which is used if the base shared pool capacity is exceeded.
Trial Capacity Upgrade on Demand	Partial or total activation of installed processors or memory for a fixed period of time.

For POWER6-based servers, the CUoD options can be seen in Table 1-4.

Table 1-4 CUoD options for POWER6-based systems

CUoD For POWER6	Description
Permanent Capacity Upgrade on Demand	Enables permanent system upgrades by activating processors or memory
On/Off Capacity Upgrade on Demand	Temporary use of processors or memory. Capacity can be activated or deactivated as required, with charging based on usage.
Utility Capacity on Demand	Processor capacity, measured in processor-minutes, can be activated or deactivated. A prepaid or post-paid agreement is required.
Trial Capacity Upgrade on Demand	Partial or total activation of installed processors or memory for a fixed period of time.

1.4 Availability of virtualized systems

Because individual System p servers are capable of hosting many system images, the importance of isolating and handling service interruptions becomes greater. These service interruptions can be planned or unplanned. Interruptions for systems maintenance should be carefully considered. When planning system maintenance windows a number of factors should be considered, including, but not limited to:

- Environmentals, including cooling and power
- System firmware
- Operating systems, for example AIX, Virtual I/O Server, and Linux
- Adapter microcode

Technologies such as Live Partition Mobility or clustering (for example HACMP™ for AIX on IBM System p) can be used to move workloads between machines, allowing for scheduled maintenance, minimizing any service interruptions.

For applications requiring near-continuous availability, clustering technology such as HACMP should also be used to provide protection across physical machines. These machines should be located such that they are not reliant on any one single support infrastructure element (for example, the same power and cooling

facilities). In addition, environmental factors such as earthquake zones or flood plains should be considered.

The IBM System p servers, based on the POWER processor, build upon a strong heritage of systems designed for industry-leading availability and reliability.

IBM takes a holistic approach to systems reliability and availability—from the microprocessor, which has dedicated circuitry and components designed into the chip, to Live Partition Mobility and the ability to move running partitions from one physical server to another. The extensive component, system, and software capabilities which focus on reliability and availability, coupled with good systems management practice, can deliver near-continuous availability.

Reducing and avoiding outages

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

For more information about the continuous availability of System p systems, refer to the whitepapers:

IBM System p5: a Highly Available Design for Business-Critical Applications, available at:

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

or I*BM POWER6 Processor-based Systems: Designed for Availability* at:

http://www-03.ibm.com/systems/p/hardware/whitepapers/power6_availability.html

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

https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

1.4.1 Serviceability in virtualized environments

In partitioned environments where business-critical applications are consolidated onto the same hardware, exceptional availability and serviceability are needed. This ensures a smooth recovery from unplanned service interruptions. The POWER Hypervisor ensures that issues affecting one partition do not propagate into other logical partitions on the sever.

For systems requiring the highest levels of availability, functions at the operating system and application levels should be investigated to allow for quick recovery of service for the end users. An example of this might be the IBM clustering technology for AIX on System p, HACMP.

Redundant Virtual I/O Servers

Since an AIX or Linux partition can be a client of one or more Virtual I/O Servers at the same time, a good strategy to improve availability for sets of AIX or Linux client partitions is to connect them to two Virtual I/O Servers. One key reason for redundancy is the ability to upgrade to the latest Virtual I/O Server technologies without affecting production workloads. Techniques discussed later in the book provide redundant configurations for each of the connections from the client partitions to the external Ethernet network or storage resources.

Figure 1-2 shows some of the mechanisms used to improve the availability and serviceability of a partitioned System p server.

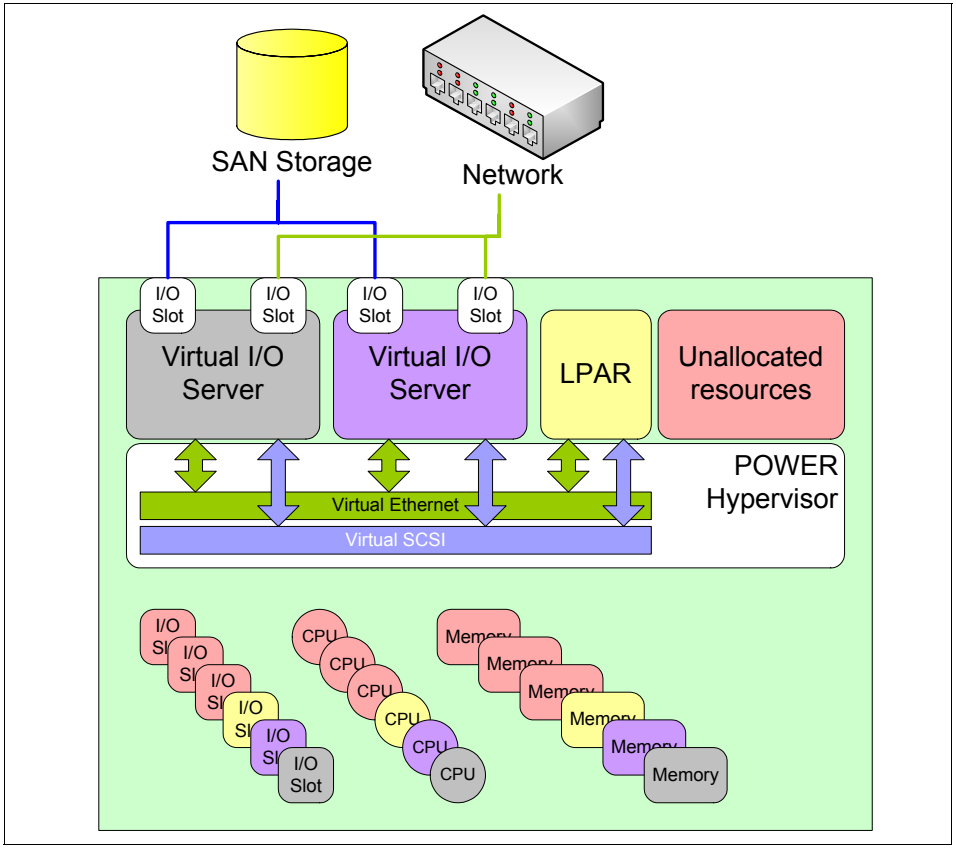


Figure 1-2 Redundant components in a virtualized system

1.5 Security in a virtualized environment

In POWER5 and POWER6 machines all resources are controlled by the POWER Hypervisor. The POWER Hypervisor ensures that any partition attempting to access resources within the system has permission to do so.

PowerVM has introduced a number of technologies allowing partitions to securely communicate within a physical system. To maintain the complete isolation of partition resources the POWER Hypervisor enforces communications standards, as normally applied to external infrastructure communications. For example, the virtual Ethernet implementation is based on the IEEE 802.1Q standard.

The IBM System p partition architecture, AIX and the Virtual I/O Server have been security certified to the EAL4+ level. For more information, see:

http://www-03.ibm.com/systems/p/os/aix/certifications/index.html

1.6 Summary of PowerVM

PowerVM on System p offers industry leading virtualization capabilities for AIX and Linux. With the Standard Edition of PowerVM, micro-partitioning allows businesses to increase the utilization of their servers, with server definitions down to 1/10th of a processor and the ability to allow server size to flex with demand. In addition, with PowerVM-SE, there is the Virtual I/O Server, which allows the sharing of expensive disk and network resources, while minimizing any management and maintenance costs.

With the introduction of PowerVM Enterprise edition, all of these features are joined by the ability to migrate running partitions and their applications from server to server. Combining these PowerVM features, we can help today's businesses further transform their computing departments into the agile, responsive and energy efficient organization demanded by today's enterprises.

2

Virtualization technologies on System p servers

In this chapter the various technologies that are part of the IBM System p servers are discussed, specifically the following:

- Editions of PowerVM
- The POWER Hypervisor
- Micro-partitioning
- Virtual I/O Server
- Integrated Virtualization Manager
- Virtual SCSI
- Virtual Ethernet
- Linux and PowerVM
- Software licensing and PowerVM
- Simultaneous Multi-threading
- Additional considerations in a PowerVM deployment

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

2.1 Editions of the PowerVM feature

This section provides information about the packaging and ordering information for the PowerVM Express Edition, Standard Edition, and Enterprise Edition, which are available on the IBM System p platform.

Table 2-1 outlines the functional elements of each edition of PowerVM.

Table 2-1 Overview of PowerVM capabilities by edition

PowerVM capability	PowerVM Express Edition	PowerVM Standard Edition	PowerVM Enterprise Edition
Servers Supported	POWER6 p520, POWER6 p550, JS22	System p System p5 JS21 blade JS22 blade	POWER6-based System p JS22 blade
Micro-partitions	Yes	Yes	Yes
Virtual I/O Server	One	Yes	Yes
Integrated Virtualization Manager	Yes	Yes	Yes
Shared Dedicated Capacity (POWER6 only)	Yes	Yes	Yes
Multiple Shared-Processor Pools (POWER6 only)		Yes	Yes
Live Partition Mobility			Yes

2.1.1 PowerVM Standard Edition feature

The PowerVM Standard Edition feature is a combination of hardware enablement and software that includes the following components that are available together as a single priced feature:

- Firmware enablement for:
  - Micro-Partitioning
  - Multiple Shared-Processor Pools
  - Shared Dedicated Capacity

- Installation image for the Virtual I/O Server software, which supports:
  - Shared Ethernet Adapter
  - Virtual SCSI server
  - Integrated Virtualization Manager (IVM) for supported systems

Virtual Ethernet and dedicated-processor LPAR are available without this feature for servers attached to an HMC or managed using the IVM.

When the hardware feature is specified with the initial system order, the firmware is shipped already activated to support Micro-Partitioning, Multiple Shared-Processor Pools, Shared Dedicated Capacity, and the Virtual I/O Server.

Note: Shared Dedicated Capacity and Multiple Shared-Processor Pools are supported on POWER6-based System p servers only.

For upgrade orders, IBM will ship a key to enable the firmware (similar to the CUoD key).

To find the current activation codes for a specific server, clients can visit the IBM Web site:

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

and enter the machine type and serial number. The activation code for PowerVM feature standard edition has a type definition of VET in the window results. You should see a screen similar to that shown in Figure 2-1 on page 22.

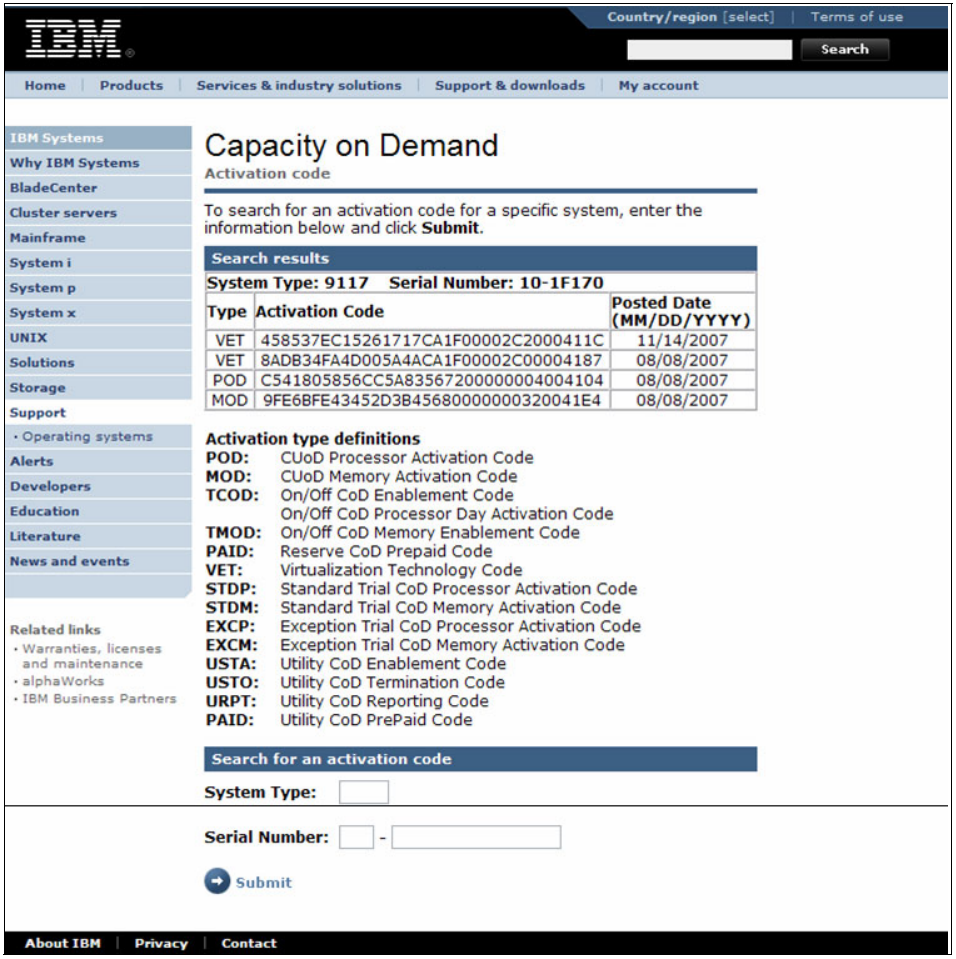


Figure 2-1 Example of virtualization activation codes Web site

For systems attached to an HMC, Figure 2-2 shows the HMC window where you activate PowerVM Standard Edition.

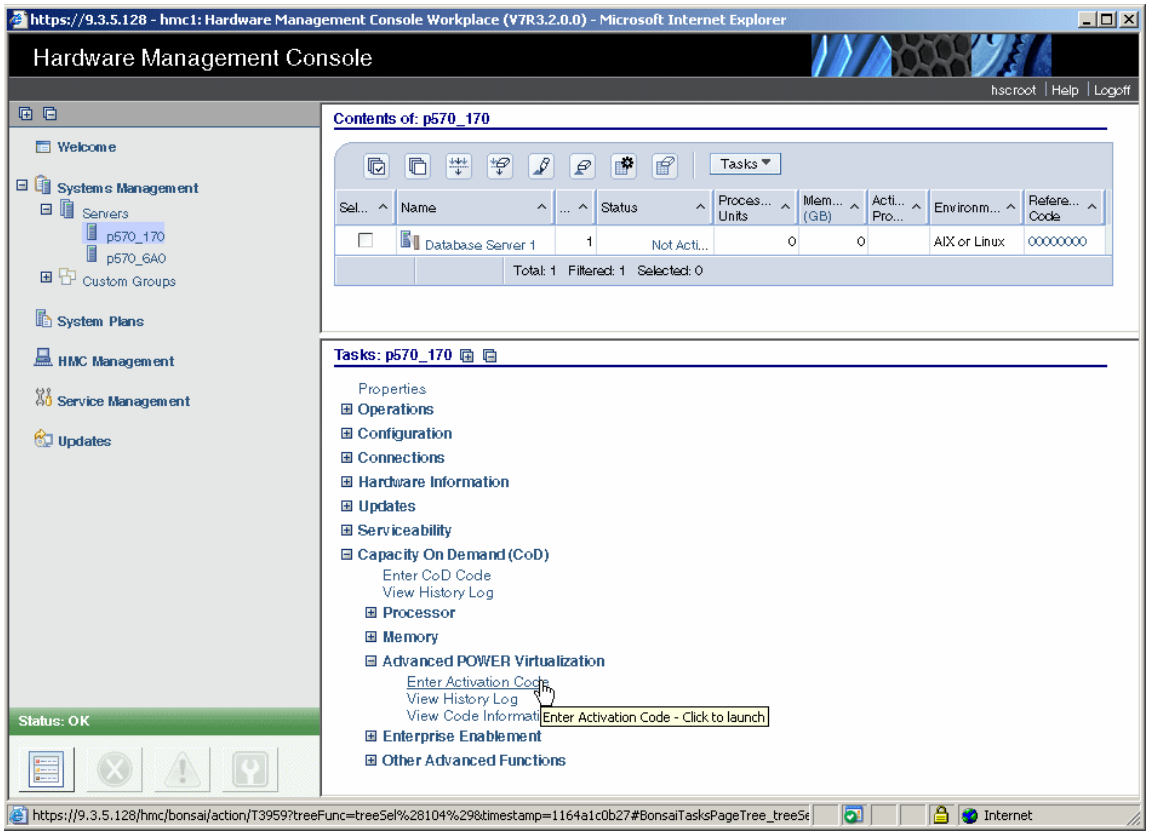


Figure 2-2 HMC window to activate PowerVM Standard Edition

The PowerVM Standard Edition feature is charged at one unit for each activated processor, including software maintenance.

Virtual I/O Server software is a licensed software component of the PowerVM Standard Edition feature. The Virtual I/O Server provides the virtual SCSI server and Shared Ethernet Adapter virtual I/O function to client partitions.

For each Virtual I/O Server license ordered, an order for either the one-year (5771-VIO) or three-year (5773-VIO) Software Maintenance (SWMA) is also submitted. You must purchase a license for each active processor on the server.

For an HMC-attached system, the processor-based license enables you to install several Virtual I/O Server partitions (usually two) on a single physical server to

provide redundancy and to spread the I/O workload across several Virtual I/O Server partitions.

Table 1-2 on page 4 shows the client operating system support for the Virtual I/O Server component of PowerVM Standard Edition.

The Virtual I/O Server provides the IVM management interface for systems without an HMC. Virtual I/O Server is an appliance-style partition that is not intended to run end-user applications, and should only provide login for system administrators.

When using the IVM within the Virtual I/O Server to manage a single system, Figure 2-3 shows the Advanced System Management Interface (ASMI) menu to enable the Virtualization Engine™ Technologies. For more information about this procedure, refer to *Integrated Virtualization Manager on IBM System p5*, REDP-4061.

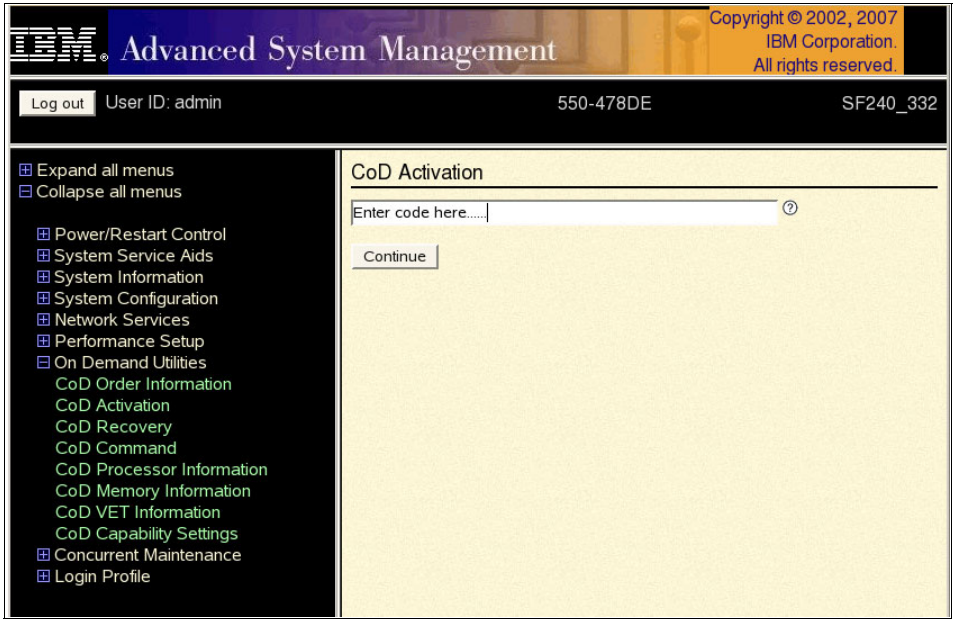


Figure 2-3 ASMI menu to enable the Virtualization Engine Technologies

2.1.2 PowerVM Enterprise Edition feature

The PowerVM Enterprise Edition feature code supersedes the feature code for the PowerVM Standard Edition. All PowerVM Standard Edition function is supplied with the PowerVM Enterprise Edition feature code.

The primary additional capability in this edition is PowerVM Live Partition Mobility. PowerVM Live Partition Mobility allows you to migrate running AIX and Linux partitions and their hosted applications from one physical server to another without disrupting the infrastructure services. The migration operation maintains complete system transactional integrity. The migration transfers the entire system environment, including processor state, memory, attached virtual devices, and connected users.

The benefits of PowerVM Live Partition Mobility include:

- Transparent maintenance - Allowing users and applications to continue operations by moving their running partitions to available alternative systems during the maintenance cycle.
- Meeting increasingly stringent service-level agreements (SLAs) because it allows you to proactively move running partitions and applications from one server to another.
- Balancing workloads and resources - Should a key application's resource requirements peak unexpectedly to a point where there is contention for server resources, you might move it to a larger server or move other, less critical, partitions to different servers, and use the freed-up resources to absorb the peak.
- A mechanism for dynamic server consolidation facilitating continuous server-estate optimization - Partitions with volatile resource requirements could use PowerVM Live Partition Mobility to consolidate partitions when appropriate or redistribute them to higher capacity servers at peak.

For more information about the Live Partition Mobility element of the PowerVM Enterprise Edition, refer to *IBM System p Live Partition Mobility*, SG24-7460.

For systems attached to an HMC, Figure 2-4 shows the HMC window where you enable the PowerVM Enterprise Edition feature.

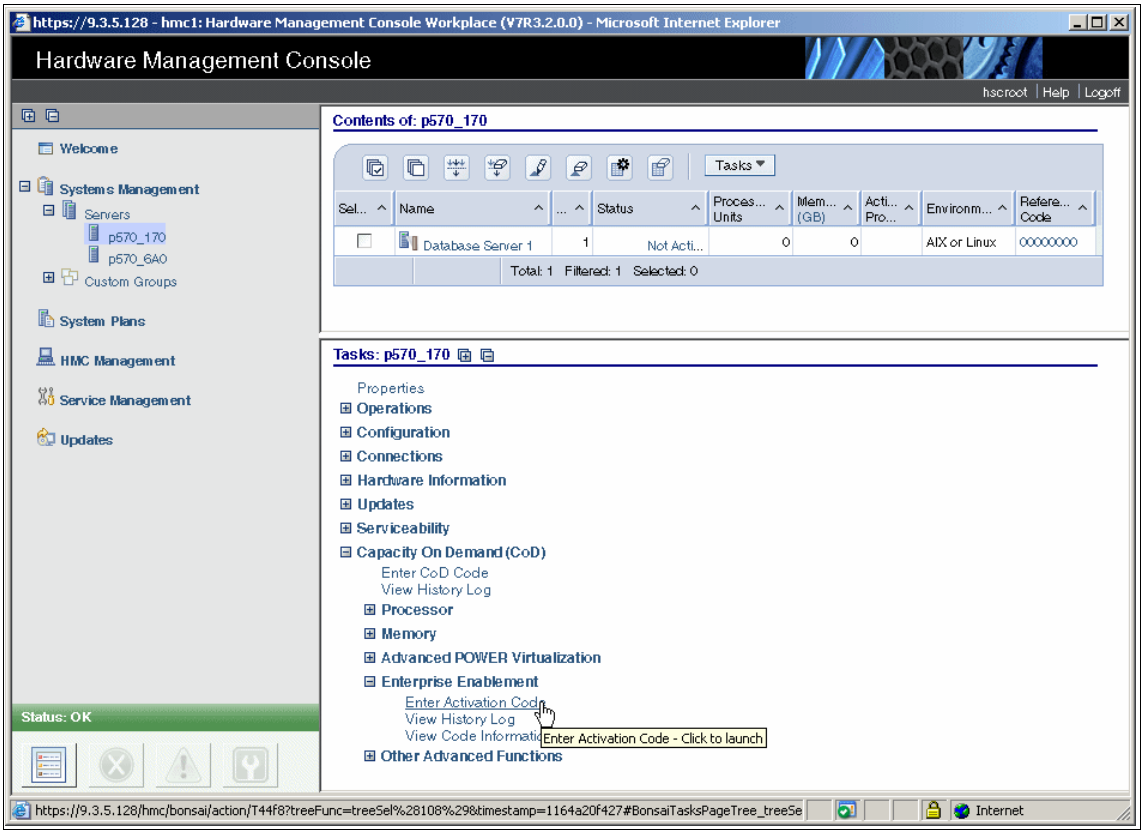


Figure 2-4 HMC window to enter the PowerVM Enterprise Edition activation code

2.1.3 Summary of PowerVM feature codes

Table 2-2 provides an overview of the PowerVM feature codes on the IBM System p systems.

Table 2-2 PowerVM feature code overview

Servers	Express Edition	Standard Edition	Enterprise Edition	Standard Edition included	IVM support
7998-61X		5409	5649[a]	Yes[b]	Yes
8203-E4A	7983	8506	8507	Express	Yes
8204-E8A	7983	7982	7986	Express	Yes
9110-51A		7432		No	Yes

Servers	Express Edition	Standard Edition	Enterprise Edition	Standard Edition included	IVM support
9115-505		7432		No	Yes
9131-52A		7940		No	Yes
9133-55A		7941		No	Yes
9116-561		7304		No	Yes
9117-570		7942		No	No
9117-MMA		7942	7995[a]	No	No
9118-575		7944		No	No
9119-590		7992		Yes	No
9119-595		7992		Yes	No

a. Feature codes for Enterprise Edition supersede codes for Standard Edition. All PowerVM Standard Edition function is supplied with the Extended Edition feature code.

b. Requires 1 year of Virtual I/O Server software maintenance.

Note: The feature codes for Enterprise Edition supersede codes for Standard Edition. All PowerVM Standard Edition function is supplied with the Extended Edition feature code.

The PowerVM Standard Edition feature is configured as an option for all System p servers except p5-590 and p5-595 systems, which include the PowerVM Standard Edition feature as part of the base system configuration. The software maintenance is charged additionally for all systems.

2.2 Introduction to the POWER Hypervisor

The POWER Hypervisor is the foundation of IBM PowerVM. Combined with features designed into the IBM POWER processors, the POWER Hypervisor delivers functions that enable capabilities including dedicated-processor partitions, Micro-Partitioning, virtual processors, IEEE VLAN compatible virtual switch, virtual Ethernet adapters, virtual SCSI adapters, and virtual consoles.

The POWER Hypervisor is a firmware layer sitting between the hosted operating systems and the server hardware, as shown in Figure 2-5 on page 28. The POWER Hypervisor is always installed and activated, regardless of system configuration. The POWER Hypervisor has no specific or dedicated processor resources assigned to it.

The POWER Hypervisor performs the following tasks:

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

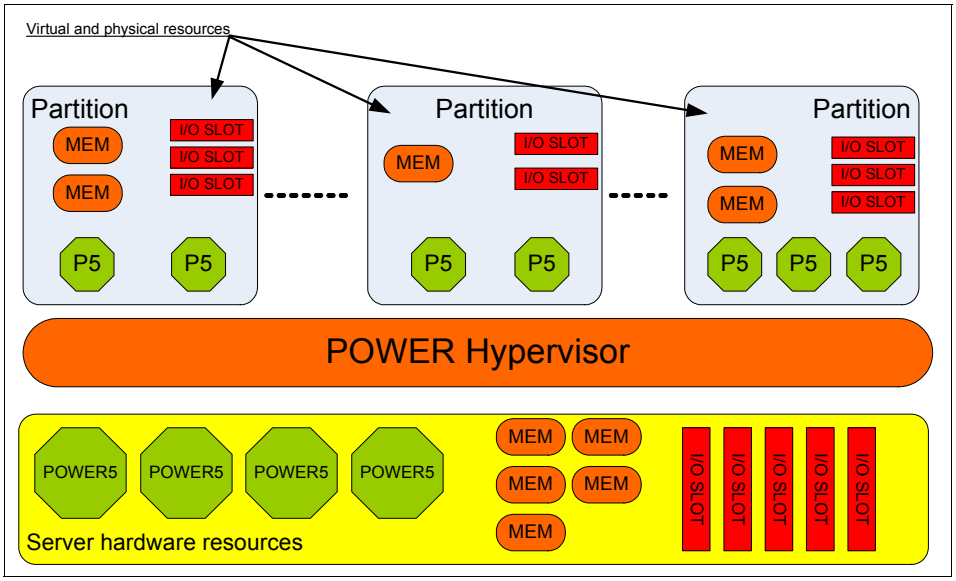


Figure 2-5 The POWER Hypervisor abstracts the physical server hardware

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

Through Micro-Partitioning, the POWER Hypervisor allows multiple instances of operating systems to run on POWER5-based and POWER6-based servers concurrently. The supported operating systems are listed in Table 1-2 on page 4.

2.2.1 POWER Hypervisor virtual processor dispatch

Physical processors are abstracted by the POWER Hypervisor and presented to micro-partitions as virtual processors. Micro-partitions are allocated a number of virtual processors when they are created. The number of virtual processors allocated to a micro-partition can be dynamically changed.

The number of virtual processors in a micro-partition and in all micro-partitions does not necessarily have any correlation to the number of physical processors in the physical shared-processor pool. In terms of capacity, a physical processor can support up to ten virtual processors.

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

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

Dispatch mechanism

To illustrate the dispatch mechanism, consider three partitions with two, one, and three virtual processors. These six virtual processors are mapped to two physical processors as shown in Figure 2-6 on page 30.

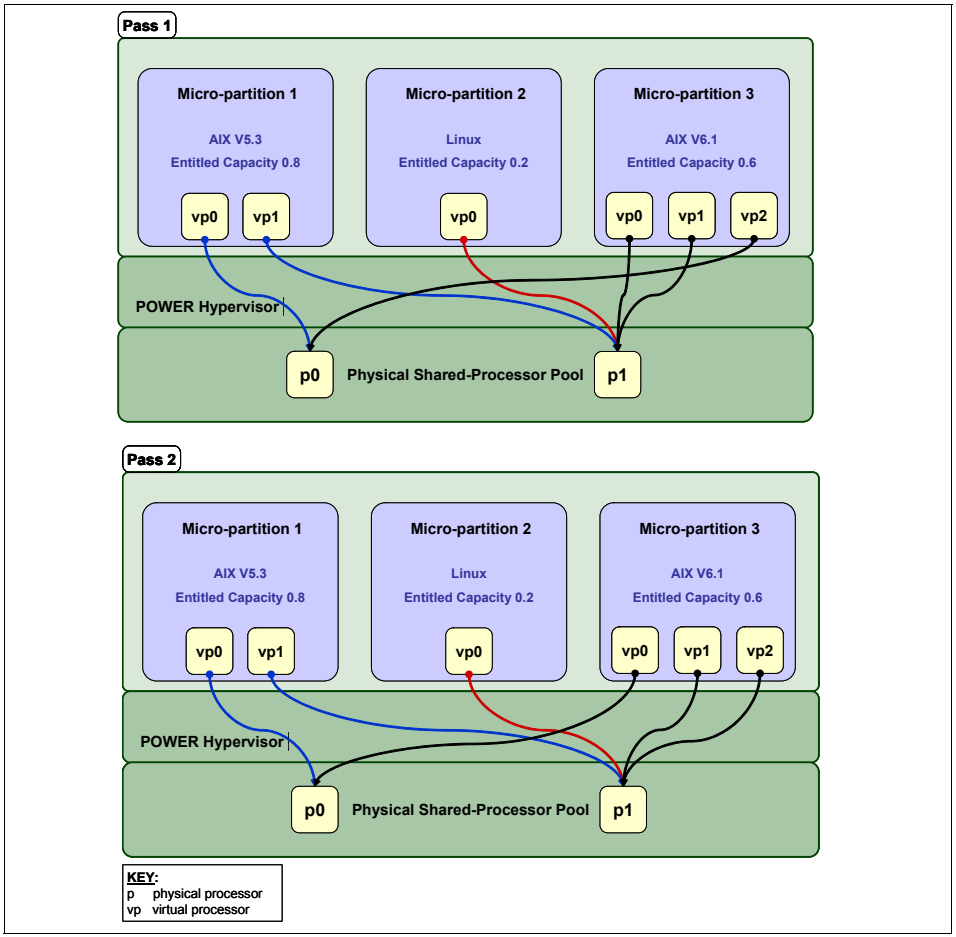


Figure 2-6 Virtual processor to physical processor mapping: pass 1 and pass 2

Figure 2-7 on page 31 shows two POWER Hypervisor dispatch cycles for two micro-partitions with a total of six virtual processors dispatched on to two physical processors.

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

Micro-partition 2 is configured with one virtual processor and an entitled capacity of 0.2 processing units, entitling it to 20 percent usage of a physical processor during each dispatch interval.

Note: It is possible for a virtual processor to be dispatched more than once during a dispatch interval. In the first dispatch interval, the workload executing on virtual processor 1 in micro-partition 1 is not continuous on the physical processor resource. This can happen if the operating system cedes cycles, and is reactivated by a prod hcall.

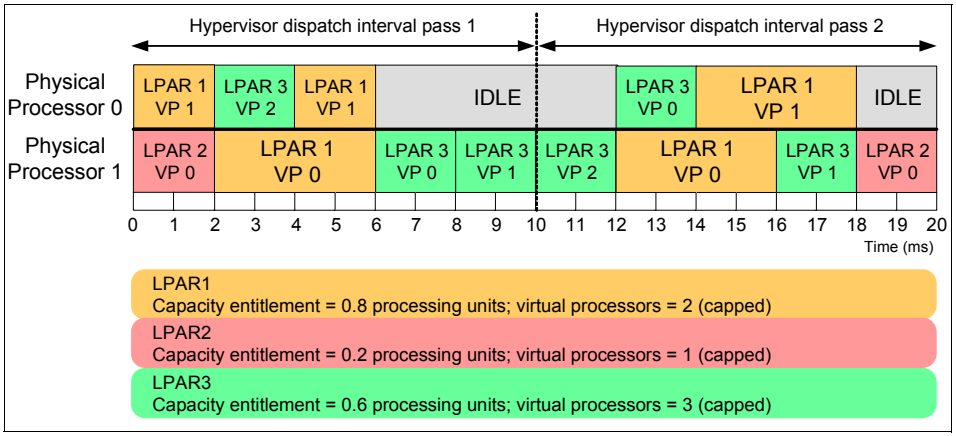


Figure 2-7 Micro-Partitioning processor dispatch

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

Processor affinity

To optimize the use of cache memory and minimize context switching, the POWER Hypervisor is designed to dispatch virtual processors on the same physical processor across dispatch cycles where possible. This behavior is called processor affinity.

The POWER Hypervisor will always attempt to dispatch the virtual processor on to the same physical processor that it previously ran on. If this is not possible, the POWER Hypervisor will broaden its search out to the other processor on the POWER5 or POWER6 chip, then to another chip on the same chip module.The System p affinity provided by the POWER Hypervisor is conducted in the following order:

1. Chip: the other processor core on the POWER5 or POWER6 chip)
2. Chip module: a processor core within the same chip module

3. Processor card/book (dependant on system type): a processor within the same processor card/book

System monitoring and statistics

The sharing of system resources with micro-partitions and the use of dual-threaded processor cores with simultaneous multithreading challenges the traditional performance data collection and reporting tools.

Within the physical processor architecture are registers to capture an accurate cycle count. This enables the measurement of micro-partition activity during the time slices dispatched on a physical processor.

The performance data collection and reporting tools are discussed in *Advanced POWER Virtualization: Managing and Monitoring*, SG24-7590.

Monitoring Hypervisor hcalls

From the introduction of AIX 5L Version 5.3, the commands **`lparstat`** and **`mpstat`** can be used to display the POWER Hypervisor and virtual processor affinity statistics. These commands are discussed in detail in *Advanced POWER Virtualization: Managing and Monitoring*, SG24-7590.

2.2.2 POWER Hypervisor and virtual I/O

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

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

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

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

I/O types supported

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

- SCSI
- Ethernet
- System Port (virtual console)

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

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

Virtual SCSI is covered in detail in 2.6, “Virtual SCSI introduction” on page 75; virtual Ethernet and the shared Ethernet adapter are discussed in 2.7, “Virtual and Shared Ethernet introduction” on page 83.

2.2.3 System port (virtual TTY/console support)

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

Depending on the system configuration, the operating system console can be provided by the HMC or IVM virtual TTY.

Note: The serial ports on an HMC-based system are inactive. Partitions requiring a TTY device must have an async adapter defined. The async adapter can be dynamically moved into or out of partitions with dynamic LPAR operations.

On the IVM, the serial ports are configured and active. They are used for initial configuration of the system.

2.3 Overview of Micro-Partitioning technologies

Micro-Partitioning is the ability to distribute the processing capacity of one or more physical processors among one or more logical partitions. Thus, processors are shared amongst logical partitions.

The benefit of Micro-Partitioning is that it allows significantly increased overall utilization of processor resources within the system. The micro-partition is provided with a processor entitlement—the processor capacity guaranteed to it by the POWER Hypervisor. A micro-partition must have a minimum of 0.1 of the capacity of a physical processor and can be up to the capacity of the system. Granularity of processor entitlement is 0.01 of a physical processor once the

minimum is met. Such fine granularity of processor capacity allocation to micro-partitions means efficient use of processing capacity.

There are a range of technologies associated with Micro-Partitioning. This section discusses the following:

- Micro-partitions
- Physical shared-processor pool
- Multiple Shared-Processor Pools
- Shared dedicated capacity
- Capacity Upgrade on Demand
- Dynamic processor deallocation and processor sparing
- Dynamic resources
- Shared-processor considerations

2.3.1 Micro-partitions

The virtualization of physical processors in POWER5 and POWER6 systems introduces an abstraction layer that is implemented within the IBM POWER Hypervisor. The POWER Hypervisor abstracts the physical processors and presents a set of virtual processors to the operating system within the micro-partitions on the system.

The operating system sees only the virtual processors and dispatches runnable tasks to them in the normal course of running a workload.

A micro-partition can have a processor entitlement from a minimum of 0.1 of a processor up to the total processor capacity in the system. The granularity of processor entitlement is 0.01 of a processor, allowing entitlement to be precisely determined and configured.

Micro-partitions may have processor entitlements of, say, 1.76, 0.14, 6.48 (all relating to the capacity of a physical processor). Thus, micro-partitions share the available processing capacity, potentially giving rise to multiple partitions executing on the same physical processor.

Micro-Partitioning is supported across the entire POWER5 and POWER6 server range, from entry level to the high-end systems.

While micro-partitions can be created with a minimum of 0.1 processing units, you can calculate the maximum number of micro-partitions for your system:

- 10 micro-partitions per activated processor.
- Maximum number of partitions on any System p server is 254.

In contrast, dedicated-processor LPARs can only be allocated whole processors, so the maximum number of dedicated-processor LPARs in a system is equal to the number of physical activated processors.

It is important to point out that the maximum number of micro-partitions supported for your system may not be the most practical configuration. Based on production workload demands, the number of micro-partitions that your system should use may be less.

Micro-partitions still require dedicated memory, but the I/O requirements of a micro-partition can be supported through either physical and/or virtual resources. A micro-partition can own dedicated network and disk-storage resources using dedicated physical adapters. Alternatively, micro-partitions may have some or all of the Ethernet and/or disk I/O resources satisfied through the use of virtual Ethernet and virtual SCSI.

Partitions are created and orchestrated by the HMC or IVM. When you start creating a partition, you have to choose between a micro-partition and a dedicated processor LPAR.

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

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

These settings are the topic of the following sections.

Processing units of capacity

Processing capacity can be configured in fractions of 0.01 processors. The minimum amount of processing capacity that has to be assigned to a micro-partition is 0.1 processors.

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

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

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

A micro-partition is guaranteed to receive its capacity entitlement under all systems and processing circumstances.

Capped and uncapped mode

Micro-partitions have a specific processing mode that determines the maximum processing capacity given to them from their Shared-Processor Pool.

The processing modes are:

Uncapped mode The processing capacity can exceed the entitled capacity when resources are available in their Shared-Processor Pool and the micro-partition is eligible to run. Extra capacity is distributed on a weighted basis. You must specify the uncapped weight of each micro-partition when it is created.

Capped mode The processing capacity given can never exceed the entitled capacity of the micro-partition.

If there is competition for additional processing capacity amongst several uncapped micro-partitions, the POWER Hypervisor distributes unused processor capacity to the eligible micro-partitions in proportion to each micro-partition's uncapped weight. The higher the uncapped weight of a micro-partition, the more processing capacity the micro-partition will receive.

The uncapped weight must be a whole number from 0 to 255. The default uncapped weight for uncapped micro-partitions is 128. A particular micro-partition's share of the unused capacity can be estimated using the following formula:

$$AdditionalCapacityShare = UCk \times \frac{WPn}{rP} \Big/ \sum_{0} We$$

Where:

AdditionalCapacityShare
: Share of unused processing capacity to be allocated to a particular partition (in *processor units* x 100)

UCk
: Unused processor capacity available in their Shared-Processor Pool for the dispatch window (in processor units)

WPn
: Uncapped weight of the particular uncapped micro-partition

rP
: The number of runnable (eligible) micro-partitions for this dispatch window

ΣWe
: Sum of the uncapped weights of all runnable uncapped micro-partitions

So additional capacity for an eligible uncapped micro-partition is computed by dividing its uncapped weight by the sum of the uncapped weights for all uncapped partitions that are currently runnable in the dispatch window.

The following is an example of this:

UCk = 200
: There are 200 units of unused processing capacity available for reallocation to eligible micro-partitions (200 = 2.0 processors)

WPn = 100
: The uncapped weighting of the micro-partition that is the subject of this calculation

rP = 5
: There are 5 runnable uncapped micro-partitions competing for the unused processor capacity denoted by UCk

ΣWe = 800
: The sum of the uncapped weighting of the runnable uncapped micro-partitons competing for the unused processor capacity

From this data we can see that the additional capacity share is:

$$AdditionalCapacityShare = 200 \times \frac{100}{800}$$

which gives:

$$AdditionalCapacityShare = 25$$

In this example, the AdditionalCapacityShare of 25 equates to 0.25 processor units.

Important: If you set the uncapped weight at 0, the POWER Hypervisor treats the micro-partition as a capped micro-partition. A micro-partition with an uncapped weight of 0 cannot be allocated additional processing capacity above its entitled capacity.

A weight of 0 allows automated workload management software to provide the equivalent function as a dynamic LPAR operation to change uncapped to capped (and the reverse).

Virtual processors

A virtual processor is a depiction or a representation of a physical processor that is presented to the operating system running in a micro-partition. The processing entitlement capacity assigned to a micro-partition, be it a whole or a fraction of a processing unit, will be distributed by the server firmware equally between the virtual processors within the micro-partition to support the workload. For example, if a micro-partition has 1.60 processing units and two virtual processors, each virtual processor will have the capacity of 0.80 processing units.

A virtual processor cannot have a greater processing capacity than a physical processor. The capacity of a virtual processor will be equal to or less than the processing capacity of a physical processor.

Selecting the optimal number of virtual processors depends on the workload in the partition. By default, the number of processing units that you specify is rounded up to the minimum whole number of virtual processors needed to satisfy the assigned number of processing units. The default settings maintain a balance of virtual processors to processor units. For example:

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

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

A micro-partition must have enough virtual processors to satisfy its assigned processing capacity. This capacity can include its entitled capacity and any additional capacity above its entitlement if the micro-partition is uncapped.

Therefore, the upper boundary of processing capacity in a micro-partition is determined by the number of virtual processors that it possesses. For example, if you have a partition with 0.50 processing units and one virtual processor, the

partition cannot exceed 1.00 processing units. However, if the same partition with 0.50 processing units was assigned two virtual processors and processing resources were available, the partition could use an additional 1.50 processing units.

The minimum number of processing units that can be allocated to each virtual processor is dependent on the server model. The maximum number of processing units that can be allocated to a virtual processor is always 1.00. Additionally, the number of processing units cannot exceed the total processing unit within a Shared-Processor Pool.

Virtual processor folding

The capability of *virtual processor folding* started with maintenance level 3, AIX 5L V5.3. In order for an uncapped micro-partition to take full advantage of unused processor capacity in the physical shared-processor pool, it must have enough virtual processors defined. In the past, these additional virtual processors can remain idle for substantial periods of time and consume a small but valuable amount of resources.

Virtual processor folding effectively puts idle virtual processors into a *hibernation state* so that they do not consume any resources. There are several important benefits of this feature including improved processor affinity, reduced POWER Hypervisor workload, and increased average time a virtual processor executes on a physical processor.

Following are the characteristics of the virtual processor folding feature:

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

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

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

Dedicated processors

Dedicated processors are whole processors that are assigned to dedicated-processor partitions (LPARs). The minimum processor allocation for an LPAR is one (1) whole processor, and can be as many as the total number of installed processors in the server.

Each processor is wholly dedicated to the LPAR. It is not possible to mix shared processors and dedicated processors in the same partition.

By default, the POWER Hypervisor will make the processors of a powered-off LPAR available to the physical shared-processor pool. When the processors are in the physical shared processor pool, an uncapped partition that requires more processing resources can utilize the additional processing capacity. However, when the LPAR is powered on, it will regain the processors and they will become dedicated to the newly powered-on LPAR.

To prevent dedicated processors from being used in the physical shared-processing pool while they are not part of a powered-on LPAR, you can disable this function on the HMC by deselecting the “Processor Sharing: Allow when partition is inactive” check box in the partition’s properties.

Note: The option “Processor Sharing: Allow when partition is inactive” is activated by default. It is not part of profile properties and it cannot be changed dynamically. For more information, see 3.5.8, “Dedicated donating processors” on page 215.

Tip: If the “Allow when partition is inactive” box is unchecked on the HMC and you want to lock the dedicated processors from being released to the physical-shared processor pool, you can boot the partition to SMS and deactivate the partition. This will hold the processors and stop them from being included in the physical shared-processor pool.

2.3.2 POWER5 physical shared-processor pool

In POWER5-based System p servers, a physical shared-processor pool is a set of physical processors that are not dedicated to any logical partition. Micro-Partitioning technology coupled with the POWER Hypervisor facilitates the sharing of processing units between micro-partitions.

An overview of the relationships between the POWER5 physical shared-processor pool, virtual processors, and micro-partitions can be seen in Figure 2-8 on page 41.

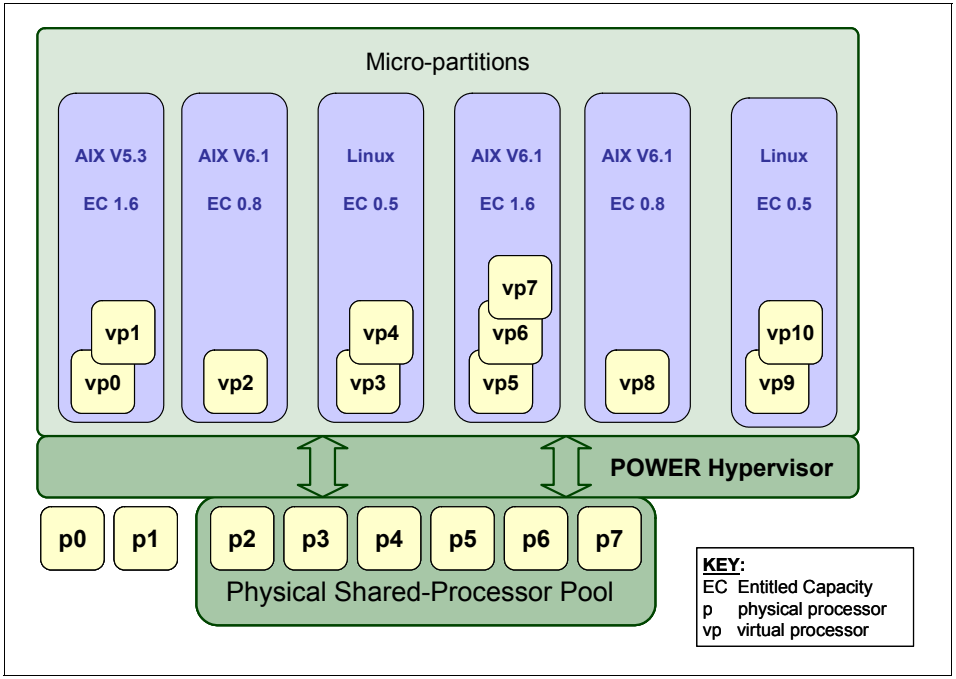


Figure 2-8 POWER5 physical shared processor pool and micro-partitions

Figure 2-8 shows that physical processors p0 and p1 are not assigned to the physical shared processor pool; they may be assigned to dedicated-processor partitions or awaiting activation.

Note: The new term *physical shared-processor pool* is used in this book to differentiate the pool of physical processors (the source of all processor capacity for micro-partitions) from the technology that implements *Multiple Shared-Processor Pools* in POWER6-based servers.

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

Affinity scheduling is designed to preserve the content of memory caches, so that the working data set of a job can be read or written in the shortest time period possible. Affinity is actively managed by the POWER Hypervisor.

Figure 2-9 shows the relationship between two partitions using a physical shared-processor pool of a single physical processor. One partition has two virtual processors and the other a single virtual processor. The figure also shows how the capacity entitlement is evenly divided over the number of virtual processors.

When you set up a partition profile, you set up the *desired*, *minimum*, and *maximum* values you want for the profile. When a partition is started, the system chooses the partition's entitled processor capacity from this specified capacity range. The value that is chosen represents a commitment of capacity that is reserved for the partition. This capacity cannot be used to start another micro-partition because the overcommittment of capacity is not permitted.

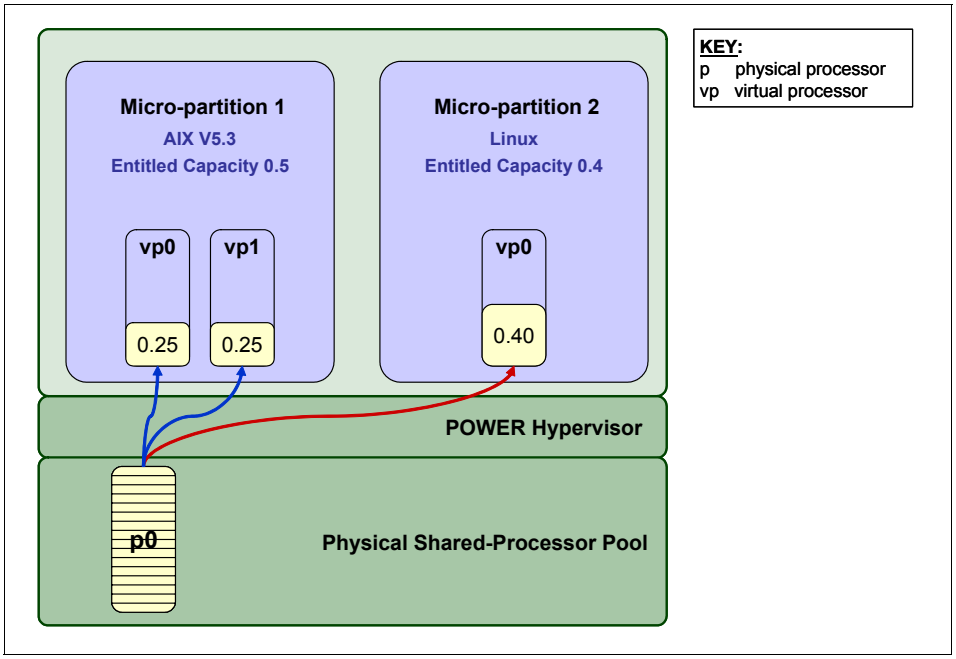


Figure 2-9 Distribution of processor capacity entitlement on virtual processors

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

The processor entitlement capacity is reserved for the partitions in the sequence the partitions are started. For example, consider a scenario where a physical shared-processor pool that has 2.0 processing units is available.

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

- Partition 1 activated
  Min. = 1.0, max = 2.0, desired = 1.5
  Allocated capacity entitlement: 1.5
- Partition 2 activated
  Min. = 1.0, max = 2.0, desired = 1.0
  Partition 2 does not start because the minimum capacity is not met.
- Partition 3 activated
  Min. = 0.1, max = 1.0, desired = 0.8
  Allocated capacity entitlement: 0.5

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

Figure 2-10 shows the behavior of a capped micro-partition of the physical shared-processor pool. Micro-partitions using the capped mode are not able to assign more processing capacity from the physical shared-processor pool than the capacity entitlement will allow.

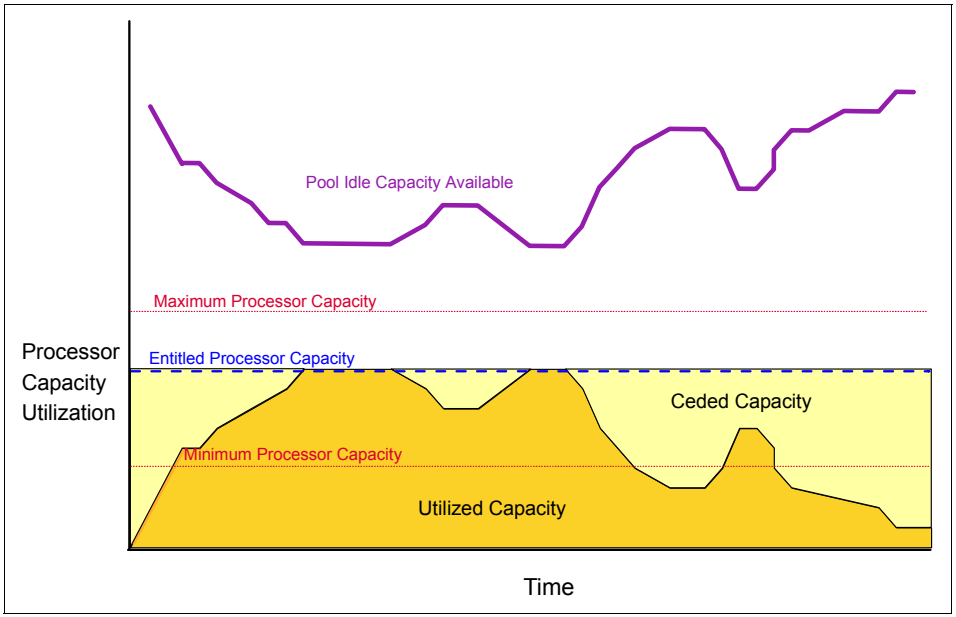


Figure 2-10 Example of capacity distribution of a capped micro-partition

Figure 2-11 shows the usage of the physical shared-processor pool by an uncapped partition. The uncapped partition is able to utilize the unused processing capacity from the physical shared processor pool if it requires more than its entitled capacity.

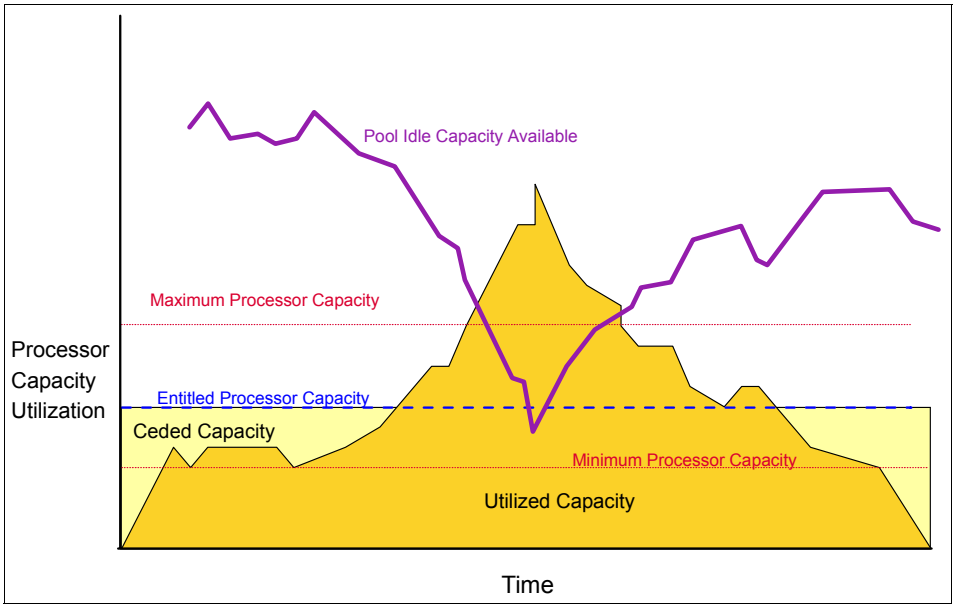


Figure 2-11 Example of capacity distribution of an uncapped micro-partition

2.3.3 POWER6 Multiple Shared-Processor Pools

Multiple Shared-Processor Pools (MSPPs) is a new capability supported on POWER6 servers. This capability allows a system administrator to create a *set of micro-partitions* with the purpose of controlling the processor capacity that can be consumed from the physical shared-processor pool.

To implement MSPPs, there is a set of underlying techniques and technologies. An overview of the architecture of Multiple Shared-Processor Pools can be seen in Figure 2-12.

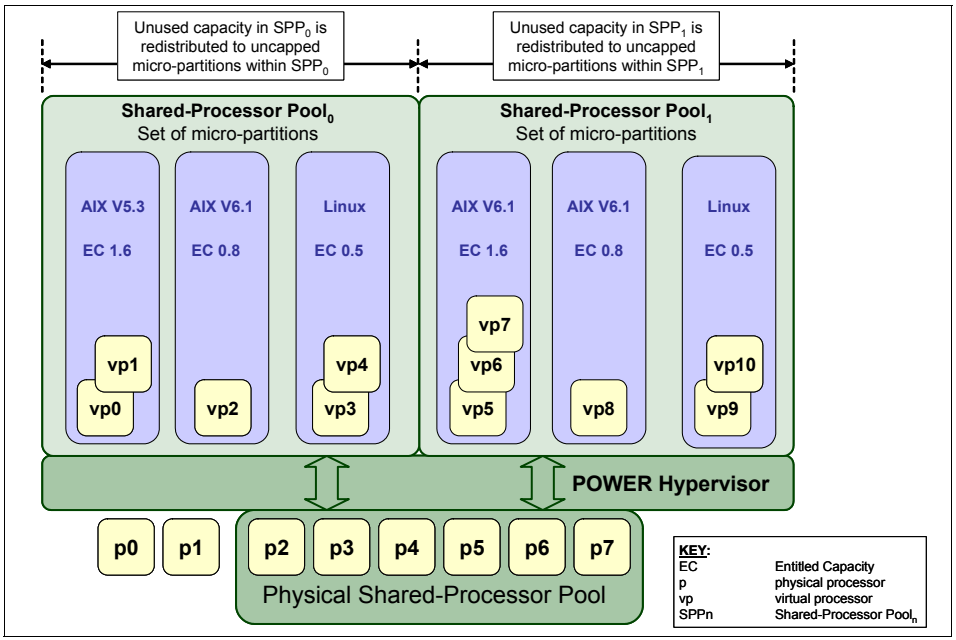


Figure 2-12 Overview of the architecture of Multiple Shared-Processor Pools

Micro-partitions are created and then identified as members of either the default Shared-Processor Pool$_0$ or a user-defined Shared-Processor Pool$_n$. The virtual processors that exist within the set of micro-partitions are monitored by the POWER Hypervisor and processor capacity is managed according to user-defined attributes.

With the Multiple Shared-Processor Pools capability, there is a set of terminology that should be understood:

Physical Shared-Processor Pool

The set of processors installed on a System p server that are used to run a set of micro-partitions. There is a maximum of one physical shared-processor pool on POWER6-based System p servers.

All active physical processors are part of the physical-processor pool unless they are assigned to a dedicated-processor partition where:

- The LPAR is active and is not capable of capacity donation, or
- The LPAR is inactive (powered-off) and the systems administrator has chosen not to make the processors available for

shared-processor work, see 3.5.8, “Dedicated donating processors” on page 215

Shared-Processor Pool$_n$ (SPP$_n$)
: A specific group of micro-partitions (and their associated virtual processors) that are designated by the system administrator to be in a set for the purpose of controlling the processor capacity that the set of micro-partitions can consume from the physical shared-processor pool. The set of micro-partitions form a unit through which processor capacity from the physical shared-processor pool can be managed.

> **Note:** The subscript n indicates the identifier for the Shared-Processor Pool. For example, the term SPP$_0$ indicates the Shared-Processor Pool ID=0 (the default Shared-Processor Pool has the ID=0).

Maximum Pool Capacity
: Each Shared-Processor Pool has a maximum capacity associated with it. The Maximum Pool Capacity defines the upper boundary of the processor capacity that can be utilized by the set of micro-partitions in the Shared-Processor Pool. The Maximum Pool Capacity must be represented by a whole number of processor units.

Reserved Pool Capacity
: The system administrator can assign an entitled capacity to a Shared-Processor Pool for the purpose of reserving processor capacity from the physical shared-processor pool for the express use of the micro-partitions in the Shared-Processor Pool. The Reserved Pool Capacity is in addition to the processor capacity entitlements of the individual micro-partitions in the Shared-Processor Pool. The Reserved Pool Capacity is distributed amongst uncapped micro-partitions in the Shared-Processor Pool according to their uncapped weighting. The default value for the Reserved Pool Capacity is zero (0).

Entitled Pool Capacity
: Associated with a Shared-Processor Pool is an Entitled Pool Capacity. The Entitled Pool Capacity of a Shared-Processor Pool defines the guaranteed processor capacity that is available to the group of micro-partitions in the Shared processor Pool. The Entitled Pool Capacity is the sum of the entitlement capacities of the micro-partitions in the Shared-Processor Pool plus the Reserved Pool Capacity.

  This can be represented by the following formula:

$$EntitledPoolCapacity = \left(\sum_{0}^{n} MicropartionEntitlement \right) + ReservedPoolCapacity$$

where n represents the number of micro-partitions in the Shared-Processor Pool.

Using the information in Table 2-3 as an example and a Reserved Pool Capacity of 1.5, it is easy to calculate the Entitled Pool Capacity.

Table 2-3 Entitled capacities for micro-partitions in a Shared-Processor Pool

Micro-partitions in a Shared-Processor Pool	Entitled capacity for micro-partition
Micro-partition 0	0.5
Micro-partition 1	1.75
Micro-partition 2	0.25
Micro-partition 3	0.25
Micro-partition 4	1.25
Micro-partition 5	0.50

The sum of the entitled capacities for the micro-partitions in this Shared-Processor Pool is 4.50, which gives us the following calculation:

$$EntitledPoolCapacity = 4.50 + 1.50$$

which gives:

$$EntitledPoolCapacity = 6.0$$

If the System p server is under heavy load, each micro-partition within a Shared-Processor Pool is guaranteed its processor entitlement plus any capacity that it may be allocated from the Reserved Pool Capacity if the micro-partition is uncapped.

If some micro-partitions in a Shared-Processor Pool do not use their capacity entitlement, the unused capacity is ceded and other uncapped micro-partitions within the same Shared-Processor Pool are allocated the additional capacity according to their uncapped weighting. In this way, the Entitled Pool Capacity of a Shared-Processor Pool is distributed to the set of micro-partitions within that Shared-Processor Pool.

The example in Figure 2-13 shows a Shared-Processor Pool under load; some micro-partitions require more processing capacity than their entitled capacity.

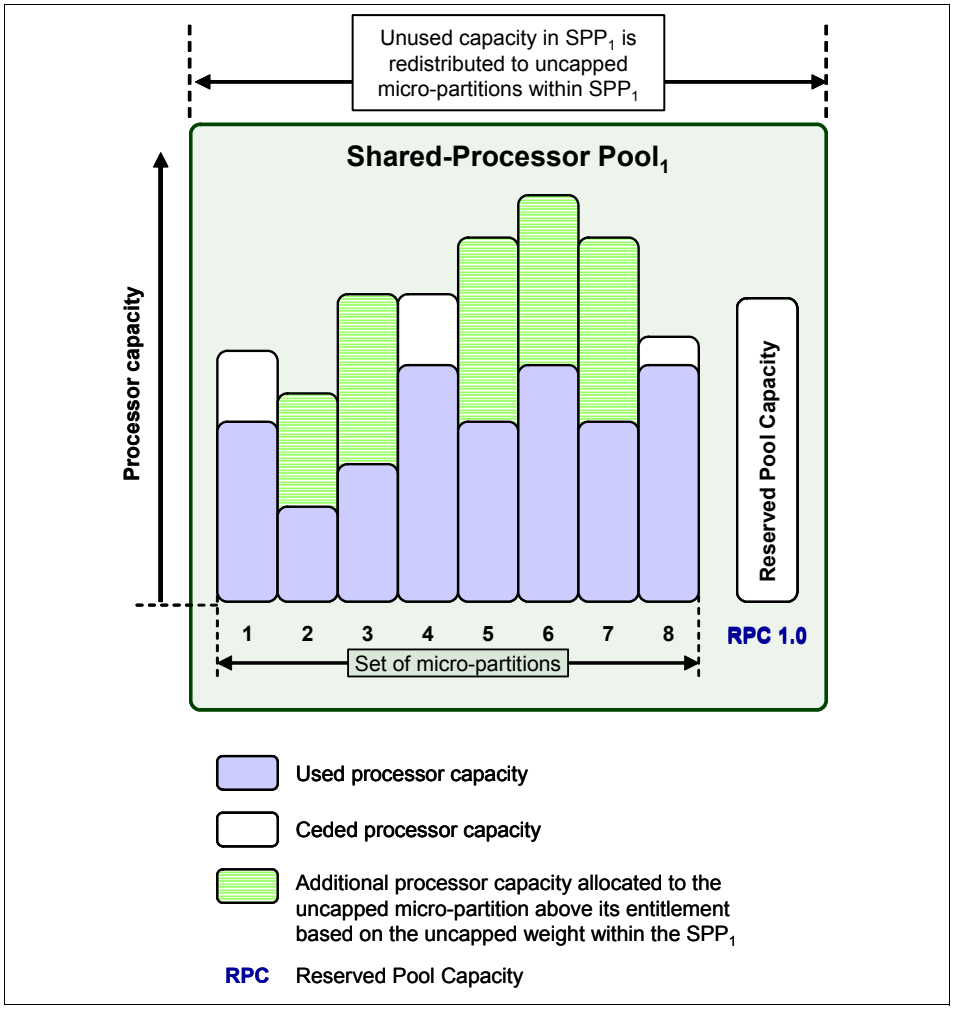


Figure 2-13 Redistribution of ceded capacity within Shared-Processor Pool$_1$

Each micro-partition in Shared-Processor Pool$_1$ (SPP$_1$) shown in Figure 2-13 is guaranteed to receive its processor entitlement if it is required. However, some micro-partitions in SPP$_1$ have not used their entire entitlement and so will cede the capacity. The ceded capacity is redistributed to the other uncapped micro-partitions within SPP$_1$ on an uncapped weighted basis. However, the ceded capacity may not be enough to satisfy the demand, and so additional capacity can be sourced from the Reserve Pool Capacity of the

Shared-Processor Pool and distributed according to the uncapped weight of the requesting micro-partitions.

All POWER6-based System p servers that support the Multiple Shared-Processor Pools capability will have a minimum of one (the default) Shared-Processor Pool and up to a maximum of 64 Shared-Processor Pools. The default Shared-Processor Pool always has an ID of 0 and is referred to as Shared-Processor $Pool_0$ (SPP_0).

Figure 2-14 shows a case where Multiple Shared-Processor Pools have been created: SPP_1, SPP_2, and SPP_3. Each of the Multiple Shared-Processor Pools has its own Reserved Pool Capacity (not shown) which it can distribute to its micro-partitions on an uncapped weighted basis. In addition, each of the Multiple Shared-Processor Pools can accumulate unused/ceded processor capacity from the under-utilized micro-partitions and again redistribute it accordingly.

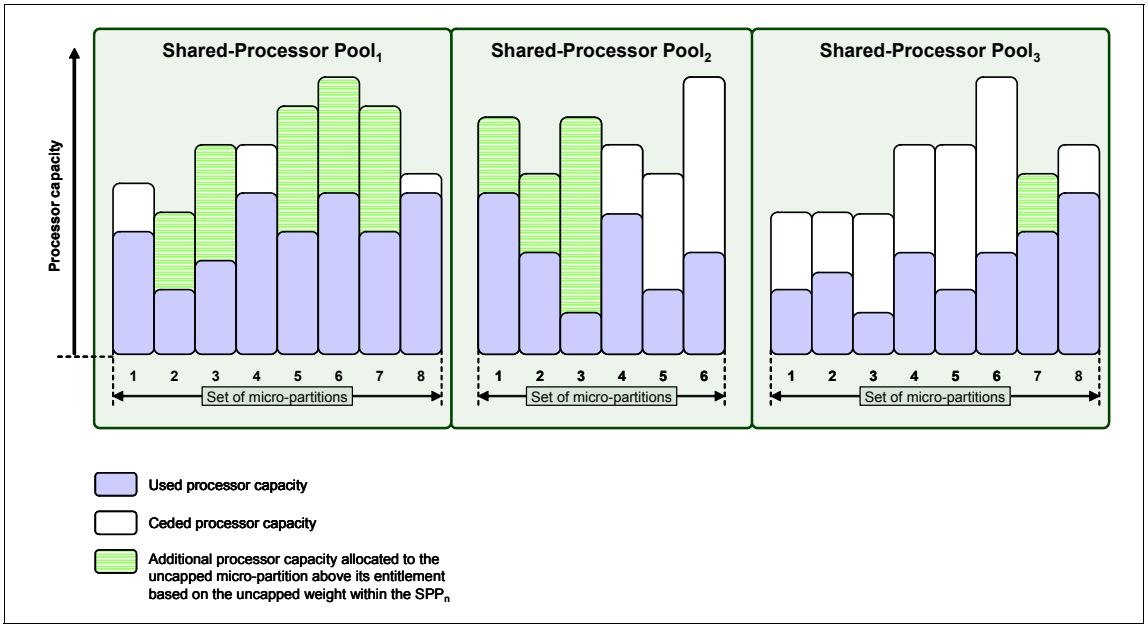


Figure 2-14 Example of Multiple Shared-Processor Pools

Note: The default Shared-Processor Pool has an ID value of zero (SPP_0) and is not shown for the sake of clarity at this stage.

SPP_1 appears to be heavily loaded as there is little unused capacity and several micro-partitions receiving addition capacity. SPP_2 has a moderate loading,

whereas SPP_3 is lightly loaded with most micro-partitions are ceding processor capacity.

Default Shared-Processor Pool (SPP_0)

On any POWER6-based System p server supporting Multiple Shared-Processor Pools, a default Shared-Processor Pool is always automatically defined. The default Shared-Processor Pool has a pool identifier of zero (SPP-ID = 0) and can also be referred to as SPP_0. The default Shared-Processor Pool has the same attributes as a user-defined Shared-Processor Pool except that these attributes are not directly under the control of the system administrator—they have fixed values.

Table 2-4 Attribute values for the default Shared-Processor Pool (SPP0)

SPP_0 attribute	**Description**	**Value**
Shared-Processor Pool ID	Default Shared-Processor Pool identifier	0
Maximum Pool Capacity	The maximum allowed capacity - the upper capacity boundary for the Shared-Processor Pool. For SPP_0, this value cannot be changed.	The value is equal to the number of active physical processors in the physical shared-processor pool (all capacity in the physical shared-processor pool). The number of processors can vary as physical processors enter or leave the physical shared-processor pool through dynamic/other partition activity.
Reserved Pool Capacity	Reserved processor capacity for this Shared-Processor Pool. For the default Shared-Processor Pool, this value cannot be changed.	0
Entitled Pool Capacity	Sum of the capacity entitlements of the micro-partitions in the default Shared-Processor Pool plus the Reserved Pool Capacity (which is always zero in the default Shared-Processor Pool)	Sum (total) of the entitled capacities of the micro-partitions in the default Shared-Processor Pool.

The maximum capacity of SPP_0 can change indirectly through system administrator action such as powering on a dedicated processor partition, or dynamically moving physical processors in or out of the physical shared-processor pool.

Note: The default behavior of the system, with only SPP_0 defined, is the current behavior of a POWER5 server with only a physical shared-processor pool defined. Micro-partitions are created within SPP_0 by default, and processor resources are shared in the same way.

Creating Multiple Shared-Processor Pools

The default Shared-Processor Pool (SPP_0) is automatically activated by the system and is always present. Its Maximum Pool Capacity is set to the capacity of the physical shared-processor pool. For SPP_0, the Reserved Pool Capacity is always 0.

All other Shared-Processor Pools exist, but by default, are inactive. By changing the Maximum Pool Capacity of a Shared-Processor Pool to a value greater than zero, it becomes active and can accept micro-partitions (either transferred from SPP_0 or newly created).

The system administrator can use the HMC to activate additional Multiple Shared-Processor Pools. As a minimum, the Maximum Pool Capacity will need to be specified. If you wish to specify a Reserved Pool Capacity, there must be enough unallocated physical processor capacity to guarantee the entitlement.

In the example shown in Figure 2-15, a Shared-Processor Pool has been activated. Within the newly activated Shared-Processor $Pool_1$, a set of eight micro-partitions have also been created.

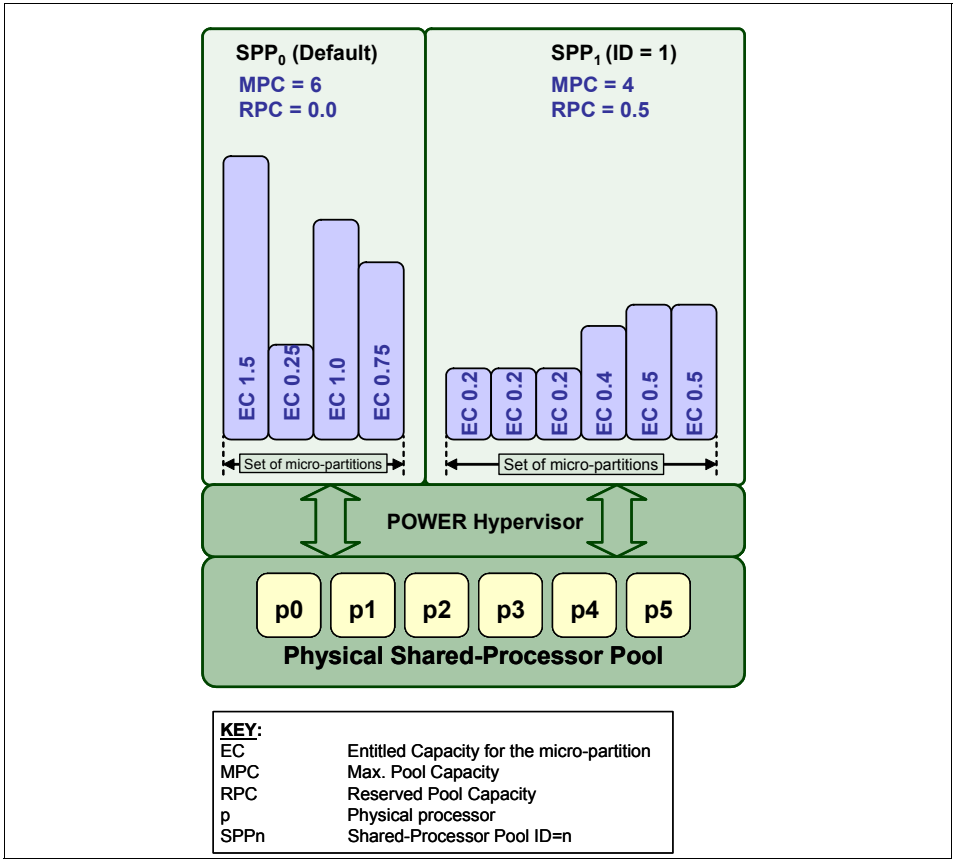


Figure 2-15 POWER6-based server with two Shared-Processor Pools defined

Shared-Processor $Pool_1$ has been activated with the following attribute values:

- ID=1
- Maximum Pool Capacity = 4
- Reserved Pool Capacity = 0.5

You can calculate the Entitled Pool Capacity by totalling the entitled capacities for the micro-partitions = 2.0 and adding the Reserved Pool Capacity of 0.5. Thus, 2.5 is the Entitled Pool Capacity for Shared-Processor $Pool_1$.

The Maximum Pool Capacity is set to 4. This means that if the Entitled Pool Capacity for SPP_1 is totally consumed by the micro-partitions within SPP_1, then

any uncapped micro-partitions in SPP_1 may be eligible for additional processor capacity that has been ceded by micro-partitions in other Shared-Processor Pools in the system. As the Maximum Pool Capacity is set to 4, the additional processor capacity that the uncapped micro-partitions in SPP_1 can receive is a maximum of an additional 1.5.

Levels of processor capacity resolution

There are two levels of processor capacity resolution implemented by the POWER Hypervisor and Multiple Shared-Processor Pools:

$Level_0$ The first level, $Level_0$, is the resolution of capacity within the same Shared-Processor Pool. Unused processor cycles from within a Shared-Processor Pool are harvested and then redistributed to any eligible micro-partition within the same Shared-Processor Pool.

$Level_1$ When all $Level_0$ capacity has been resolved within the Multiple Shared-Processor Pools, the POWER Hypervisor harvests unused processor cycles and redistributes them to eligible micro-partitions regardless of the Multiple Shared-Processor Pools structure. This is the second level of processor capacity resolution.

You can see the two levels of unused capacity redistribution implemented by the POWER Hypervisor in Figure 2-16.

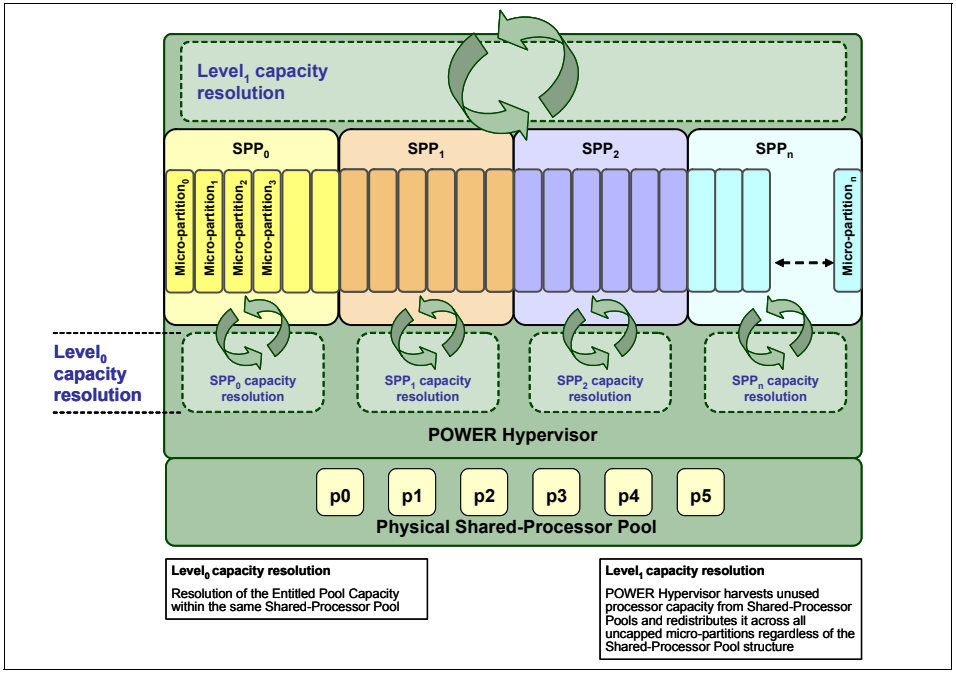


Figure 2-16 The two levels of unused capacity redistribution

Capacity allocation above the Entitled Pool Capacity ($Level_1$)

The POWER Hypervisor initially manages the Entitled Pool Capacity at the Shared-Processor Pool level. This is where unused processor capacity within a Shared-Processor Pool is harvested and then redistributed to uncapped micro-partitions within the same Shared-Processor Pool. This level of processor capacity management is sometimes referred to as $Level_0$ capacity resolution.

At a higher level, the POWER Hypervisor harvests unused processor capacity from the Multiple Shared-Processor Pools that do not consume all of their Entitled Pool Capacity. If a particular Shared-Processor Pool is heavily loaded and some of the uncapped micro-partitions within it require additional processor capacity (above the Entitled Pool Capacity) then the POWER Hypervisor redistributes some of the extra capacity to the uncapped micro-partitions. This level of processor capacity management is sometimes referred to as $Level_1$ capacity resolution.

To redistribute unused processor capacity to uncapped micro-partitions in Multiple Shared-Processor Pools above the Entitled Pool Capacity, the POWER Hypervisor uses a higher level of redistribution, $Level_1$.

Important: $Level_1$ capacity resolution - when allocating additional processor capacity in excess of the Entitled Pool Capacity of the Shared-Processor Pool, the POWER Hypervisor takes the uncapped weights of *all micro-partitions in the system* into account, *regardless of the Multiple Shared-Processor Pool structure*.

Where there is unused processor capacity in underutilized Shared-Processor Pools, the micro-partitions within the Shared-Processor Pools cede the capacity to the POWER Hypervisor.

In busy Shared-Processor Pools where the micro-partitions have used all of the Entitled Pool Capacity, the POWER Hypervisor will allocate additional cycles to micro-partitions where:

- The Maximum Pool Capacity of the Shared-Processor Pool hosting the micro-partition has not been met.
- (and) The micro-partition is uncapped.
- (and) The micro-partition has enough virtual-processors to take advantage of the additional capacity.

Under these circumstances, the POWER Hypervisor allocates additional processor capacity to micro-partitions on the basis of their uncapped weights independent of the Shared-Processor Pool hosting the micro-partitions. This can be referred to as $Level_1$ capacity resolution. Consequently, when allocating

additional processor capacity in excess of the Entitled Pool Capacity of the Shared-Processor Pools, the POWER Hypervisor takes the uncapped weights of all micro-partitions in the system into account, regardless of the Multiple Shared-Processor Pools structure.

You can see in Figure 2-16 on page 53 that in $Level_0$, the POWER Hypervisor restricts the scope for processor capacity management to the individual Shared-Processor Pool. It resolves capacity management issues within the restricted scope (bounded by the Shared-Processor Pool) before harvesting or allocating processor cycles at the higher level, $Level_1$.

Once $Level_0$ capacity management is resolved, unused processor capacity can be redistributed. $Level_1$ allocates these unused processor cycles on an uncapped-weighted basis looking across all the micro-partitions on the system regardless of the Shared-Processor Pools structure. Therefore, the scope for $Level_1$ is the total set of micro-partitions on the system, and the reallocation of cycles takes into account the uncapped weights of individual micro-partitions competing for additional processor capacity regardless of the Shared-Processor Pool definitions on the system.

Dynamic adjustment of Maximum Pool Capacity

The Maximum Pool Capacity of a Shared-Processor Pool, other than the default Shared-Processor $Pool_0$, can be adjusted dynamically from the HMC using either the graphical or CLI interface.

Dynamic adjustment of Reserve Pool Capacity

The Reserved Pool Capacity of a Shared-Processor Pool, other than the default Shared-Processor $Pool_0$, can be adjusted dynamically from the HMC using either the graphical or CLI interface.

Dynamic movement between Shared-Processor Pools

A micro-partition can be moved dynamically from one Shared-Processor Pool to another using the HMC using either the graphical or CLI interface. As the Entitled Pool Capacity is partly made up of the sum of the entitled capacities of the micro-partitions, removing a micro-partition from a Shared-Processor Pool will reduce the Entitled Pool Capacity for that Shared-Processor Pool. Similarly, the Entitled Pool Capacity of the Shared-Processor Pool that the micro-partition joins will increase.

Figure 2-17 depicts an example of a micro-partition moving from one Shared-Processor Pool to another.

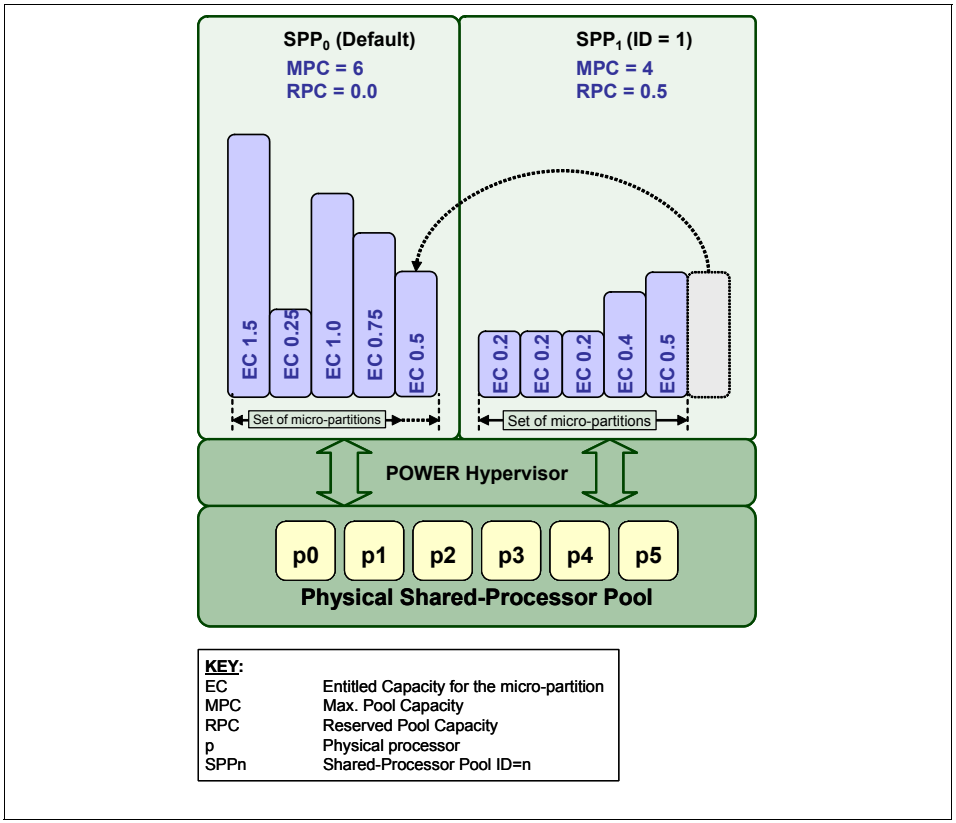


Figure 2-17 Example of a micro-partition moving between Shared-Processor Pools

The movement of micro-partitions between Shared-Processor Pools is really a simple reassignment of the Shared-Processor Pool-ID that a particular micro-partition is associated with. From the example in Figure 2-17 we can see that a micro-partition within Shared-Processor $Pool_1$ is reassigned to the default Shared-Processor $Pool_0$.

This movement reduces the Entitled Pool Capacity of Shared-Processor $Pool_1$ by 0.5 and correspondingly increases the Entitled Pool Capacity of Shared-Processor $Pool_0$ by 0.5 as well. The Reserved Pool Capacity and Maximum Pool Capacity values are not affected.

Note: The Maximum Pool Capacity must be equal to or greater than the Entitled Pool Capacity in a Shared-Processor Pool. If the movement of a micro-partition to a target Shared-Processor Pool pushes the Entitled Pool Capacity past the Maximum Pool Capacity, then movement of the micro-partition will fail.

Deleting a Shared-Processor Pool

Shared-Processor Pools cannot be deleted from the system. However, they are deactivated by setting the Maximum Pool Capacity and the Reserved Pool Capacity to zero. The Shared-Processor Pool will still exist but will not be active. Use the HMC interface to deactivate a Shared-Processor Pool. A Shared-Processor Pool cannot be deactivated unless all micro-partitions hosted by the Shared-Processor Pool have been removed.

Notes:

1. Shared-Processor Pools cannot be deleted from the system—they are deactivated.
2. A Shared-Processor Pool cannot be deactivated unless all micro-partitions hosted by the Shared-Processor Pool have been removed.

Live Partition Mobility and Multiple Shared-Processor Pools

A micro-partition may leave a Shared-Processor Pool due to PowerVM Live Partition Mobility. Similarly, a micro-partition may join a Shared-Processor Pool in the same way. When performing PowerVM Live Partition Mobility, you are given the opportunity to designate a destination Shared-Processor Pool on the target server to receive and host the migrating micro-partition.

As several simultaneous micro-partition migrations are supported by PowerVM Live Partition Mobility, it is conceivable to migrate the entire Shared-Processor Pool from one server to another.

2.3.4 Examples of Multiple Shared-Processor Pools

This section puts Multiple Shared-Processor Pool technologies and capabilities into a more relevant context.

Figure 2-18 provides an example of how a Web-facing deployment maps onto a set of micro-partitions within a Shared-Processor Pool structure. There are three Web servers, two application servers, and a single database server.

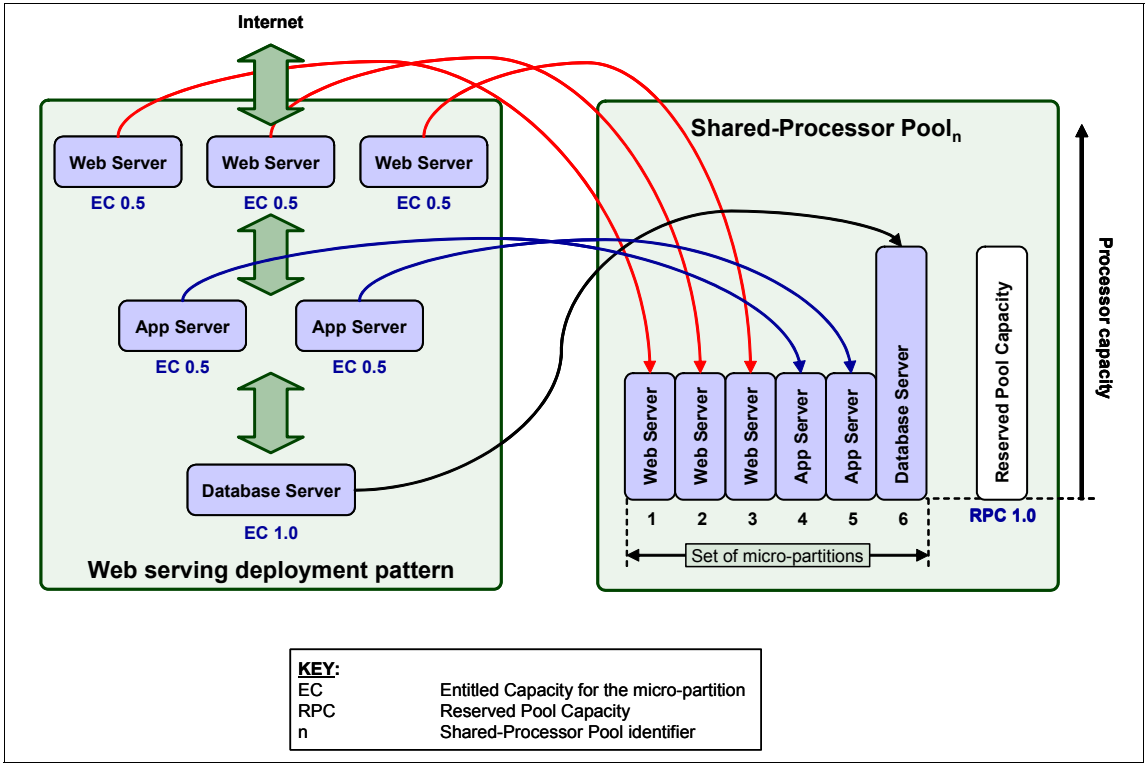


Figure 2-18 Example of a Web-facing deployment mapped onto a set of micro-partitions

Each of the Web server micro-partitions and application server micro-partitions have an entitled capacity of 0.5 and the database server micro-partition has an entitled capacity of 1.0, making the total entitled capacity for this group of micro-partitions 3.5 processors. In addition to this, Shared-Processor Pool$_n$ has a Reserved Pool Capacity of 1.0 which makes the Entitled Pool Capacity for Shared-Processor Pool$_n$ 4.5 processors.

If you assume that all of the micro-partitions in Shared-Processor Pool$_n$ are uncapped (and they have adequate virtual processors configured), then all micro-partitions will become eligible to receive extra processor capacity when required. The Reserved Pool Capacity of 1.0 ensures that there will always be some capacity to be allocated above the entitled capacity of the individual micro-partitions even if the micro-partitions in Shared-Processor Pool$_n$ are under heavy load.

You can see this resolution of processor capacity within Shared-Processor Pool$_n$ in Figure 2-19. The left of the diagram outlines the definition of Shared-Processor Pool$_n$ and the micro-partitions that make it up. The example on the right of the diagram emphasizes the processor capacity resolution that takes place within Shared-Processor Pool$_n$ during operation.

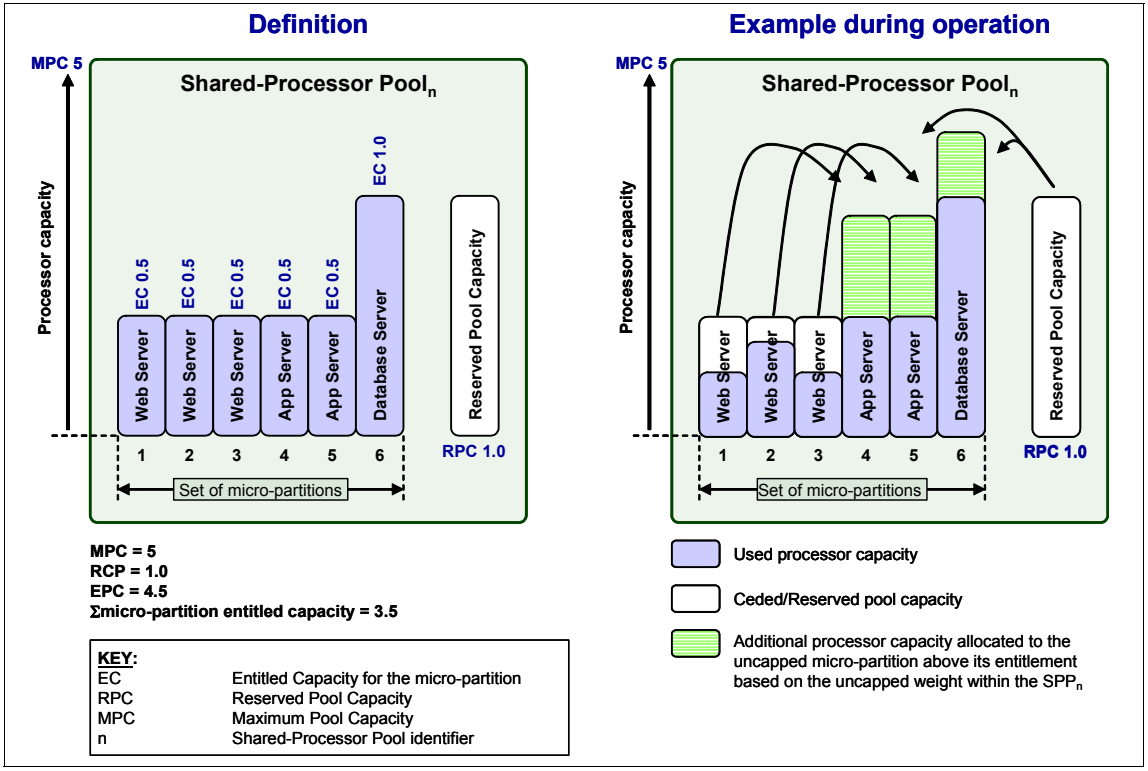


Figure 2-19 Web deployment using within a Shared-Processor Pool

In this example, during operation (one 10-ms POWER Hypervisor dispatch cycle) the Web servers are underutilized and cede processor capacity. However, the application servers and database server are heavily loaded and require far more processor cycles. These extra cycles are sourced from the Reserved Pool Capacity and the ceded capacity from the Web servers. This additional processor capacity is allocated to the application servers and database server using their uncapped weighting factor within Shared-Processor Pool$_n$ (Level$_0$ capacity resolution).

You will notice that the Maximum Pool Capacity is 0.5 above the Entitled Pool Capacity of Shared-Processor Pool$_n$ and so Level$_1$ capacity resolution can operate. This means that the uncapped micro-partitions within Shared-Processor

Pool$_n$ could also receive some additional processor capacity from Level$_1$ as long as the total capacity consumed is no greater than 5 processors (0.5 above the Entitled Pool Capacity).

Important: Level$_1$ capacity - additional processor capacity above the Entitled Pool Capacity and up to the Maximum Pool Capacity is *not* allocated to the Shared-Processor Pool for distribution to the micro-partitions within it.

The additional cycles are allocated directly to individual micro-partitions on an uncapped weighted basis within the system as a whole regardless of the Multiple Shared-Processor Pools structure. The total additional (Level$_1$) capacity allocated to micro-partitions within a particular Shared-Processor Pool cannot be greater than the Maximum Pool Capacity for that Shared-Processor Pool.

The example shown in Figure 2-19 on page 59 outlines a functional deployment group, in this case a Web-facing deployment. Such a deployment group is likely to provide a specific service and is self-contained. This is particularly useful for providing controlled processor capacity to a specific business line (such as Sales or Manufacturing) and their functional applications.

There are other circumstances in which you might want to control the allocation of processor capacity and yet gain the advantages of capacity redistribution using the Multiple Shared-Processor Pools capabilities. In Figure 2-20 on page 61 you can see a set of micro-partitions that are all database servers. You can see from the micro-partition definitions (left of the diagram) the entitled capacity of each micro-partition, but there is no Reserved Pool Capacity.

Also, the Entitled Pool Capacity equals the Maximum Pool Capacity (EPC = MPC). This essentially caps Shared-Processor Pool$_n$ and prohibits the micro-partitions within Shared-Processor Pool$_n$ from receiving any additional processor capacity from Level$_1$ capacity resolution.

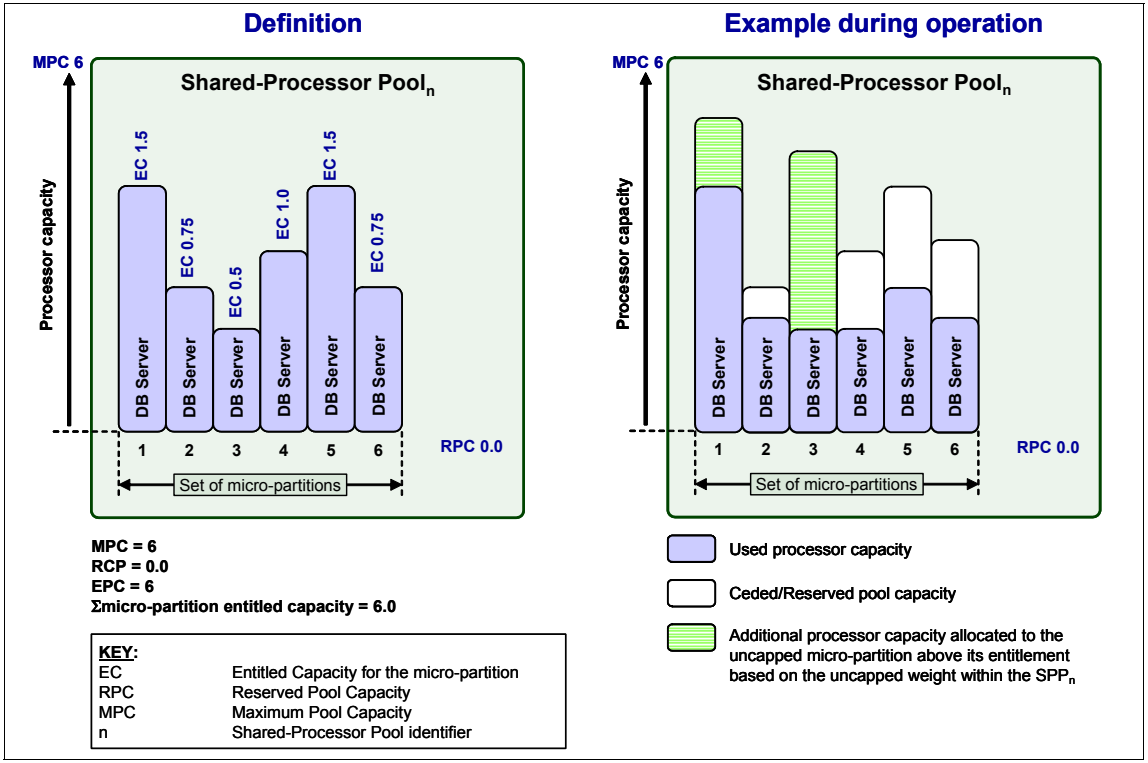


Figure 2-20 Capped Shared-Processor Pool offering database services

Such an arrangement restricts the processor capacity for Shared-Processor Pool$_n$ and therefore can restrict the software licensing liability, yet it provides the flexibility of processor capacity resolution within Shared-Processor Pool$_n$ (Level$_0$ capacity resolution). This optimizes the use of any software licensing because the maximum amount of work is done for the investment in the software license.

In the example in Figure 2-20, the Shared-Processor Pool$_n$ configuration limits the processor capacity of 6 processors, which provides the opportunity to maximize the workload throughput for the corresponding software investment.

You can, of course, change this definition to include a Reserved Pool Capacity. This additional guaranteed capacity will be distributed to the micro-partitions within Shared-Processor Pool$_n$ on an uncapped weighted basis (when a micro-partition requires the extra resources and has enough virtual processors to exploit it). For the example in Figure 2-20, to accommodate an increase in Reserved Pool Capacity you will also have to increase the Maximum Pool Capacity. In addition, for the increase in the processor capacity for

Shared-Processor Pool$_n$ there will probably be an increase in the software licensing costs.

If the Maximum Pool Capacity is increased further so that it is greater than the Entitled Pool Capacity, then uncapped micro-partitions within Shared-Processor Pool$_n$ may become eligible for Level$_1$ capacity resolution—additional processor capacity from elsewhere in the system. This may mean that any software for Shared-Processor Pool$_n$ would likely be licensed for the Maximum Pool Capacity whether or not the micro-partitions in Shared-Processor Pool$_n$ actually receive additional cycles above the Entitled Pool Capacity.

Figure 2-21 gives a simple example of a system with Multiple Shared-Processor Pools. One of the three Shared-Processor Pools is Shared-Processor Pool$_0$, the default Shared-Processor Pool.

As you can see, the Maximum Pool Capacity and the Entitled Pool Capacity for the DB_Services Shared-Processor Pool (Shared-Processor Pool$_2$) are equal, (MCP=EPC=6). Thus, DB_Services is effectively capped and cannot receive any additional processor capacity beyond that guaranteed to it by it's Entitled Pool Capacity. Consequently, it is only affected by Level$_0$ capacity resolution.

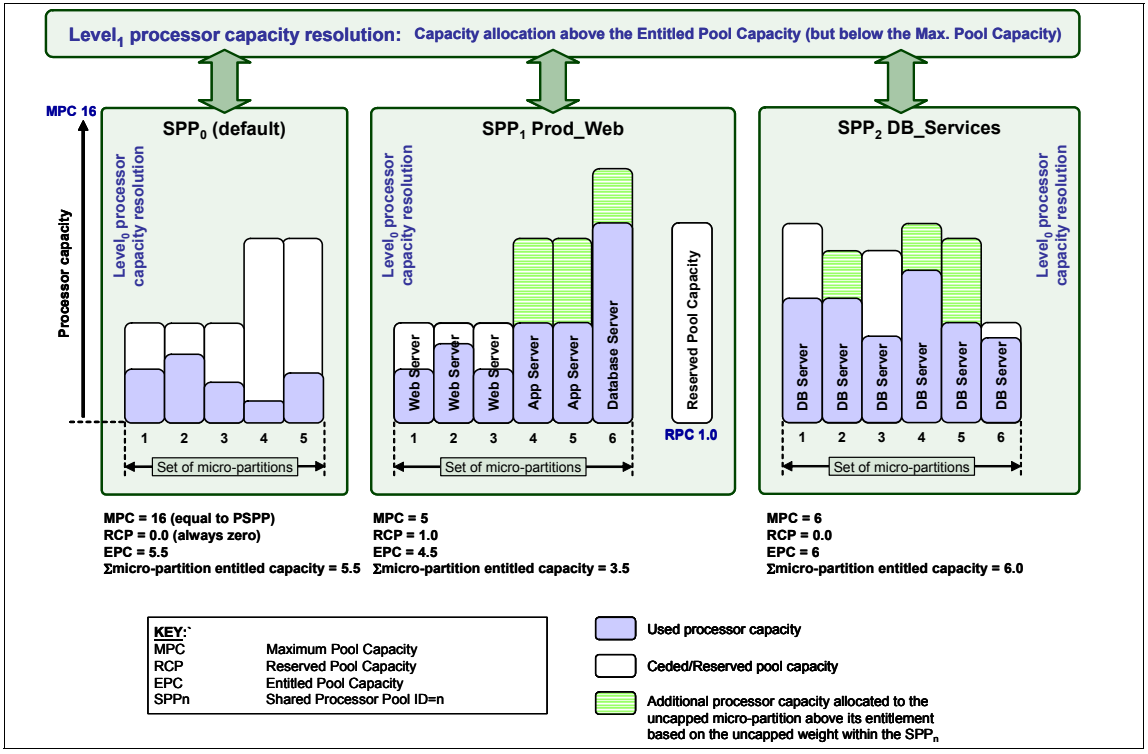


Figure 2-21 Example of a system with Multiple Shared-Processor Pools

However, the Prod_Web Shared-Processor Pool (SPP_1) has a Maximum Pool Capacity greater than its Entitled Pool Capacity and, therefore, the micro-partitions within Prod_Web may be allocated additional capacity using $Level_1$ capacity resolution.

Shared-Processor $Pool_0$ (default) and SPP_1 both participate in $Level_1$ capacity resolution under certain circumstances. As the attributes of SPP_0 are set to default values and cannot be altered by the systems administrator, it is always capable of consuming all the processor capacity in the physical shared-processor pool. Of course, this is assuming that at least one micro-partition in SPP_0 is uncapped and there are enough virtual processors to utilize the additional capacity.

2.3.5 Shared dedicated capacity

POWER6-based servers offer the capability of harvesting unused processor cycles from dedicated-processor partitions. These unused cycles are then donated to the physical shared-processor pool associated with Micro-Partitioning. This ensures the opportunity for maximum processor utilization throughout the system.

The system administrator can control which dedicated-processor partitions can donate unused cycles and the dedicated-processor partition must be identified as a donating partition. For more information, see 3.5.8, “Dedicated donating processors” on page 215.

2.4 Virtual I/O Server

The Virtual I/O Server is part of the IBM System p PowerVM Standard Edition or Enterprise Edition hardware feature. The Virtual I/O Server allows the sharing of physical resources between partitions to allow more efficient utilization. A possible use is for server consolidation.

Figure 2-22 on page 64 shows a very basic overview of a Virtual I/O Server configuration.

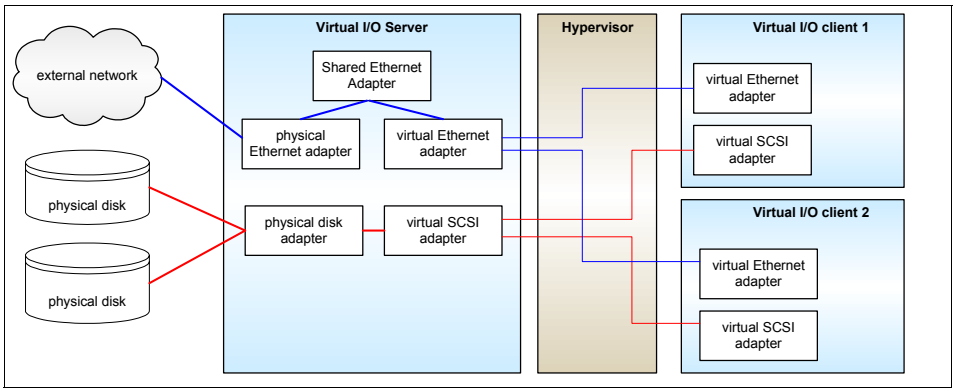


Figure 2-22 Simple Virtual I/O Server configuration

The list below shows the new functions which were added in Version 1.5 of the Virtual I/O Server:

- File-backed virtual SCSI devices
- Expansion pack with Kerberos, SNMPv3 and LDAP
- IVM support for physical I/O (POWER6 only)
- PowerVM Live Partition Mobility support
- Additional CLI statistics in svmon, vmstat fcstat and topas
- vSCSI eRAS
- Tivoli® Identity Manager (TIM) support
- Support for POWER6 blades

The following paragraphs describe the main functions and features of the Virtual I/O Server.

2.4.1 Supported platforms

The Virtual I/O Server can run on any POWER5 or POWER6-based System p server which has the PowerVM Standard Edition or Enterprise Edition feature enabled. Also supported are BladeCenter® JS21 and BladeCenter JS22 servers. Virtual I/O Servers can be deployed in pairs to provide continuous availability.

2.4.2 Storage virtualization

The Virtual I/O Server allows virtualization of physical storage resources. Virtualized storage devices are accessed by the client partitions through virtual

SCSI devices. All virtual SCSI devices are accessed as standard SCSI compliant LUNs by the client partition.

Virtualized storage devices can be backed by the following types of storage devices:

Internal physical disks Server internal disks such as SCSI disks located in I/O drawers.

External LUNs LUNs residing external storage subsystems from IBM as well as certain third party storage manufacturers. See “External storage subsystems” on page 65 for more detailed information about supported solutions.

Optical Storage Devices such as DVD-RAM, DVD-ROM and CD-ROM. Writing to a shared optical device is currently limited to DVD-RAM. DVD+RW and DVD-RW are not supported. A virtual optical device can only be assigned to one client partition at a time.

Additionally, the following logical storage devices can be used to back virtualized storage devices:

Logical volumes Internal disks as well as LUNs residing on external storage subsystems can be split into logical volumes on the Virtual I/O Server and then be exported to the client partitions.

Files A single file in a file residing in a local filesystem on the Virtual I/O Server. The file can for example be an ISO image.

External storage subsystems

A large number of IBM storage solutions are supported, including:

- DS8000™ series
- DS6000™ series
- DS4000™ series
- NSeries network attaches storage with Fibre Channel or iSCSI attach
- Enterprise Storage Server® (ESS)
- SAN Volume Controller (SVC)

Additionally, the Virtual I/O Server has been tested on selected configurations that include specific models from the following disk subsystem suppliers:

- EMC

- HDS
- HP
- Netapp

You can find a complete list of supported configurations on the IBM Internet at:

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

2.4.3 Shared Ethernet Adapter

The Virtual I/O Server allows shared access to external networks through the Shared Ethernet Adapter (SEA). The Shared Ethernet Adapter supports the following features:

Link Aggregation Bundling of several physical network adapters into one logical device using EtherChannel functionality.

SEA failover The SEA failover feature allows highly available configurations by using two Shared Ethernet Adapters running in two different Virtual I/O Servers.

TCP segmentation offload The SEA supports the large send and large receive features.

GVRP GVRP (GARP VLAN Registration Protocol) is a protocol that facilitates control of VLANs within larger networks. It helps to maintain VLAN configurations dynamically based on network adapter configurations.

2.4.4 Network security

The Virtual I/O Server supports OpenSSH for secure remote logins. It also provides a firewall for limiting access by ports, network services and IP addresses.

With Virtual I/O Server 1.5, an expansion pack is provided that delivers additional security functions. These include:

SNMP v3 SNMPv3 provides secure access by a combination of authenticating and encrypting packets over the network.

Kerberos	Kerberos is a system that provides a central authentication mechanism for a variety of client/server applications using passwords and secret keys.
LDAP	LDAP is a directory service that can be used for centralized user management.

2.4.5 Command line interface

The Virtual I/O Server provides a command line interface to perform management tasks such as:

- Management of mappings between physical and virtual resources
- Gathering and displaying utilization data of resources through commands like topas, vmstat, iostat or viostat for performance management
- Troubleshooting of physical and virtual resources using the hardware error log
- Updating the Virtual I/O Server
- Securing the Virtual I/O Server by configuring user security and firewall policies

Note: Some tasks can also be performed through SMIT style menus using the **`cfgassist`** command.

2.4.6 Hardware Management Console integration

The Hardware Management Console (HMC) provides functions to simplify the handling of the Virtual I/O Server environment. For example, an overview of the Virtual Ethernet and Virtual SCSI topologies is available. It is also possible to execute Virtual I/O Server commands from the HMC.

2.4.7 System Planning Tool support

Using the System Planning Tool, complex Virtual I/O Server configurations can be planned and deployed including:

- Virtual SCSI adapter configuration
- Virtual Ethernet adapter configuration
- Shared Ethernet Adapter configuration with failover and EtherChannel

More information about how to use the System Planning Tool can be found in 3.6.3, “Simplification by using system plans and System Planning Tool” on page 222.

2.4.8 Performance Toolbox support

Included with the Virtual I/O Server is a Performance Toolbox (PTX®) agent which extracts performance data. This data can be viewed through an X Windows® GUI if you have licensed the AIX Performance Toolbox.

2.4.9 Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) is used to manage selected POWER5 or POWER6-based System servers using a Web-based graphical interface without requiring an HMC.

This reduces the hardware needed for the adoption of virtualization technology, particularly for low-end systems. This solution fits in small and functionally simple environments where only few servers are deployed or some advanced HMC functions are required.

The Integrated Virtualization Manager (IVM) is a basic hardware management solution, included in the VIO software that inherits key Hardware Management Console (HMC) features. For further information on IVM, see 2.5, “Integrated Virtualization Manager” on page 70 or the Redpaper *Integrated Virtualization manager on IBM System p5*, REDP-4061.

2.4.10 Tivoli support

Included with the Virtual I/O Server are a number of preinstalled Tivoli agents which allow easy integration into an existing Tivoli Systems Management infrastructure.

IBM Tivoli Monitoring System Edition for System p (ITMSE)

ITMSE enables you to monitor the health and availability of the Virtual I/O server. It also provides graphical views of the virtualization environment. You will need a server running the Tivoli Enterprise Management Server and Tivoli Enterprise Portal in order to view the information collected by the agent.

IBM Tivoli Monitoring System Edition for System p can be downloaded for no cost for one year at:

http://www-306.ibm.com/software/tivoli/products/monitor-systemp/

More information about ITMSE can be found in A*dvanced POWER Virtualization: Managing and Monitoring*, SG24-7590.

Tivoli Storage Manager (TSM) client

The Tivoli Storage Manager (TSM) client can be used to back up Virtual I/O Server configuration data to a TSM server.

More information about TSM can be found at:

http://www-306.ibm.com/software/tivoli/products/storage-mgr/

IBM Tivoli Application Dependency Discovery Manager

IBM Tivoli Application Dependency Discovery Manager (TADDM) provides deep-dive discovery for System p servers including their dependencies on the network and applications along with its configuration data, subsystems, and virtualized LPARs. TADDM is currently capable of recognizing a Virtual I/O Server and the software level it is running.

More information about TADDM can be found at:

http://www-306.ibm.com/software/tivoli/products/taddm/

IBM Tivoli Usage and Accounting Management agent

The IBM Tivoli Usage and Accounting Management (ITUAM) agent can be used to collect accounting and usage data of Virtual I/O Sever resources so that they can be fed into ITUAM where they can be analyzed and used for reporting and billing.

More information about ITUAM can be found at:

http://www-306.ibm.com/software/tivoli/products/usage-accounting/

Tivoli Identity Manager

Tivoli Identity Manager (TIM) provides a secure, automated and policy-based user management solution which can be used the manage Virtual I/O Server users.

More information about TIM can be found at:

http://www-306.ibm.com/software/tivoli/products/identity-mgr/

2.5 Integrated Virtualization Manager

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

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

IBM has developed the IVM, a hardware management solution that performs a subset of the HMC features for a single server, avoiding the need for a dedicated HMC server. IVM manages standalone servers so a second server managed by IVM would have its own instance of the IVM. With the subset of HMC server functionality, IVM provides a solution that enables the administrator to quickly set up a system. The IVM is integrated within the Virtual I/O Server product, which services I/O and processor virtualization in IBM System p systems.

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

The HMC is a centralized point of hardware control. A single HMC can handle multiple POWER5 and/or POWER6 systems, and two HMCs may manage the same set of servers in a dual-active configuration providing resilience.

Hardware management is done using the HMC interface (Web browser-based starting with HMC version 7), which communicates to the servers using a standard Ethernet connection to the service processor of each POWER5 or POWER6 system. Interacting with the service processor, the HMC is able to create, manage, and modify logical partitions, modify the hardware configuration of the managed system, and manage service calls.

2.5.1 IVM setup guidelines

To manage a system using IVM, some implicit rules apply to the server configuration and setup. The following guidelines are designed to assist you:

- The system is configured in the *Factory Default* mode, which means that a single partition with service authority predefined owns all the hardware resources. If the system is not configured in Factory Default mode because it is already partitioned or attached to an HMC, you can reset the system to the Factory Default mode using the Advanced System Management Interface (ASMI) for the service processor.
- The predefined partition is started automatically at system power on. A terminal console will be required for the initial IVM install and configuration of

an IP address, before the IVM GUI can be used. A null-modem serial cable and a computer running Windows (HyperTerm) or Linux (minicom) can be used for this.

- The PowerVM Standard Edition or Enterprise Edition feature has to be enabled. When ordering one of the features with the system, it should be enabled by default; otherwise, it can be enabled using the ASMI.
- Virtual I/O Server Version 1.2 or higher has to be installed on the predefined partition.

The Virtual I/O Server then automatically allocates all I/O resources. All other LPARs are configured using the built-in IVM on the Virtual I/O Server. Starting with IVM 1.5.1.1 and supported POWER6 systems, you can assign physical adapters to LPARs, otherwise all physical resources are owned by the IVM partition.

The configuration can be done using the IVM GUI or by using the command line interface on the IVM Server. The administrator can use a Web browser to connect to IVM to set up the system configuration.

Figure 2-23 shows a sample configuration using IVM on a POWER6 machine with dedicated physical and virtual adapters configured.

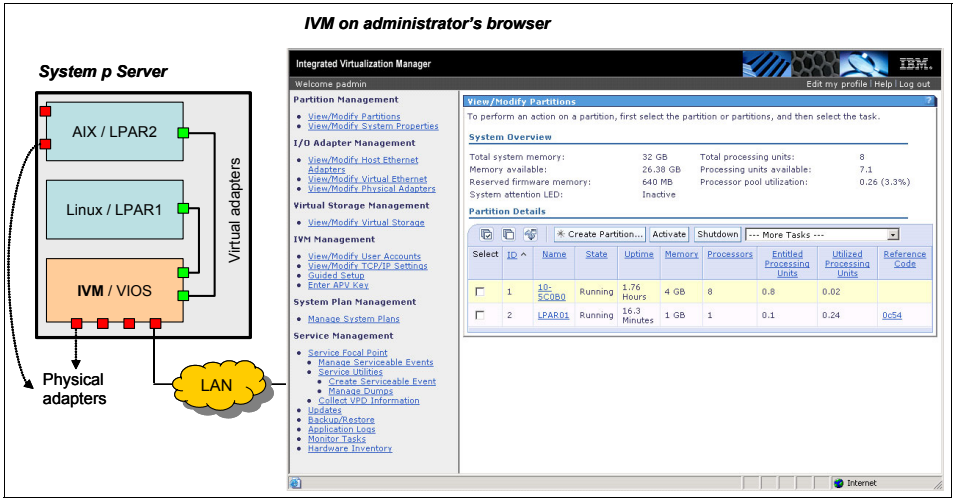


Figure 2-23 Integrated Virtualization Manager configuration on a POWER6 server

The tight relationship between the Virtual I/O Server and IVM enables the administrator to manage a partitioned system without the HMC. The software that is normally running on the HMC has been reworked to fit inside the Virtual I/O Server, selecting the subset of functions required by the IVM configuration

model. Since IVM is running using system resources, the design has been developed to have minimal impact on their consumption.

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

The VMC device allows IVM to provide basic logical partitioning functions:

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

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

LPAR management with IVM is accomplished through a Web interface developed to make administration tasks easier and quicker for those unfamiliar with the full HMC solution. It is important to recognize, though, that the HMC can manage more than one server simultaneously and offers some advanced features not present in IVM. Being integrated within the Virtual I/O Server code, IVM also handles all virtualization tasks that normally require Virtual I/O Server commands to be run.

IVM manages the system in a similar way to the HMC, but using a different interface. A new System p administrator will quickly learn the required skills, while an HMC expert should study the differences before using IVM.

2.5.2 Partition configuration with IVM

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

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

LPARs in an IVM-managed system are isolated exactly as in all POWER5 and POWER6 systems and cannot interact except using the virtual (and now physical) devices. Only IVM has been enabled to perform limited actions on the other LPARs, such as:

- Power on and power off
- Shut down the operating system gracefully
- Create and delete LPARs
- View and change configuration of LPARs

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

Processors

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

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

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

Virtual Ethernet

The IVM managed system is configured with four predefined virtual Ethernet networks, each having a virtual Ethernet ID ranging from 1 to 4. Starting with IVM Version 1.5, you can now add additional VLANs and adapters to LPARs across the entire 802.1Q supported range of 1-4096 (using IVM line-mode commands).

Prior to Version 1.5, every LPAR can have up to two virtual Ethernet adapters that can be connected to any of the four virtual networks in the system.

Each virtual Ethernet network can be bridged by Virtual I/O Server to a physical network using only one physical adapter or, if POWER6 systems, logical HEA adapter. The same physical adapter cannot bridge more than one virtual network.

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

Virtual storage

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

Virtual optical device

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

Virtual TTY

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

Integrated Virtual Ethernet

For supported POWER6 systems, the IVM allows the configuration of Integrated Virtual Ethernet and the assignment of HEA adapters to LPARs.

The wizard allows the user to configure speed and duplex of the physical HEA ports and will present the user with a simple checkbox to allow the configuration of one logical HEA adapter per physical HEA port when defining LPARs.

Values such as MCS are set to their minimum values; for more information, see *Integrated Virtual Ethernet Adapter*, REDP-4340.

Physical adapters

Starting with IVM 1.5 on selected POWER6 systems you can assign dedicated I/O adapters to partitions other than the IVM LPAR. Using this functionality you can have IVM-controlled LPARs with dedicated Ethernet or disk adapters in addition to the virtual resources described above.

Note: At the time of writing, Multiple Shared Processor Pools are not supported on IVM-managed POWER6 systems.

2.6 Virtual SCSI introduction

Virtual SCSI is used to refer to a virtualized implementation of the SCSI protocol. Virtual SCSI requires POWER5 or POWER6 hardware with the PowerVM Standard or Enterprise edition feature activated. It provides virtual SCSI support for AIX Version 5.3 and 6.1 and supported versions of Linux.

Considering the two most popular ways of provisioning storage to servers:

- Integrated disks

  With integrated server disks growing ever larger requiring fewer disks for a given amount of storage, a significant cost can be associated with the adapters and the attachment of these disks to servers. With such large disks it is also more difficult to utilize all the available space.

- External storage subsystems, for example SAN disks or NAS disks

  Again with the introduction of ever larger and cheaper disks driving down the costs per gigabyte of storage, this leaves the costs of adapters (and additionally any switches and cabling) as a significant investment if a number of servers are involved.

For systems such as development or test servers, where there are much lower storage bandwidth requirements, it would be beneficial to combine the storage requirements through a single adapter to better utilize the available bandwidth. Taking into account the cost savings of not only the server adapters, but also:

- Switches and switch ports (SAN or Etherchannel)
- Purchase of cables
- Installation of cables and patching panels

Very quickly the cost benefits for virtualizing storage can be realized, and that is before considering any additional benefits from the simplification of processes or organization in the business.

At the time of writing, PowerVM on the IBM System p platform supports up to 10 partitions per processor, up to a maximum of 254 partitions per server. The server hardware, depending on the machine model and number of attached I/O drawers, can provide up to 240 I/O slots and 192 internal SCSI disks per machine. With each partition typically requiring one I/O slot for disk attachment and a second Ethernet attachment, it is possible to build approximately 120 dedicated I/O partitions with a single server.

254 partitions require 508 slots to provision one network adapter and one disk adapter per partition, and that is before any resilience or adapter redundancy is considered. To overcome these physical requirements, I/O resources can be

shared. Virtual SCSI, provided via the Virtual I/O Server, provides the means to do this.

Most customers are deploying a pair of Virtual I/O Servers per physical server and using multi-path I/O or LVM mirroring technology to provide resilient access to storage. This configuration provides continuous availability to the disk resources, even if there is a requirement to perform maintenance on the Virtual I/O Servers.

Note: You will see different terms in this book that refer to the various components involved with virtual SCSI. Depending on the context, these terms may vary. With SCSI, usually the terms *initiator* and *target* are used, so you may see terms such as *virtual SCSI initiator* and *virtual SCSI target*. On the HMC or IVM, the terms *virtual SCSI server adapter* and *virtual SCSI client adapter* are used, which are synonymous.

2.6.1 Partition access to virtual SCSI devices

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

Virtual SCSI client and server architecture overview

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

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

Physical disks presented to the Virtual I/O Server can be exported and assigned to a client partition in a number of different ways:

- The entire disk is presented to the client partition.
- The disk is divided into several logical volumes, which can be presented to a single client or multiple different clients.
- With the introduction of Virtual I/O Server 1.5, files can be created on these disks and file-backed storage can be created.

The logical volumes or files can be assigned to different partitions. Therefore, virtual SCSI enables sharing of adapters as well as disk devices.

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

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

The client partition accesses its assigned disks through a virtual SCSI client adapter. The virtual SCSI client adapter sees the disks, logical volumes or file-backed storage through this virtual adapter as virtual SCSI disk devices. The commands in the following example show how the disks appear on an AIX client partition:

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

# lscfg -vpl hdisk2
hdisk2 U9117.MMA.100F6A0-V4-C40-T1-L810000000000 Virtual SCSI Disk
Drive
```

The vhost SCSI adapter is the same as a normal SCSI adapter. You can have multiple disks available from it. Usually one virtual SCSI server mapped to one virtual SCSI client will be configured, mapping backing devices through to individual LPARs. It is possible to map these virtual SCSI servers to multiple LPARs, which is useful for creating virtual optical devices, allowing CD or DVD media devices to be moved amongst LPARs using simple AIX commands.

The mapping of Virtual I/O Server virtual SCSI adapters to virtual SCSI client adapters is performed on the HMC or the IVM. See 3.6.1, “Virtual SCSI resources” on page 218 for more details.

Figure 2-24 shows an example where one physical disk is divided into two logical volumes by the Virtual I/O Server. Each of the two client partitions is assigned one logical volume, which is then accessed through a virtual I/O adapter (VSCSI Client Adapter). Inside the partition, the disk is seen as a normal hdisk.

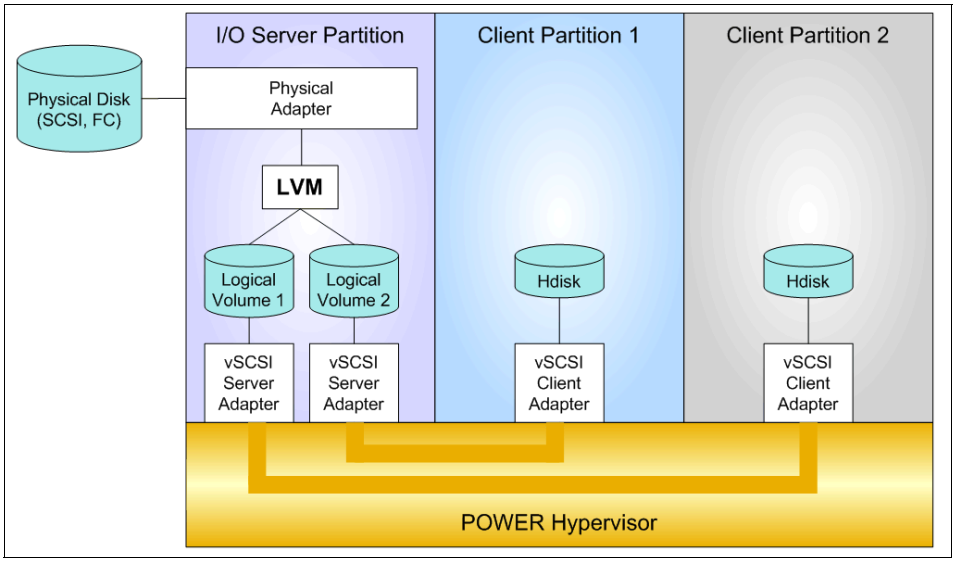


Figure 2-24 Virtual SCSI architecture overview

Multiple backing devices per virtual SCSI link

Once these virtual SCSI server/client links have been set up, one or more backing devices (whole disks, logical volumes or files) can be presented using the same link.

At the time of writing each link can handle 510 I/O requests in parallel (also referred to as the queue depth), meaning multiple backing devices can be assigned to each link; see Figure 2-25 on page 79.

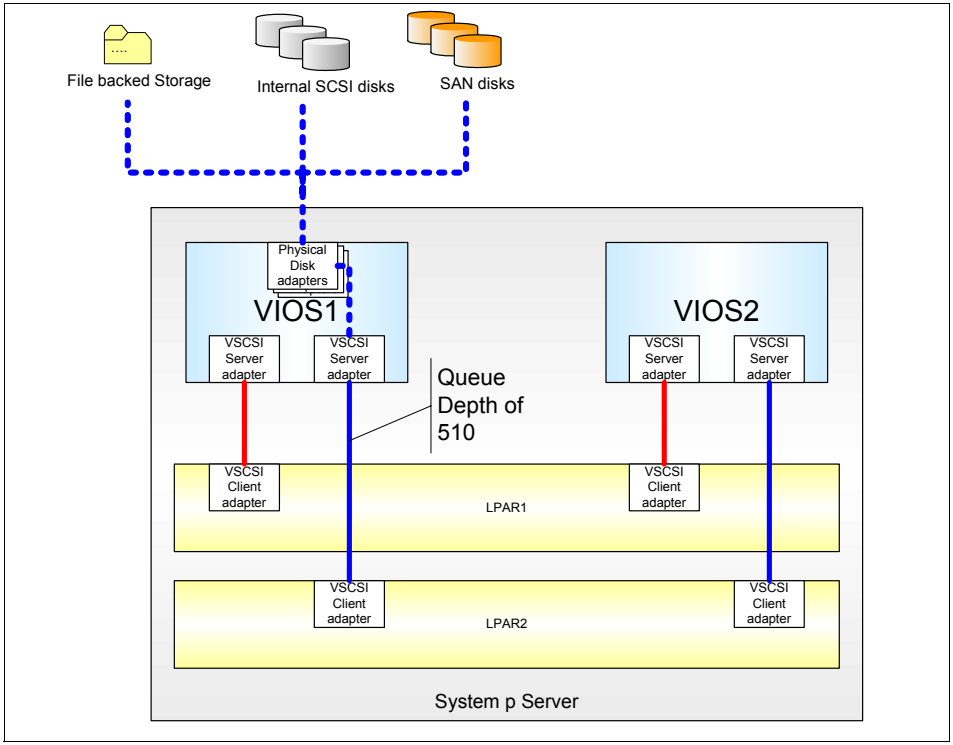


Figure 2-25 Queue depths and virtual SCSI considerations

Each device will have an associated queue depth, which is the number of I/O requests each device can handle concurrently. Table 2-5 shows some default values and some suggested maximum number of devices to present through a single virtual SCSI server/client linkage.

Note: If you have disks with high I/O requirements or have tuned the queue depths, make sure you take these into consideration. The file-backed devices will inherit the queue depth of their backing device.

Table 2-5 Suggested maximum number of devices per virtual SCSI link

Disk Type	Default Queue Depth	Suggested max per virtual SCSI linkage
Internal SCSI disk	3	85
SAN disk	10	26

SCSI Remote Direct Memory Access

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

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

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

Figure 2-26 demonstrates data transfer using LRDMA. The VSCSI initiator of the client partition uses the POWER Hypervisor to request data access from the VSCSI target device.

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

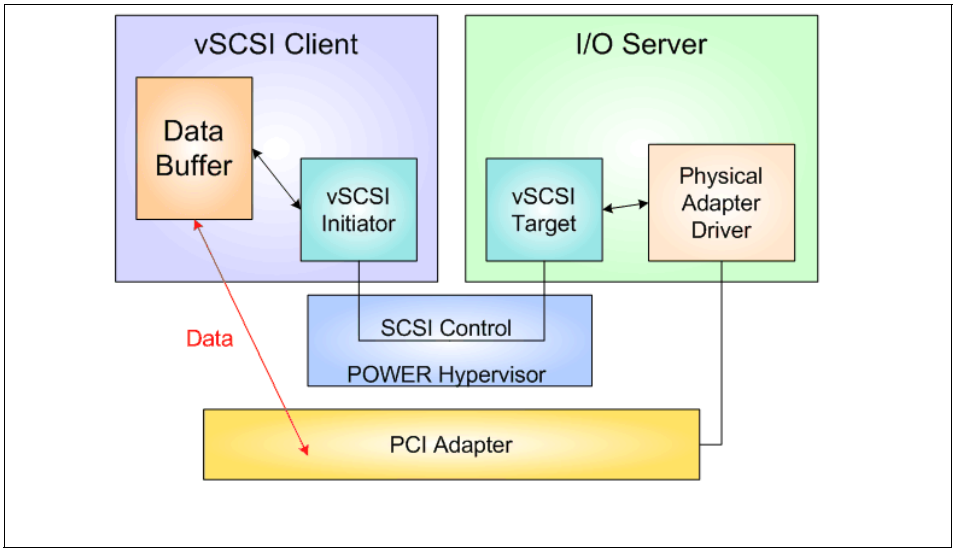


Figure 2-26 Logical Remote Direct Memory Access

Dynamic partitioning for virtual SCSI devices

Virtual SCSI server and client devices can be assigned and removed dynamically using the HMC or IVM. This should be coordinated with the linking and removal of the backing devices in the Virtual I/O Servers.

With a physical device mapped to a virtual server (vhost) adapter, there are two options for moving disks between partitions:

- Reassign the VSCSI adapter on the server partition to a new client partition, then create a VSCSI client adapter in the new target partition. On the client, the **cfgmgr** command is run and the new VSCSI disk is available.
- If the virtual SCSI server and client adapters are already created for the new LPAR, just remove and recreate the linkage (vtscsi) device on the VIO server and then run the **cfgmgr** command on the target LPAR.

Virtual SCSI optical devices

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

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

For more information, see 3.5.3, “Virtual SCSI optical devices” on page 190.

2.6.2 General virtual SCSI considerations

The following areas should be considered when implementing virtual SCSI:

- At the time of writing, virtual SCSI supports Fibre Channel, parallel SCSI, SCSI RAID devices, and optical devices, including DVD-RAM and DVD-ROM. Other protocols, such as SSA and tape devices, are not supported.
- A logical volume on the Virtual I/O Server used as a virtual SCSI disk cannot exceed 1 TB in size.
- The SCSI protocol defines mandatory and optional commands. While virtual SCSI supports all the mandatory commands, not all optional commands are supported.

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

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

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

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

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

Only one disk should appear in the resulting list.

Installation and migration considerations

The following are the major installation and migration considerations:

- Planning client partition root volume group sizes prior to creating logical volumes is recommended. Increasing a rootvg by extending its associated Virtual I/O Server logical volume is not supported.
- Migrating a disk from physically connected SCSI to a virtually connected SCSI device is not supported at this time. At the time of writing the only tested method for migration involved a backup and restore of the data when migrating from a physical device to a virtual device.
- Virtual SCSI itself does not have any maximums in terms of number of supported devices or adapters. At the time of writing, the Virtual I/O Server supports a maximum of 1024 virtual I/O slots per Virtual I/O Server. A maximum of 256 virtual I/O slots can be assigned to a single client partition.

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

Performance considerations

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

Virtual Ethernet, having non-persistent traffic, runs at a higher priority than the virtual SCSI on the VIO server. To make sure that high volumes of networking traffic will not starve the virtual SCSI of CPU cycles, a threaded mode of operation has been implemented for the Virtual I/O Server by default since version 1.2.

2.7 Virtual and Shared Ethernet introduction

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

Virtual Ethernet requires the following:

- An IBM System p server (POWER5 or POWER6)
- The appropriate level of AIX (5L v5.3 or v6.1) or Linux
- Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) to define the virtual Ethernet devices

Virtual Ethernet does not require the purchase of any additional features or software. However, Shared Ethernet Adapters and the Virtual I/O Server software does require the PowerVM feature (Standard or Enterprise editions).

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

- Virtual LAN: a general overview of Virtual LAN concepts and its use with AIX is given.
- Inter-partition networking: using virtual Ethernet
- Sharing of physical Ethernet adapters: on IBM System p to allow multiple partitions to access external networks
- An Example: Virtual LAN and Shared Ethernet Adapter

2.7.1 Virtual LAN

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

Virtual LAN overview

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

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

In Figure 2-27 on page 85, two VLANs (VLAN 1 and 2) are defined on three switches (Switch A, B, and C). There are seven hosts (A-1, A-2, B-1, B-2, B-3, C-1, and C-2) connected to the three switches. The physical network topology of the LAN forms a tree, which is typical for a non-redundant LAN:

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

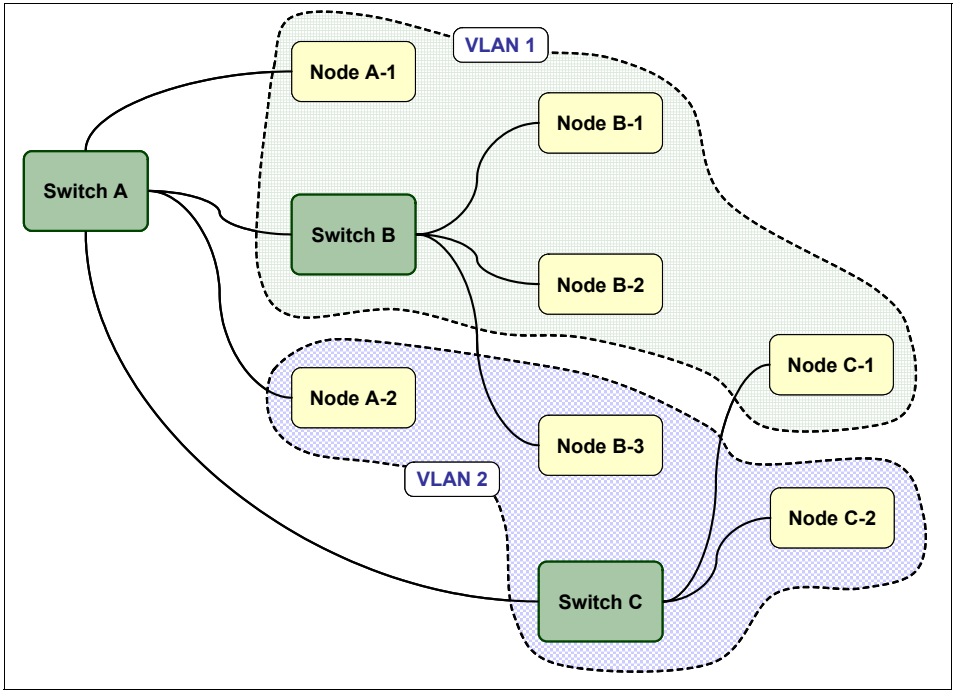


Figure 2-27 Example of VLANs

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

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

Although nodes C-1 and C-2 are physically connected to the same switch C, traffic between two nodes is blocked. To enable communication between VLAN 1 and 2, L3 routing or inter-VLAN bridging would have to be established between

them; this would typically be provided by an L3 device, for example, a router or firewall plugged into switch A.

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

Virtual LAN benefits

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

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

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

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

AIX virtual LAN support

Some of the technologies for implementing VLANs include:

- Port-based VLAN
- Layer-2 VLAN

- Policy-based VLAN
- IEEE 802.1Q VLAN

Note: VLAN support in AIX and PowerVM is based on the IEEE 802.1Q VLAN implementation.

Port-based VLAN can also be used with AIX and is completely transparent to AIX. VLAN support is not specific to PowerVM, it is available on all IBM System p servers with the appropriate level of AIX.

The IEEE 802.1Q VLAN support is achieved by letting the AIX VLAN device driver add a VLAN ID tag to every Ethernet frame, as shown in Figure 2-28, and the Ethernet switches restricting the frames to ports that are authorized to receive frames with that VLAN ID.

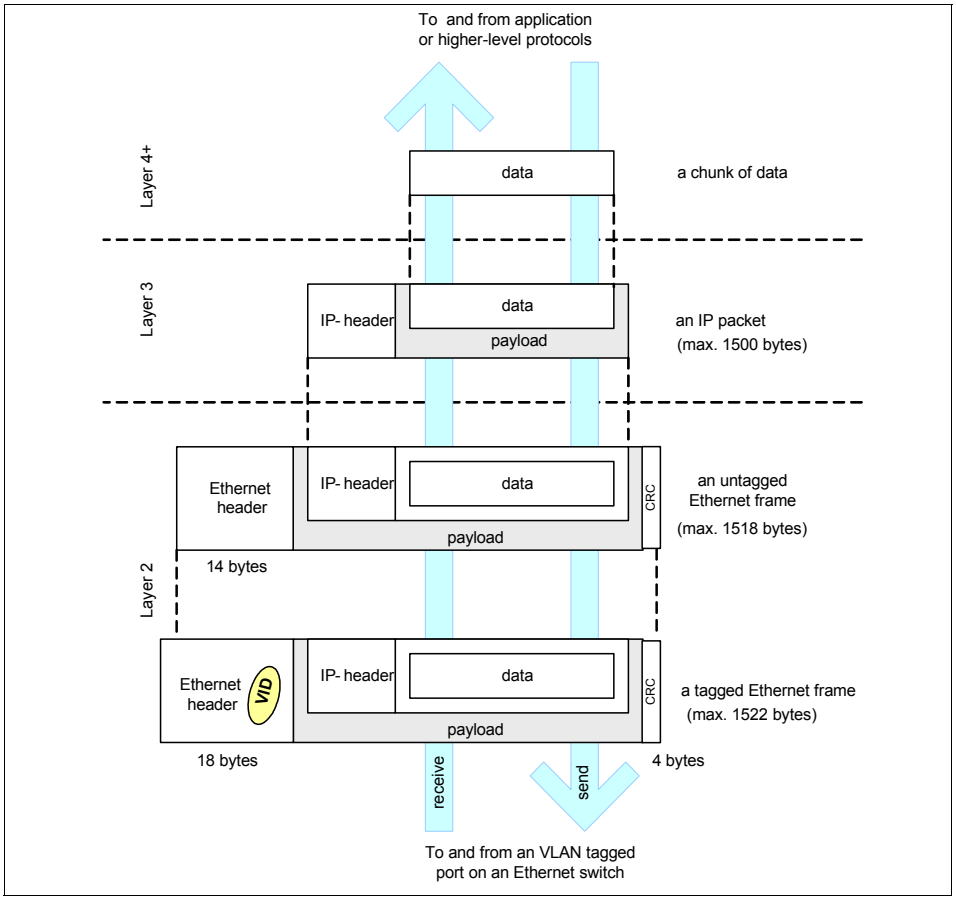


Figure 2-28 The VID is placed in the extended Ethernet header

The VLAN ID is placed in the Ethernet header and consequently does not create an additional header. To be able to do this, the Ethernet frame size for tagged frames was increased from 1518 bytes to 1522 bytes and the Ethernet header format was slightly modified with the introduction of IEEE802.1Q. Thus, in contrast to, for example, Point-to-Point-Protocol-over-Ethernet (PPPoE), which is commonly used for xDSL with a MTU of 1492, you do not have to care about reducing the TCP/IP's MTU of 1500 with respect to VLAN ID tagging.

Notes:

1. You do not have to reduce the TCP/IP default MTU size of 1500 for Ethernet due to the additional 4 bytes introduced by IEEE 802.1Q VLANs.
2. If you increase the TCP/IP's MTU size for virtual Ethernet adapters that are implemented by the POWER Hypervisor, as introduced in 2.7.2, "Inter-partition networking with virtual Ethernet" on page 91, you must take the additional 4 bytes introduced by IEEE 802.1Q VLANs into account: the maximum MTU is 65394 without VLANs and 65390 bytes with VLANs. This is due to a limit of 65408 bytes for virtual Ethernet frames transmitted through the POWER Hypervisor. (The Ethernet headers are 14 and 18 bytes, respectively, but there is no need for the 4-byte CRC in the POWER Hypervisor).

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

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

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

Receiving packets on a tagged port

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

1. Tagged port receives an untagged packet: the packet will be tagged with the PVID, then forwarded.

2. Tagged port receives a packet tagged with the PVID or one of the assigned VIDs: the packet will be forwarded without modification.
3. Tagged port receives a packet tagged with any VLAN ID other than the PVID or assigned additional VIDs: the packet will be discarded.

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

Receiving packets on an untagged port

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

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

1. Untagged port receives an untagged packet: the packet is tagged with the PVID, then forwarded
2. Untagged port receives a packet tagged with the PVID: the packet is forwarded without modification
3. Untagged port receives a packet tagged with any VLAN ID other than the PVID: the packet is discarded

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

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

Sending packets on a tagged or untagged port

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

- Tagged port sends out a packet: the PVID or VID remains tagged to the packet.
- Untagged port sends out a packet: the PVID is stripped from the packet.

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

Ethernet adapters and interfaces in AIX

AIX differentiates between a network adapter and network interface:

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

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

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

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

When using VLAN, EtherChannel (EC), Link Aggregation, (LA) or Network Interface Backup (NIB) with AIX, the general concept is that Ethernet adapters are being associated with other Ethernet adapters, as shown in Figure 2-29 on

page 91. EtherChannel and Link Aggregation will be explained in more detail in 4.3.4, “Using Link Aggregation on the Virtual I/O Server” on page 258.

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

If you have one physical Ethernet adapter with device name ent0, which is connected to a tagged switch port with PVID=1 and VID=100, the administrator will generate an additional device name ent1 for the VLAN with VID=100. The original device name ent0 will represent the untagged Port VLAN with PVID=1. Ethernet interfaces can be put on both adapters: en0 would be stacked on ent0 and en1 on ent1, and different IP addresses will be configured to en0 and en1. This is shown in Figure 2-29.

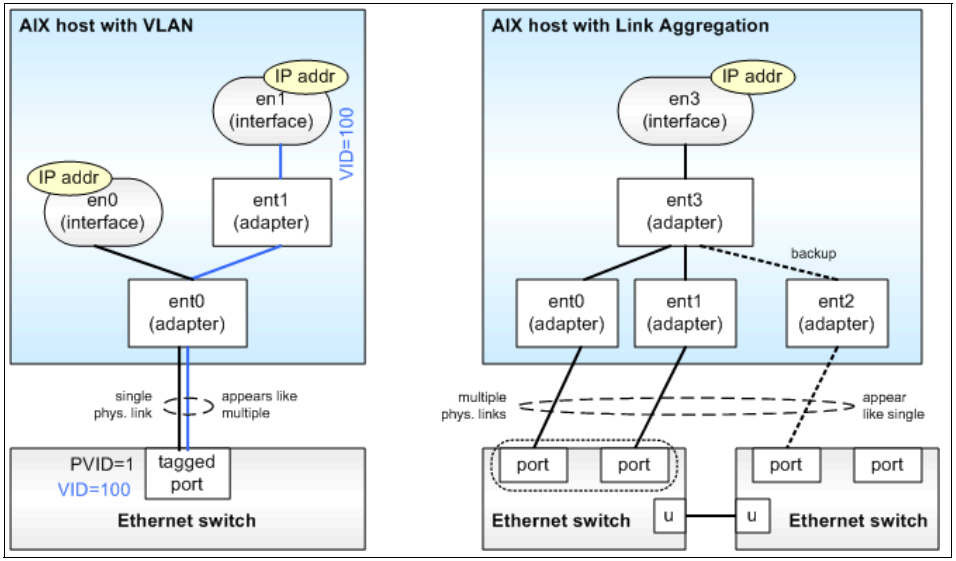


Figure 2-29 adapters and interfaces with VLANs (left) and LA (right)

2.7.2 Inter-partition networking with virtual Ethernet

The POWER Hypervisor implements an IEEE 802.1Q VLAN style virtual Ethernet switch. Similar to a physical IEEE 802.1Q Ethernet switch it can support tagged and untagged ports. A virtual switch does not really need ports, so the

virtual ports correspond directly to virtual Ethernet adapters that can be assigned to partitions from the HMC or IVM. There is no need to explicitly attach a virtual Ethernet adapter to a virtual Ethernet switch port. To draw on the analogy of physical Ethernet switches, a virtual Ethernet switch port is configured when you configure the virtual Ethernet adapter on the HMC or IVM.

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

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

But it cannot be used for EtherChannel or Link Aggregation.

The POWER Hypervisor's virtual Ethernet switch can support virtual Ethernet frames of up to 65408 bytes size, which is much larger than what physical switches support: 1522 bytes is standard and 9000 bytes are supported with Gigabit Ethernet Jumbo Frames. Thus, with the POWER Hypervisor's virtual Ethernet, you can increase TCP/IP's MTU size to 65394 (= 65408 - 14 for the header, no CRC) in the non-VLAN-case and to 65390 (= 65408 - 14 - 4 for the VLAN, again no CRC) if you use VLAN.

Increasing the MTU size could benefit performance because it may improve the efficiency of the transport. This is dependant on the communication data requirements of the running workload.

2.7.3 Sharing physical Ethernet adapters

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

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

Routing

By enabling the IP forwarding capabilities of an AIX or Linux partition with virtual and physical Ethernet adapters, the partition can act as router. Figure 2-30 on page 93 shows a sample configuration. The client partitions have their default routes set to the partition. This routes the traffic to the external network.

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

The routing approach has the following characteristics:

- It does not require the purchase of either PowerVM features and use of a Virtual I/O Server.
- IP filtering, firewalling, or Quality of Service (QoS) could be implemented on these routing partitions.
- The routing partitions could also act as endpoints for IPsec tunnels, thus providing for encrypted communication over external networks for all partitions, without having to configure IPSec on all partitions.
- Continuous availability can be enhanced by implementing more than one such routing partition and by configuring IP multipathing on the clients, or by implementing IP address fail over on routing partitions. This is discussed in 4.3, “Shared Ethernet redundancy” on page 251.

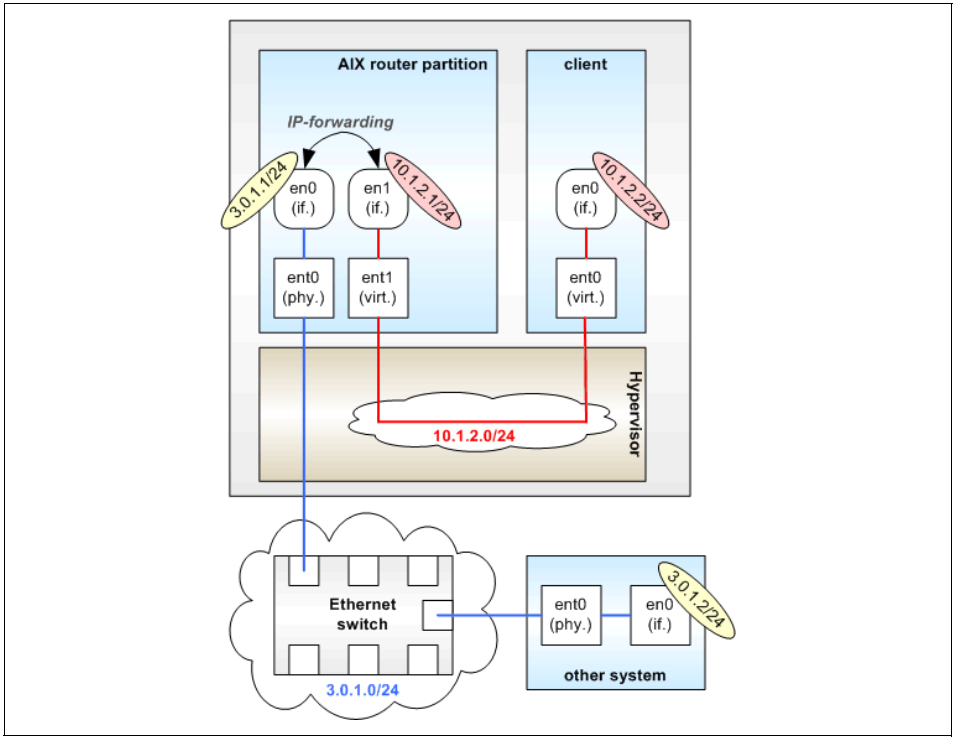


Figure 2-30 Connection to external network using routing

Shared Ethernet Adapter

A Shared Ethernet Adapter (SEA) can be used to connect a physical Ethernet network to a virtual Ethernet network. It also provides the ability for several client partitions to share one physical adapter. Using a SEA, you can connect internal

and external VLANs using a physical adapter. The SEA hosted in the Virtual I/O Server acts as a layer-2 bridge between the internal and external network.

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

Tip: A Linux partition can provide bridging function as well, with the **brctl** command.

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

- Virtual Ethernet requires the POWER Hypervisor and PowerVM feature (Standard or Enterprise Edition) and the installation of a Virtual I/O Server.
- Virtual Ethernet cannot be used prior to AIX 5L Version 5.3. Thus, an AIX 5L Version 5.2 partition will need a physical Ethernet adapter.

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

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

In order to bridge network traffic between the virtual Ethernet and external networks, the Virtual I/O Server has to be configured with at least one physical Ethernet adapter. One SEA can be shared by multiple virtual Ethernet adapters and each can support multiple VLANs. Figure 2-31 on page 95 shows a configuration example of an SEA with one physical and two virtual Ethernet adapters. An SEA can include up to 16 virtual Ethernet adapters on the Virtual I/O Server that share the physical access.

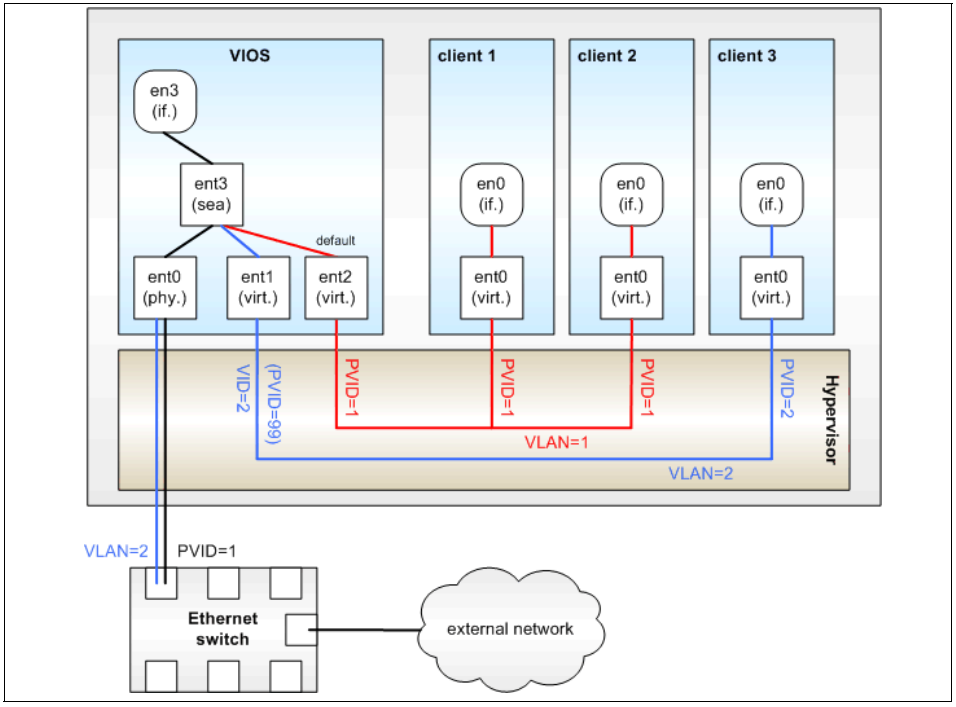


Figure 2-31 Shared Ethernet Adapter

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

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

The SEA directs packets based on the VLAN ID tags. One of the virtual adapters in the Shared Ethernet Adapter on the Virtual I/O Server must be designated as the *default* PVID adapter. Ethernet frames without any VLAN ID tags that the SEA receives from the external network are forwarded to this adapter and assigned the default PVID. In Figure 2-31 on page 95, ent2 is designated as the default adapter, so all untagged frames received by ent0 from the external network will be forwarded to ent2. Since ent1 is not the default PVID adapter, only VID=2 will be used on this adapter, and the PVID=99 of ent1 is not important. It could be set to any unused VLAN ID. Alternatively, ent1 and ent2 could also be merged into a single virtual adapter ent1 with PVID=1 and VID=2, being flagged as the default adapter.

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

Theoretically, one adapter can act as the only contact with external networks for all client partitions. For more demanding network traffic scenarios (large number of client partitions or heavy network usage), it is important to adjust the throughput capabilities of the physical Ethernet configuration to accommodate the demand.

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

- Using Link Aggregation (EtherChannel), several physical network adapters can be aggregated. Refer to 4.3.4, “Using Link Aggregation on the Virtual I/O Server” on page 258 for more details.
- Using several Shared Ethernet Adapters provides more queues and more performance.

Other aspects that have to be taken into consideration are availability (refer to 4.3, “Shared Ethernet redundancy” on page 251) and the ability to connect to different networks.

When to use routing or bridging

Where several servers are consolidated onto a single system or where LPARs are moved to another server, bridging is often the preferred choice. This is because the network topology does not have to be changed and IP subnets and IP addresses of the consolidated servers can stay unmodified. Even an existing multiple VLAN scheme can be bridged.

Routing may be worth consideration under various circumstances:

- Additional functions to basic packet forwarding are required and should be performed in a central place:
  - IP filtering
  - Firewalling
  - QoS Routing
  - IPsec tunneling
- The external network is a layer-3-switched Ethernet with the dynamic routing protocol OSPF, as found in many IBM System z™ environments.
- Avoid purchasing the PowerVM feature (Standard or Enterprise Edition) because the routing approach does not require the use of the Virtual I/O Server.

To summarize: in most typical environments, bridging will be the most appropriate option, being simpler to configure. It should be considered as the default approach.

2.7.4 Virtual and Shared Ethernet configuration example

After having introduced the basic concepts of VLANs, virtual Ethernet, and Shared Ethernet Adapters in the previous sections, this section discusses in more detail how communication between partitions and with external networks operates. The sample configuration in Figure 2-32 on page 98 is used as an example.

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

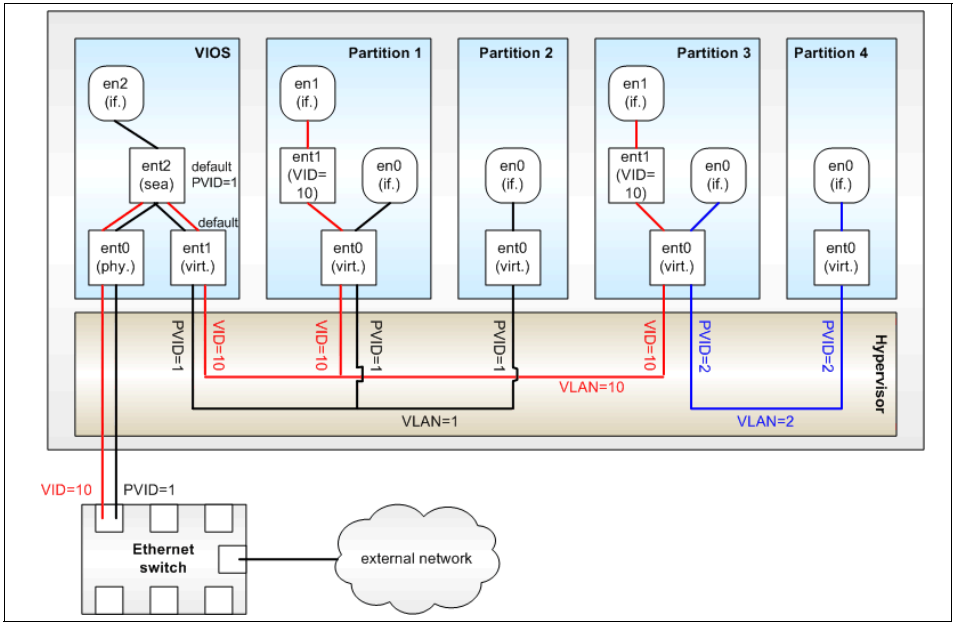


Figure 2-32 VLAN configuration example

Inter-partition networking

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

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

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

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

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

In the configuration shown in Figure 2-32 on page 98, the Virtual I/O Server (VIOS) bridges both VLAN 1 and VLAN 10 through the Shared Ethernet Adapter (SEA) to the external Ethernet switch. But the Virtual I/O Server itself can only communicate with VLAN 1 through its network interface en2 attached to the SEA. Because this is associated with the PVID, VLAN tags are automatically added and removed by the POWER Hypervisor when sending and receiving packets to other internal partitions through interface en2.

Table 2-6 summarizes which partitions in the virtual Ethernet configuration from Figure 2-32 on page 98 can communicate with each other internally through which network interfaces.

Table 2-6 Inter-partition VLAN communication

Internal VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / en0 Virtual I/O Server / en2
2	Partition 3 / en0 Partition 4 / en0
10	Partition 1 / en1 Partition 3 / en1

If the Virtual I/O Server is required to communicate with VLAN 10 as well, then it would need to have an additional Ethernet adapter and network interface with an IP address for VLAN 10, as shown on the left of Figure 2-33 on page 100. A VLAN-unaware virtual Ethernet adapter with a PVID only, as shown in the left of Figure 2-33 on page 100, would be sufficient; there is no need for a VLAN-aware Ethernet adapter (ent4), as shown in the center of Figure 2-33 on page 100. The simpler configuration with a PVID only would be effective, since the Virtual IO/Server already has access to VLAN 1 through the network interface (en2) attached to the SEA (ent2). Alternatively, you could associate an additional VLAN Ethernet adapter (ent3) to the SEA (ent2), as shown on the right in Figure 2-33 on page 100.

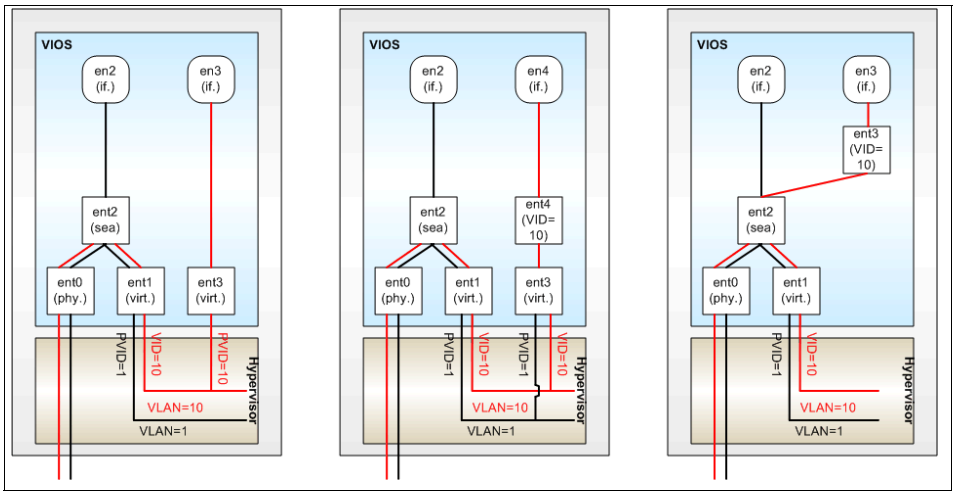


Figure 2-33 Adding virtual Ethernet adapters on the Virtual I/O Server for VLANs

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

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

Communication with external networks

The Shared Ethernet Adapter (SEA) of Figure 2-32 on page 98 is configured with default PVID 1 and default adapter ent1. This means that untagged packets or packets with VID=1 that are received by the SEA from the external network are forwarded to adapter ent1. The virtual Ethernet adapter ent1 has the additional VID 10. Thus, packets tagged with VID 10 will be forwarded to ent1 as well.

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

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

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

In the virtual Ethernet and VLAN configuration example of Figure 2-32 on page 98, partition 1 and partition 2 have access to the external Ethernet through network interface ent0 using PVID 1.

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

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

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

Partition 4 has no access to the external Ethernet.

Table 2-7 summarizes which partitions in the virtual Ethernet configuration from Figure 2-32 on page 98 can communicate with which external VLANs through which network interface.

Table 2-7 VLAN communication to external network

External VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / en0 Virtual I/O Server / en2
10	Partition 1 / en1 Partition 3 / en1

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

- An additional physical Ethernet adapter could be added to Partition 4.
- An additional virtual Ethernet adapter ent1 with PVID=1 could be added to Partition 4. Then Partition 4 would be able to communicate with devices on the external network using the default VLAN=1.
- An additional virtual Ethernet adapter ent1 with PVID=10 could be added to Partition 4. Then Partition 4 would be able to communicate with devices on the external network using VLAN=10.
- VLAN 2 could be added as additional VID to ent1 of the Virtual I/O Server partition, thus bridging VLAN 2 to the external Ethernet, just like VLAN 10. Then Partition 4 would be able to communicate with devices on the external network using VLAN=2. This would work only if VLAN 2 is also known to the external Ethernet and there are some devices on the external network in VLAN 2.
- Partition 3 could act as a router between VLAN 2 and VLAN 10 by enabling IP forwarding on Partition 3 and adding a default route via Partition 3 to Partition 4.

2.7.5 PowerVM Enterprise Edition and virtual Ethernet

A major additional capability of PowerVM Enterprise Edition is PowerVM Live Partition Mobility. PowerVM Live Partition Mobility allows you to migrate running AIX and Linux partitions (and their hosted applications) from one physical server to another without disrupting the infrastructure services. The migration operation maintains complete system transactional integrity. The migration transfers the entire system environment, including processor state, memory, attached virtual devices, and connected users.

For more information, refer to *IBM System p Live Partition Mobility*, SG24-7460.

Note: PowerVM Live Partition Mobility is *not* supported by the PowerVM Standard Edition feature.

PowerVM Live Partition Mobility supports two types of migration; *Inactive migration*, and *Active migration*.

- Inactive migration moves the definition of a powered off logical partition from one system to another along with its network and disk configuration.
- Active migration moves a running logical partition from a source system to a destination system with no disruption of partition operation or user service.

PowerVM Live Partition Mobility requires a specific hardware infrastructure and configuration. The primary requirements for the migration of a logical partition are:

- Two POWER6-based systems controlled by the same HMC (and optional redundant HMC is supported) or two POWER6-based systems controlled by IVM.
- The destination system must have enough processor capacity and memory resources to host the mobile partition.
- The operating system, applications, and data of the mobile partition must reside on virtual storage on SAN storage.
- No physical adapters may be used by the mobile partition at the time of the migration.
- The mobile partition's network and disk access must be virtualized using one or more Virtual I/O Servers.
  - The Virtual I/O Servers on both systems must have a shared Ethernet adapter configured to bridge to the same Ethernet network used by the mobile partition
  - The Virtual I/O Server on both systems must be capable of providing virtual access to all disk resources used by the mobile partition

Note: PowerVM Live Partition Mobility depends on the use of the Virtual I/O Server and fully virtualized resources for both disk storage and networks.

2.8 Linux virtual I/O concepts

In addition to AIX, Linux can also be used on IBM System p. Linux can be installed in a dedicated or shared processor partition. Linux running in a partition can use physical devices and virtual devices. It can also participate in virtual Ethernet and can access external networks through Shared Ethernet Adapters (SEAs). A Linux partition can use virtual SCSI disks.

The following terms and definitions are general:

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

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

PowerVM VIOS The PowerVM Virtual I/O Server from IBM for System p. This is a special-function appliance partition that can only

be used as an Virtual I/O Server, but is not intended to run general applications.

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

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

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

2.8.1 Linux device drivers for IBM System p virtual devices

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

Table 2.1 shows all the kernel modules for IBM System p virtual devices.

Table 2-8 Kernel modules for IBM System p5 virtual devices

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

The Linux 2.6 kernel source can be downloaded from:

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

Precompiled Linux kernel modules are included with some Linux distributions.

2.8.2 Linux as a VIO client

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

Virtual console

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

Virtual Ethernet

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

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

Virtual SCSI client

The IBM virtual SCSI client for Linux is implemented by the ibmvscsic Linux kernel module. When this kernel module is loaded, it will scan and auto-discover any virtual SCSI disks provided by the Virtual I/O Servers. Discovery can also be triggered manually after additional virtual SCSI disks have been added to the virtual SCSI adapter at the Virtual I/O Server.

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

MPIO

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

MPIO involving the Linux client can be configured the following ways:

- MPIO access to the same disk through two Virtual I/O servers
- MPIO access to the same disk through one Virtual I/O server that has two paths to the same disk

MPIO Single client and single Virtual I/O Server

Figure 2-34 on page 106 shows how MPIO can be implemented in the Virtual I/O Server to provide redundant access to external disks for the VIO client. However, implementing MPIO in the Virtual I/O Server instead of in the VIO client does not provide the same degree of high availability to the VIO client, because the VIO client has to be shut down when the single Virtual I/O Server is brought down, for example, when the Virtual I/O Server is upgraded.

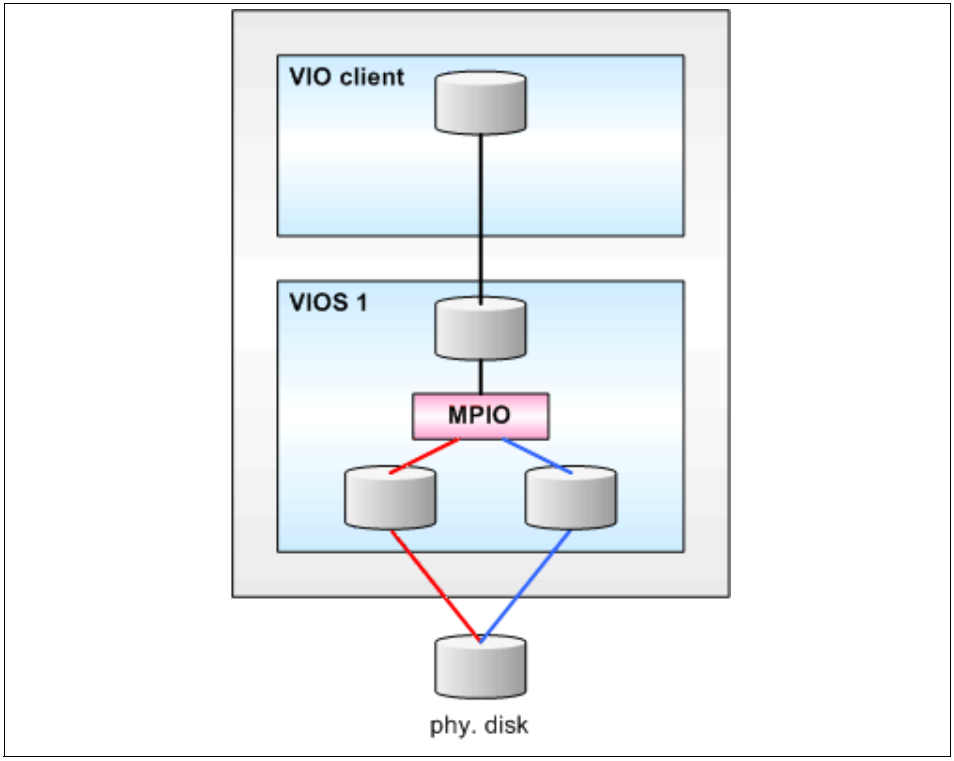


Figure 2-34 MPIO Single Client with Single VIOS with dual paths to the same disk

MPIO Single client and dual Virtual I/O Server

MPIO can also be implemented using a single virtual Linux client and a dual Virtual I/O AIX server where the two Virtual I/O servers are accessing the same disks. This creates an environment of flexibility and reliability. In the event that one Virtual I/O Server is shut down, the virtual client can utilize the other Virtual I/O Server to access the single disk. This capability is possible in SLES 9, SLES 10, RHEL 4, and RHEL 5. Red Hat distributions require one boot parameter for this configuration to work correctly. The parameter “install mpath” must be added to the kernel boot line for the configuration shown in Figure 2-35 on page 107 to work correctly.

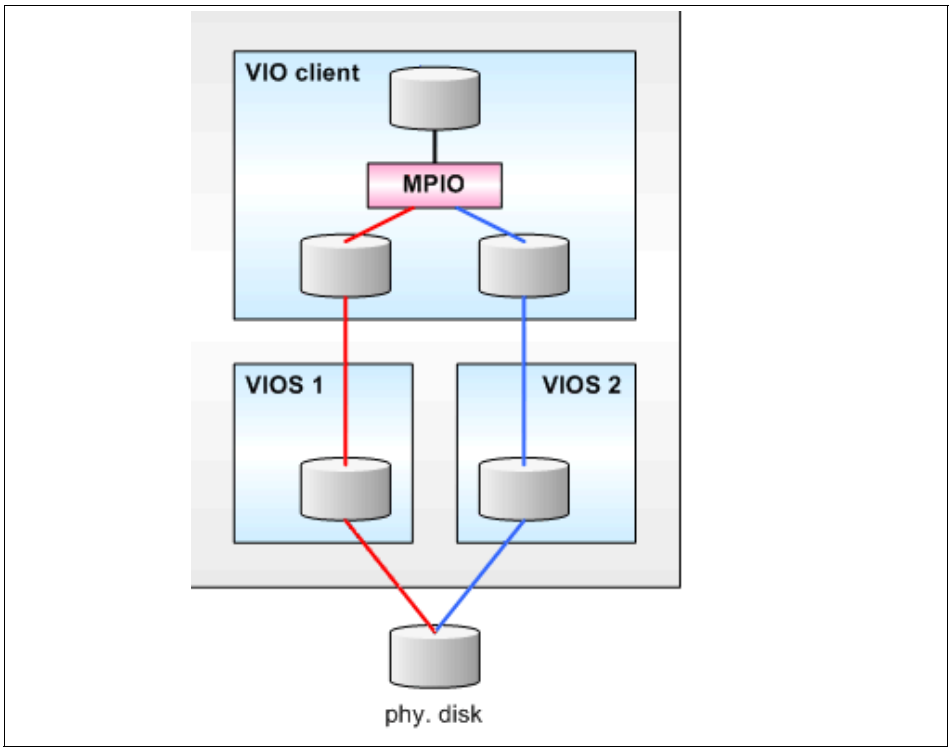


Figure 2-35 MPIO Single Client with Dual VIOS accessing the same disk

Mirroring

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

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

Currently, LVM mirroring of virtual disks within the PowerVM Virtual I/O Server is not recommended. Implementing mirroring in a single Virtual I/O Server instead of mirroring the VIO client storage within the client through two Virtual I/O Servers does not provide the same degree of high availability. This is because the VIO client would lose its connection to the storage when the single Virtual I/O Server is upgraded or serviced. The difference between mirroring in the VIO client and in the Virtual I/O Server is shown in Figure 2-36 on page 108.

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

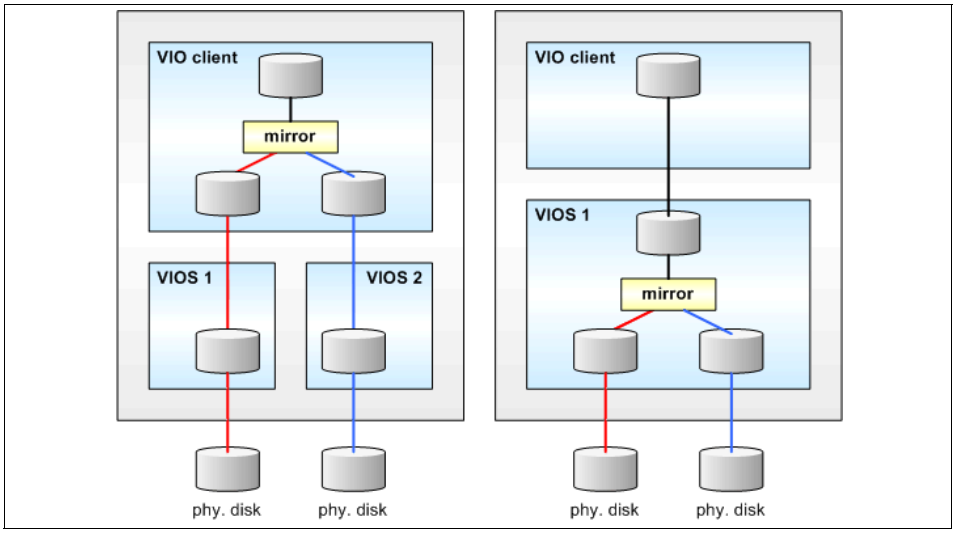


Figure 2-36 Implementing Mirroring in the VIO client or VIO server

LVM

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

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

2.8.3 Considerations

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

- Supported VIO clients are AIX and specific Linux distributions.
- For AIX VIO clients, only the PowerVM Virtual I/O Server is supported by IBM.
- Use of the PowerVM Virtual I/O Server may require the purchase of an additional feature, depending on the IBM System p model.
- Dynamic LPAR on Linux partitions uses the Hot-Plugging subsystem and behaves differently from AIX. Some operations require a reboot of the Linux system, for example, resizing of memory, adding fiber channel cards, etc.

2.8.4 Further reading

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

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

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

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

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

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

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

2.9 Software licensing in a virtualized environment

As the virtualization capabilities of IBM System p servers evolves, so should the software licensing models to support and take best advantage of this. A number of software vendors, including IBM, are working towards new licensing methods to best take advantage of these technologies. By making use of features such as Multiple Shared-Processor Pool technology (MSPP) there is an opportunity for software vendors to provide new and innovative licensing terms, allowing us to be able to define pools of processors supporting different licensed software. Within this shared processor pool all of the LPARs can be uncapped, allowing them to flex within the license boundary set by the MSPP.

For more information it is suggested that you contact your software vendor. For IBM software, at the time of writing, the latest information describing the sub-capacity licensing model can be found here:

http://www-306.ibm.com/software/lotus/passportadvantage/licensing.html

and a statement of direction covering System p virtualization technologies is at:

http://www-01.ibm.com/common/ssi/rep_ca/3/897/ENUS207-273/ENUS207273.PDF

2.9.1 Software licensing methods for UNIX operating systems

Most software vendors license operating systems using a per-CPU model may be applying different weighting factors, depending on the power of the processors. As part of the license agreement most companies will provide some form of software maintenance agreement which provides support and updates for the operating system.

An IBM System p server can have a number of installed processors, not all of which need to be activated. These processors can be supplied as Capacity Upgrade on Demand processors which can be activated with a simple code. For the active processors an operating system license and software maintenance agreement should be in place.

For example, a System p570 server with 16 installed processors, 8 of which are active, would require an operating system software license to cover the 8 active processors only. When a Capacity Upgrade on Demand code is requested, it will cover the additional operating system licensing and software maintenance cost associated with activating additional processors.

2.9.2 Licensing factors in a virtualized system

With the mainstream adoption of virtualization, more and more Independent Software Vendors are adapting their licensing to accommodate the new virtualization technologies. At the time of writing there was no clear sub-processor licensing model prevalent in the industry. Rather, a number of different models exist, varying with the Independent Software Vendors. When calculating the cost of licensing and evaluating which virtualization technology to use, the following factors should be considered:

- ISV recognition of virtualization technology and capacity capping method
- ISV sub-capacity licensing available for selected software products
- ISV method for monitoring and management of sub-capacity licensing
- ISV flexibility as license requirements change

A careful consideration of the licensing factors in advance can help reduce the overall cost in providing business applications. Traditional software licensing has been based on a fixed machine with a fixed amount of resources. With these new PowerVM technologies there are a number of challenges to this model:

- It is possible to migrate partitions between different physical machines (with different speeds and numbers of total processors activated).
- Consider a number of LPARs which, at different times, are all using four processors. However, these can all now be grouped using Multiple Shared Processor Pool technology, which will cap the overall CPU always at four CPUs in total.

Once the ISV support for these technologies is in place, it is anticipated that it will be possible to increase the utilization within a fixed cost of software licenses.

Active processors and hardware boundaries

The upper boundary for licensing is always the quantity of active processors in the physical system (assigned and unassigned), since only active processors can be real engines for software.

Above the physical system level, on IBM System p servers, LPARs can be defined. Most software vendors consider each LPAR as a standalone server and, depending on whether it is using dedicated processors or micro-partitioning, will license software per-LPAR.

The quantity of processors for a certain partition can vary over time, for example with dynamic LPAR operations, but the overall licenses should equal or exceed the total number of processors used by the software at any point in time. If you are using uncapped micro-partitions, then the licensing must take into account the fact that the LPAR can use extra processor cycles above the initial capacity entitlement.

2.9.3 Capacity capping of LPARs

There are two main models for licensing software:

- Pre-pay license based on server capacity or number of users
- Post-pay license based on auditing and accounting for actual capacity used

With most software vendors offering the pre-pay method, the question most vendors will ask will be about how much capacity an LPAR could use. With this in mind the following sections illustrate how to calculate the amount of processing power an LPAR could use.

Dedicated or dedicated donating partitions

In an LPAR with dedicated processors, the initial licensing should be based on the number of processors assigned to the LPAR at activation. Depending on the partition profile maximums, if there are additional active processors or Capacity Upgrade on Demand processors available in the system, these can be added dynamically allowing operators to increase the quantity of processors.

The number of software licenses should be considered before any additional processors are added, even temporarily, for example with dynamic LPAR operations. Clients should note that some ISVs can require licenses for the maximum possible number of processors for each of the partitions where the software is installed (the maximum quantity of processors in the partition profile).

The sharing of idle processor cycles from running dedicated processor partitions will not change the licensing considerations.

Micro-partitions

A number of factors should be considered when calculating the capacity of micro-partitions. To allow the POWER Hypervisor to create micro-partitions the physical processors are presented to the operating system as virtual processors. As micro-partitions are allocated processing time by the POWER Hypervisor, these virtual processors are dispatched on physical processors on a time-share basis.

With each logical processor mapping to a physical processor, the maximum capacity an uncapped micro-partition can use is the number of available virtual processors, assuming:

- This capacity does not exceed the number of active processors in the physical machine.
- This capacity does not exceed the available capacity in the shared processing pool (more on this below).

The following sections discuss the different configurations possible and the licensing implications of each.

Capped micro-partition

For a micro-partition, the desired entitled capacity is a guaranteed capacity of computing power that a partition is given upon activation. For a capped micro-partition, the entitled capacity is also the maximum processing power the partition can use,

Using DLPAR operations, you can vary this between the maximum or minimum values in the profile by executing dynamic LPAR operations.

Uncapped micro-partition without MSPP technology

For an uncapped micro-partition, the entitled capacity given to the partition is not necessarily a limit on the processing power. An uncapped micro-partition can use more than the entitled capacity if there are some available resources within the system.

In this case, on a POWER5 system using the Shared Processor Pool or using only the default shared processor pool on POWER6, the limiting factor for uncapped micro-partition is the number of virtual processors. The micro-partition can use up to the number of physical processors in the shared processor pool, since each virtual processor is dispatched to a physical processor.

With only the single pool, the total resources available in the shared processor pool will be equal to the activated processors in the machine minus any dedicated (non-donating) partitions. This assumes that at a point in time all other partitions will be completely idle.

The total licensing liability for an uncapped partition without MSPP technology will be either the number of virtual processors or the number of processors in the shared processor pool, whichever is smallest.

Uncapped micro-partition with MSPP technology

As above, for an uncapped micro-partition the entitled capacity given to the partition is not necessarily a limit on the processing power. An uncapped micro-partition can use more than the entitled capacity if there are some available resources within the system.

For POWER6 systems, using Multiple Shared Processor Pool technology, it is possible to group micro-partitions together and place a limit on the overall group maximum processing units. After defining a Shared Processor Pool group, operators can group specific micro-partitions together that are running the same software (software licensing terms permitting) allowing a pool of capacity that can then be shared among a number of different micro-partitions. However, overall, the total capacity used at any point in time will never exceed the pool maximum.

Summary of licensing factors

Depending on the licensing model supported by the software vendor, it is possible to work out licensing costs based on:

- Capped versus uncapped
- Number of virtual processors
- Unused processing cycles available in the machine, from dedicated donating LPARs and other micro-partitions

- (Multiple) shared processor pool maximum
- Active physical processors in the system

An example of the license boundaries is illustrated in Figure 2-37.

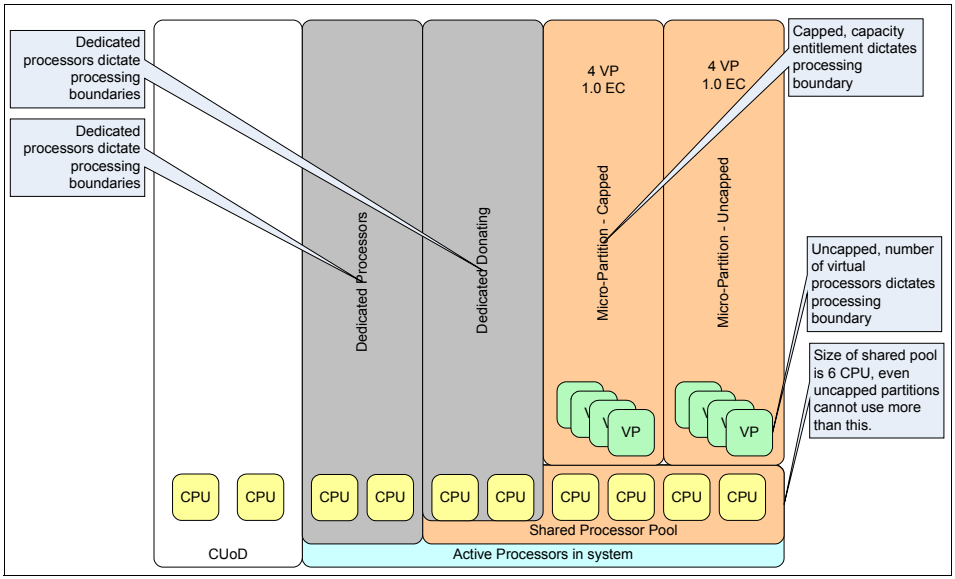


Figure 2-37 License boundaries with different processor and pool modes

System with Capacity Upgrade on Demand processors

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

- They become temporarily or permanently active and are assigned to LPARs
- They become temporarily or permanently active in systems with PowerVM and can be used by micro-partitions.

Clients can provision licenses of selected software for temporary or permanent use on their systems. Such licenses can be used to align with the possible temporary or permanent use of CUoD processors in existing or new AIX or Linux partitions.

For more information about processors on demand On/Off, refer to 2.10.4, “Capacity Upgrade on Demand” on page 123.

2.9.4 License planning and license provisioning of IBM software

With the widespread introduction of multi-core chips, IBM software is moving to a licensing model based on a Processor Value Unit.

In the past a new processor may have provided a throughput gain with a speed increase over an existing generation. Nowadays a new processor may improve throughput using other methods, such as:

- Number of cores per physical chip
- Number of threads each core can simultaneously dispatch
- Size and speed of cache
- Specialist processing units, for example decimal floating point in POWER6

The traditional measure for software licensing, the number and speed of processors, is no longer a direct comparison of chip performance. In addition to this vendors are introducing virtualization technologies, allowing the definition of logical servers that use fractions of processing power.

The Processor Value Units will allow the licensing of software to reflect the relative processor performance and allow the licensing of sub-capacity units.

For non-IBM software it is suggested that you contact your independent software vendor sales representative. For more information on IBM software licensing, see the links in 2.9, “Software licensing in a virtualized environment” on page 109.

Selected IBM software programs eligible under IBM Passport Advantage® and licensed on a per-processor basis may qualify for sub-capacity terms, so licenses are required only for those partitions where the programs are installed. To be eligible for sub-capacity terms, the client must agree to the terms of the IBM International Passport Advantage Agreement Attachment for sub-capacity Terms.

Capacity Upgrade on Demand

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

Initial licensing The client calculates the initial license entitlements, based on the licensing rules and the drivers for licensing. The client purchases processor licenses based on the planned needs. The client can also purchase temporary On/Off licenses of selected System p related software.

Additional licensing The client checks the actual usage of software licenses or future planned needs and calculates the additional license entitlements (temporary On/Off licenses also) based on the licensing rules and the drivers for licensing.

On demand licensing The client contacts IBM or a Business Partner for the submission of a Passport Advantage Program

enrollment. The client follows the procedures of the licensing method (sub-capacity licensing for selected IBM Passport Advantage eligible programs) including any auditing requirements and is billed for the capacity used.

2.9.5 Sub-capacity licensing for IBM software

With the introduction of POWER6 and AIX 6.1 technologies, there have been a number of challenges to the traditional licensing model, the traditional model being based on a fixed number of processors in a machine with a fixed serial number. Some of these challenges are discussed in 2.9.2, “Licensing factors in a virtualized system” on page 110.

IBM already offers some software under sub-capacity licenses. For more information on the terms, see:

http://www-306.ibm.com/software/lotus/passportadvantage/licensing.html

At the time of writing this book, IBM is working to adopt these technologies into their licensing model. More information can be found at:

http://www-01.ibm.com/common/ssi/rep_ca/3/897/ENUS207-273/ENUS207273.PDF

Example of how sub-capacity licensing might be used

At the time of writing this book the final licensing scheme was not available. However, it is anticipated that by using micro-partitioning an implementation similar to the following may be possible.

Considering the traditional server with a fixed number of cores running an MQ and DB2® workload, the licensing calculation might be similar to that in Figure 2-38 on page 117.

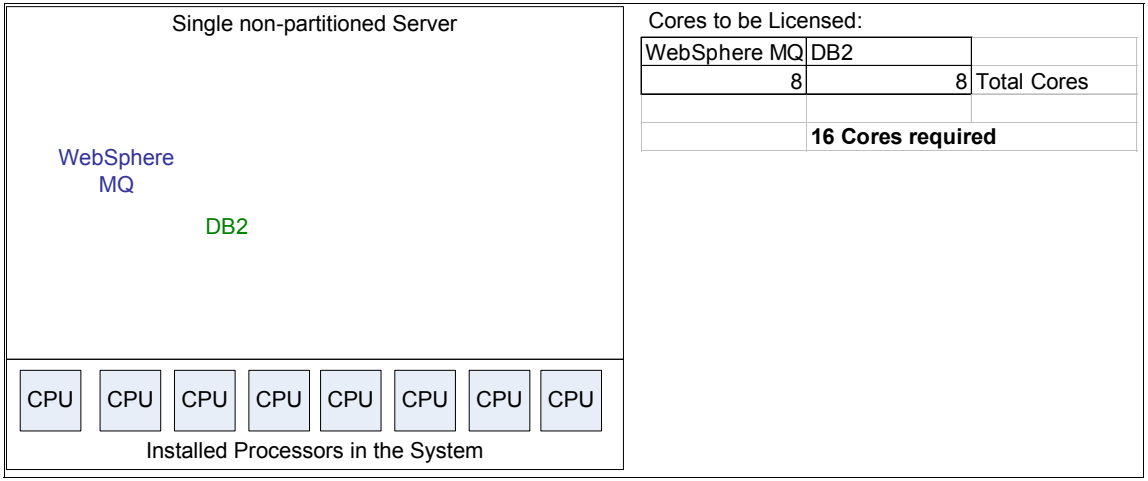


Figure 2-38 Licensing requirements for a non-partitioned server

Taking advantage of IBM System p virtualization technology, depending on the final licensing terms, it would be possible to take advantage of micro-partitioning to reduce the licensing costs in a manner similar to that in Figure 2-39.

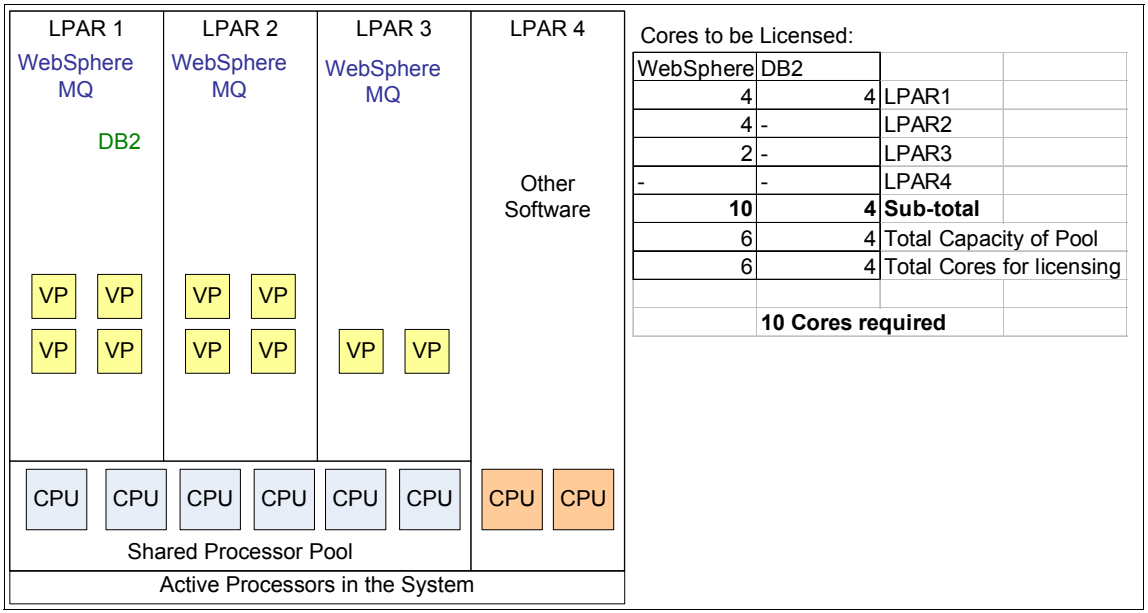


Figure 2-39 Licensing requirements in a micro-partitioned server

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

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

2.9.6 Linux operating system licensing

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

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

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

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

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

2.10 Introduction to simultaneous multithreading

Simultaneous Multithreading (SMT) is a method used to increase the throughput for a given amount of hardware. The principle behind SMT is to allow instructions from more than one thread to be executed at the same time on a processor. This allows the processor to continue performing useful work even if one thread has to wait for data to be loaded.

To perform work a Central Processing Unit needs input information in the form of instructions. Ideally this information will have been loaded into the CPU cache, which allows the information to be quickly accessed. If this information cannot be found in the processor cache, it must be fetched from other storage (other levels of cache, memory or disk) which, in computer terms, can take a long time. While

this is happening, the CPU has no information to process, which can result in the CPU idling instead of performing useful work.

2.10.1 POWER processor SMT

The SMT implementation for POWER6 differs slightly from that in POWER5, increasing the throughput improvements offered by SMT as seen in most commercial workloads.

Comparing, at a high level, the implementation of SMT in the two processors:

- In POWER5, the processor uses two separate program counters, one for each thread. Instruction fetches alternate between the two threads. The two threads share the instruction cache.
- In POWER6, the two threads form a single group of up to seven instructions to be dispatched simultaneously (with up to five from a single thread), increasing the throughput benefit over POWER5 by between 15 to 30 percent.

Although almost all applications benefit from SMT, some highly optimized workloads may not. For this reason, the POWER processors support *single-threaded* (ST) execution mode. In this mode, the POWER processor gives all the physical processor resources to the active thread.

It is possible to switch between ST and SMT modes dynamically (without rebooting) in AIX 5.3 or 6.1. Linux partitions require a restart to change SMT mode.

The benefit of SMT is greatest where there are numerous concurrently executing threads, as is typical in commercial environments, for example, for a Web server or database server. Some specific workloads, for example certain high performance computing workloads, will generally perform better with ST.

2.10.2 SMT and AIX

The AIX operating system scheduler dispatches execution threads to logical processors. Dedicated and virtual processors have one or two logical processors associated with them, depending on whether SMT is enabled or not. Within each partition it is possible to list the dedicated or virtual processors and their associated logical processors. Because a pair of logical processors will be associated with a single virtual or dedicated processor, they will all be in the same partition.

Figure 2-40 on page 120 shows the relationship between physical (at the bottom of the figure), dedicated (the physical tied to a CPU, not a virtual CPU) and

logical processors for dedicated and micro-processor partitions. It is possible to have some micro-partitions with SMT enabled and others with SMT disabled running within the same physical system.

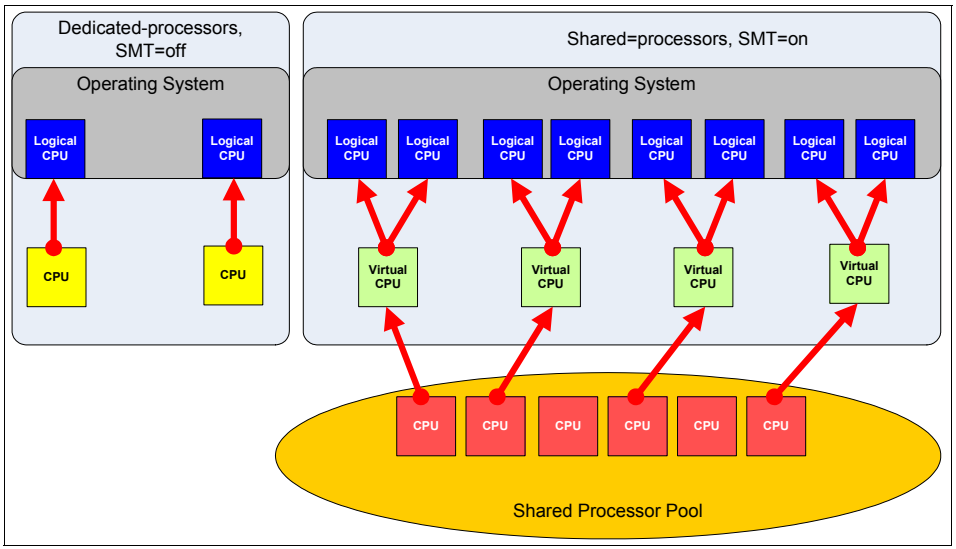


Figure 2-40 Physical, virtual, and logical processors

SMT control in AIX

SMT is controlled by the AIX **smtctl** command or with the system management interface tool (SMIT). SMT can be enabled or disabled in two different ways:

- Dynamically on a logical partition
- After the next operating system reboot is performed

Setting SMT mode using the command line

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

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

-m off Will set SMT mode to disabled.

-m on Will set SMT mode to enabled.

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

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

The **smtctl** command does not rebuild the boot image. If you want to change the default SMT mode of AIX, the **bosboot** command must be used to rebuild the

boot image. The boot image in AIX Version 5.3 and later has been extended to include an indicator that controls the default SMT mode.

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

The **smtctl** command entered without a flag will show the current state of SMT in the partition. An example of the **smtctl** command in AIX 6.1 follows:

```
# smtctl

This system is SMT capable.
SMT is currently enabled.
SMT boot mode is not set.
SMT threads are bound to the same virtual processor.

proc0 has 2 SMT threads.
Bind processor 0 is bound with proc0
Bind processor 1 is bound with proc0


proc2 has 2 SMT threads.
Bind processor 2 is bound with proc2
Bind processor 3 is bound with proc2
```

Setting SMT mode using SMIT

Use the **smitty smt** fast path to access the SMIT SMT control panel. From the main SMIT panel, the selection sequence is **Performance & Resource Scheduling → Simultaneous Multi-Threading Processor Mode → Change SMT Mode**. Figure 2-41 shows the SMIT SMT panel.

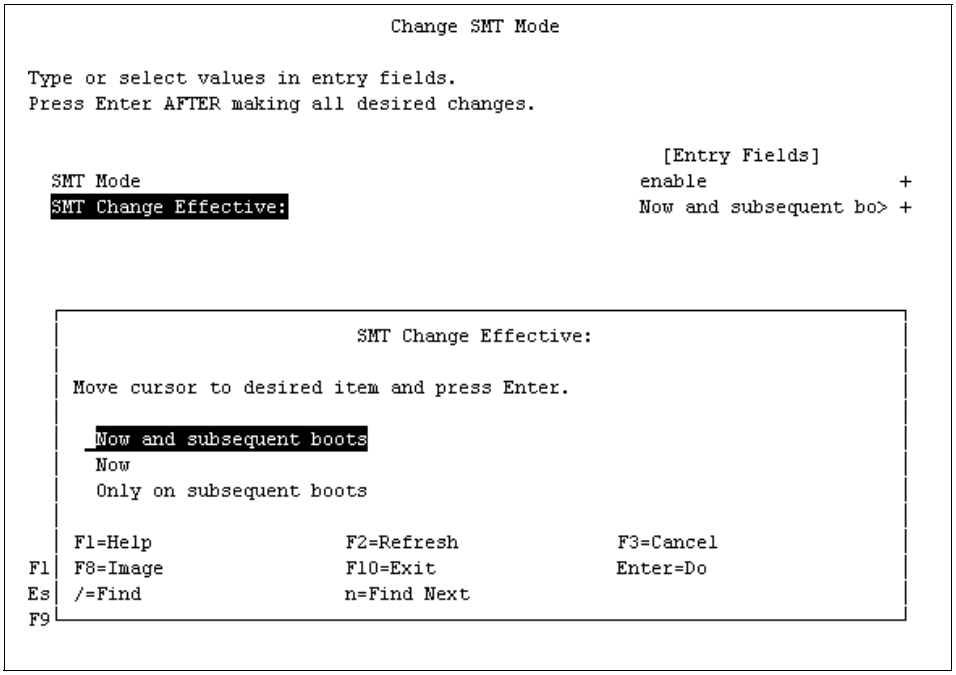


Figure 2-41 SMIT SMT panel with options

SMT performance monitor and tuning

AIX includes additional commands or extended options to existing commands for the monitoring and tuning of system parameters in SMT mode. For more information, see *Advanced POWER Virtualization: Managing and Monitoring*, SG24-7590.

2.10.3 SMT control in Linux

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

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

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

2.10.4 Capacity Upgrade on Demand

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

Capacity Upgrade on Demand (CUoD) for System p servers allows you to increase available processor and memory resource, paying for increased capacity only as your needs grow, through the following:

- Permanent capacity for non-disruptive growth

  Provides a means of planned growth for clients who know they will need increased capacity but aren't sure when. This allows you to install more processors and memory than are initially required and then bring that additional capacity online as demand increases. With AIX and Linux operating environments that support dynamic LPAR, processor and memory resource can be allocated dynamically without interrupting system or partition operations.

  Available processor resource can be incremented in one processor units while memory can be incremented in 1 GB units. As workload demands require more processing power, you can increase available processor resource by placing an order for an activation feature code. You receive an electronically encrypted activation key over the Internet that unlocks the desired amount of processors. There is no hardware to ship and install, and no additional contract is required.

- On/Off Capacity on Demand

  Provides for planned and unplanned short-term growth driven by temporary processing requirements such as seasonal activity, period-end requirements or special promotions. Activation is user managed. On/Off CoD provides a flexible capability for planned or predictable peek processing requirements for p570, p5-570 (1.65/1.9/2.2 GHz), p5-590, and p5-595 servers.

  When you order an On/Off CoD feature for p570, p5-570, p5-590, and p5-595 servers, you receive an enablement key that enables the administrator to turn processors and memory on and off as desired, in increments of one processor and 1 GB of memory. The system monitors the amount and duration of the activations and generates a usage report which must be sent to IBM monthly. Billing for the processors and memory is then based on the usage report on a quarterly basis. You pay only for the processor days and memory days after you have used them.

- Reserve Capacity on Demand

  Provides automated use of CUoD processors on IBM POWER5 processor-based systems including the p5-570, p5-590 and p5-595 servers for planned and unplanned short-term workloads which are run in a shared processor pool.

  Reserve CoD allows p5-570 (1.65/1.9/2.1 GHz), p5-590 and p5-595 servers to have optimized, automatically managed temporary allocation of CUoD processors. The user purchases a block of 30 Processor Days of usage time and then assigns processors to the shared processor pool. If the demand for processing power from the partitions in the shared pool is both less than the defined maximum processor allocation and more than the available assigned resource, processors can be allocated from the reserve pool. The server then automatically manages the workload and only charges against the Processor Day account when the workload exceeds 100% of the base (permanently activated) processors in the shared processing pool by 1/10 of a processor unit.

- Utility Capacity on Demand

  Utility CUoD provides automated use of CUoD processors on the IBM POWER6 processor-based systems including the IBM System p570 for planned and unplanned short-term workloads which are run in a shared processor pool.

  Utility CoD is for clients with unpredictable or short-term workload demand spikes who need an automated way to deliver sufficient server performance as needed. Use is measured in Processor Minute increments, and is reported by the client at the Utility Capacity On Demand Web site. Clients have the option to pre-pay or post-pay for processor minutes.

- Trial Capacity on Demand

  Provides the flexibility to evaluate how additional resources will affect existing workloads, or to test new applications.

  Owners of System p servers with CUoD capabilities can use all available CUoD processors and memory resources one time, for up to 30 contiguous days, at no charge. Trial Capacity on Demand is enabled by registering at the System p CoD Web site and electronically receiving an activation key. When clients purchase permanent processor activation on p570, p5-570, p5-590, and p5-595 servers, they have another 30 contiguous day trial. These subsequent trials are limited to the use of two processors and 4 GB of memory.

- Capacity BackUp

  Supplements temporarily back up p5-590 and p5-595 servers with the extra processing capacity they need in case of an unforeseen production server outage.

Capacity BackUp (CBU) is available for p5-590 and p5-595 servers to provide a system backup capability. CBU configurations include four active processors and either 28 or 60 inactive processors and 900 or 1800 days of pre-paid On/Off processor days. Clients may purchase additional capacity or processor days at regular On/Off CoD prices. CoD memory may also be purchased as part of the backup procedures.

The full resources in a Capacity BackUp system can be temporarily turned on at any time by using the On/Off CoD activation procedure. This enables the client to have the necessary performance during an unplanned system outage.

If IBM HACMP V5 or HACMP/XD software (5765-F62) is installed, scripts can be launched during a failover process which can automatically activate LPAR and CoD resources. This assists clients to provide round-the-clock business continuity and disaster recovery with no loss of data without any need for manual intervention. For more information on how to use HACMP with CBU for automated backup capability, see Appendix H: Using DLPAR and CUoD in an HACMP Cluster in *HACMP for AIX 5L Administration Guide Version 5.4,* SC23-4862.

The ability of CBU configurations to effectively back up production systems can be tested using Trial CoD. This temporary backup is most frequently purchased to provide disaster recovery or critical production backup, but the same technology can be applied in other circumstances where production is temporarily moved. This could include upgrades, planned maintenance, and other similar activities.

Capacity BackUp offerings are not intended to be used as full-time backup servers for 24x7 high-availability solutions. Since CBU is intended for temporary backup, additional processor resource cannot be permanently activated. High usage of additional processors may add significant cost to an IT operation.

Before removing a CUoD processor (for example, when using On/Off CUoD, which enables users to temporarily activate processors), you should ensure that there are enough processing units remaining to continue running the workload when the target processor is deactivated. You can dynamically remove the needed capacity entitlement from the partitions.

For further CUoD information, refer to the following Web site:

`http://www.ibm.com/systems/p/advantages/cod/index.html`

2.10.5 Instruction retry recovery

POWER6 processors introduce a new state-of-the art reliability feature, instruction retry recovery (IRR). IRR is usually only available on mainframe systems. IRR provides full hardware error recovery of those registers that are defined by the instruction set architecture. This is accomplished by taking a checkpoint of the defined state for both of the core threads and recovering the machine state back to a known good point.

2.10.6 Dynamic processor de-allocation and processor sparing

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

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

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

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

2.10.7 Dynamic resources

In both dedicated-processor partitions and micro-partitions, resources can be dynamically added and removed. In order to execute dynamic LPAR operations, each LPAR required communication access to the HMC or IVM.

Dedicated-processor partitions

Support for dynamic resources in dedicated-processor partitions provides for the dynamic movement of the following resources:

- One dedicated processor

- A 16 MB memory region (dependent on installed memory)
- One I/O adapter slot (either physical or virtual)

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

Dynamic resources are supported in dedicated-processor partitions running AIX 5L Version 5.2 or later.

Micro-partitions

The resources and attributes for micro-partitions include processor capacity, capped or uncapped mode, memory, and virtual or physical I/O adapter slots. All of these can be dynamically changed. For micro-partitions, it is possible to carry out the following operations dynamically:

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

Dynamic LPAR operations

There are several things you should be aware of when carrying out dynamic LPAR operations that pertain to both virtualized and non-virtualized server resources:

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

2.10.8 Shared processor considerations

The following considerations should be taken into account when implementing micro-partitions:

- The minimum size for a micro-partition is 0.1 processing units of a physical processor. So the number of micro-partitions you can create for a system depends mostly on the number of activated processors in a system.
- The maximum number of micro-partitions supported in a System p server is 254.
- The maximum number of virtual processors in a micro-partition is 64.
- The minimum number of processing units you can have for each virtual processor depends on the server model. The maximum number of processing units that you can have for each virtual processor is always 1.00. This means that a micro-partition cannot use more processing units than the number of virtual processors that it is assigned, even if the micro-partition is uncapped.
- A partition is either a dedicated-processor partition or a micro-partition, it cannot be both. However, processor capacity for a micro-partition may come from Shared Dedicated Capacity. This is unused processor capacity from processor(s) dedicated to a partition but are capable of capacity donation. This situation does not change the characteristics of either the dedicated-processor partition or the micro-partition.
- If you wish to dynamically remove a virtual processor, you cannot select a specific virtual processor to be removed. The operating system will choose the virtual processor to be removed.
- AIX will utilize affinity domain information provided by firmware (POWER Hypervisor) to build associations of virtual processors to memory, and it will continue to show preference to redispatching a thread to the virtual processor that it last ran on. However, this cannot be guaranteed in all circumstances.
- An uncapped micro-partition with a weight of 0 is effectively the same as a micro-partition that is capped. This is because it will never receive any additional capacity above it's capacity entitlement. The HMC or IVM can dynamically change the weighting of a micro-partition. Similarly, the HMC or IVM can change the mode of a micro-partition from capped to uncapped (and the reverse).

Number of virtual processors

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

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

Table 2-9 shows several reasonable settings of number of virtual processor, processing units, and the capped and uncapped mode.

Table 2-9 Reasonable settings for shared processor partitions

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

a - Virtual processors
b - Processing units

Virtual and physical processor relationship

The nature of the Micro-Partitioning technology increases system flexibility and utilization. Part of the mechanism enabling these benefits is the abstraction of processors into virtual processors and then scheduling the virtual processors on to physical processors to run. This approach guarantees that the correct processor capacity is delivered consistently to the micro-partition running during the POWER Hypervisor dispatch window.

Virtual processors have the concept of a home physical processor. The POWER Hypervisor tries to maximize and capitalize on memory affinity wherever possible. To do this, the POWER Hypervisor attempts to schedule the virtual processor to the same physical processor whenever possible, but if this is unavailable, it will broaden the search in a specific manner to gain the most benefit from whatever memory affinity is achievable.

Some transactional workloads are very sensitive to any variance in the elapsed time to complete each individual transaction rather than a volume of transactions in a given (short) time. Due to the nature of scheduling for virtual processors, it may not be possible to deliver this level of transaction consistency in all circumstances. If you have such a sensitive workload, you should undertake testing to determine the correct partitioning approach to use.

2.11 Integrated Virtual Ethernet overview

POWER6-based systems have extended the virtualization capabilities for networking solutions. Integrated Virtual Ethernet (IVE) is the collective name referring to a number of technologies including:

- Hardware - Host Ethernet Adapter (HEA)
- Software components
- POWER Hypervisor functions

Together these technologies make up IVE and provide integrated high-speed Ethernet adapter ports with hardware-assisted virtualization capability.

IVE is a standard set of features that is offered on selected POWER6-based System p servers. You can select from different offerings according to the specific IBM System p server. IVE offers the following capability:

- Either two 10 Gbps Ethernet ports (on fiber), or four 1 Gbps Ethernet ports (on copper), or two 1 Gbps Ethernet ports (on copper)
- External network connectivity for partitions using dedicated ports without the need for a Virtual I/O Server
- Industry-standard acceleration with flexible configuration possibilities
- The performance of the GX+ bus

IVE offers greater throughput and lower latency than PCIe or PCI-X bus connected Ethernet adapters because it is connected to the GX+ bus of the POWER6-based System p server. In addition, IVE includes special hardware features that provide logical Ethernet adapters which can communicate directly with logical partitions running on the system.

IVE offers an alternative to the use of virtual Ethernet or the Shared Ethernet Adapter service of the Virtual I/O Server.

Figure 2-42 on page 131 shows a sample Virtual I/O Server and Integrated Virtual Ethernet configuration.

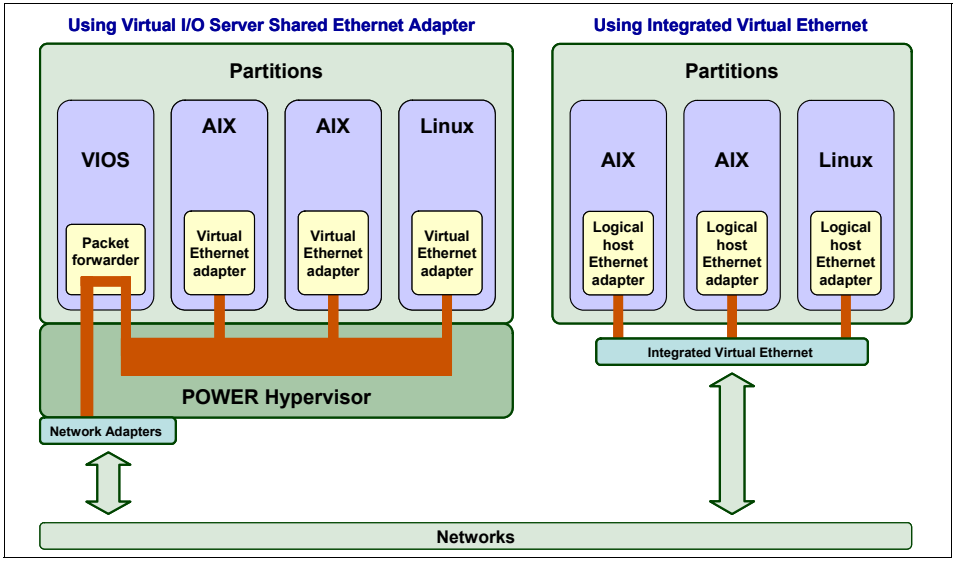


Figure 2-42 Virtual I/O Server SEA comparison with Integrated Virtual Ethernet

The IVE design is comprised of the virtualization of a network port that can connect to external networks and share this connectivity to all virtualized logical ports without using the Virtual I/O Server. This approach can reduce complexity in overall system deployment.

Logical ports have the following characteristics:

- Logical ports are grouped together into a port-group and any port-group offers 16 ports with one or two physical ports in the group.
- An IVE has layer-2 switches, one per physical port-group.
- Each logical port can be assigned to any partition and partitions can have one logical port per physical port.
- Every logical port on a particular IVE has a unique MAC address which is determined by a base-address stored in the VPD (vital product data) of that card.
- A logical port belongs only to one physical port and one virtual switchq—there may be two logical ports in the same port-group that do not share a virtual switch if located on different physical ports.
- From a logical point of view, any logical port could be the Ethernet interface of a client device.
- Partitions that have logical ports associated with the same physical port (and therefore the same virtual layer-2 switch) and can communicate without the need for external hardware.

- You can use NIB between 2 logical ports on different physical adapters to provide redundancy.

For more information regarding Integrated Virtual Ethernet, refer to the Redpaper, *Integrated Virtual Ethernet Adapter Technical Overview and Introduction*, REDP-4340.

3

Setting up virtualization: the basics

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

The topics include:

- Getting started
- Creating a Virtual I/O Server partition
- Virtual I/O Server software installation
- Basic Virtual I/O Server scenario
- Interaction with UNIX client partitions
- Using the System Planning Tool for deployment
- Using system plans to document the configuration

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

3.1 Getting started

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

- 3.1.1, “Command line interface” on page 134 of the Virtual I/O Server, also named the IOSCLI
- 3.1.2, “Hardware resources managed” on page 138
- 3.1.3, “Software packaging and support” on page 139

3.1.1 Command line interface

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

Important: Only supported 3rd party storage configuration should be done under the oem_setup_env shell environment. The **cfgassist** command offers smit menus for common configuration tasks, as shown in Figure 3-1.

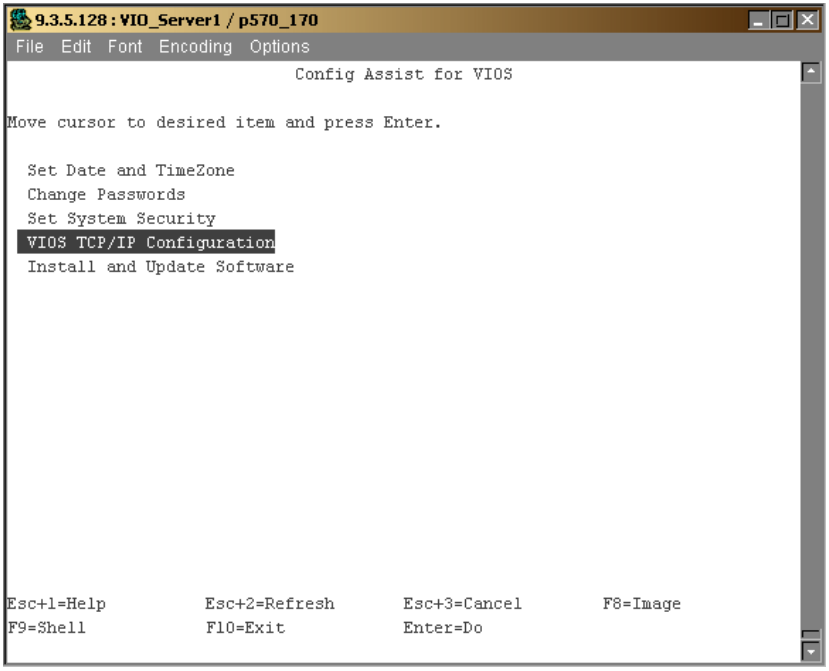


Figure 3-1 The Config Assist for VIOS menu

The following Virtual I/O Server administration activities are done through the Virtual I/O Server command line interface:

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

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

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

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

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

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

Example 3-1 Supported commands on Virtual I/O Server Version 1.5

```
$ help
Install Commands                        Security Commands
 ioslevel                                lsfailedlogin
 license                                 lsgcl
 lssw                                    viosecure
 oem_platform_level                      mkldap
 oem_setup_env                           ldapadd
 remote_management                       ldapsearch
 updateios                               snmpv3_ssw
                                         mkkrb5clnt
```

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

Device Commands
 chdev
 chpath
 cfgdev
 lsdev
 lsmap
 lspath
 mkpath
 mkvdev
 mkvt
 rmdev
 rmpath
 rmvdev
 rmvt

Physical Volume Commands
 lspv
 migratepv

Logical Volume Commands
 chlv
 cplv
 extendlv
 lslv
 mklv

UserID Commands
 chuser
 lsuser
 mkuser
 passwd
 rmuser

Maintenance Commands
 alt_root_vg
 backup
 backupios
 bootlist
 cattracerpt
 chdate
 chlang
 cfgassist
 cl_snmp
 dsmc
 diagmenu
 errlog
 fsck
 invscout
 ldfware
 loginmsg
 lsfware
 lslparinfo
 motd
 mount
 pdump
 replphyvol
 restore
 restorevgstruct
 save_base
 savevgstruct
 showmount
 shutdown
 snap
 snmp_info
 snmp_trap
 startsysdump
 starttrace
 stoptrace
 svmon
 sysstat
```

```
 mklvcopy
 rmlv
 rmlvcopy

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

Storage Pool Commands
 chbdsp
 chsp
 lssp
 mkbdsp
 mksp
 rmbdsp
 rmsp

Virtual Media Commands
 chrep
 chvopt
 loadopt
 lsrep
 lsvopt
 mkrep
 mkvopt
 rmrep
 rmvopt
 unloadopt
 topas
 uname
 unmount
 viostat
 vmstat
 wkldmgr
 wkldagent
 wkldout

Monitoring Commands
 cfgsvc
 lssvc
 startsvc
 stopsvc

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

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

Example 3-2 Help command

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

       Displays or clears the error log.

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

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

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

3.1.2 Hardware resources managed

The PowerVM feature (Standard or Enterprise) that enables Micro-Partitioning on a server provides the Virtual I/O Server installation media. A Virtual I/O Server partition with dedicated physical resources to share to other partitions is also required. The following minimum hardware requirements must be met to create the Virtual I/O Server:

POWER5 or POWER6 server

The Virtual I/O capable machine.

Hardware Management Console or IVM

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

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

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

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

Memory At least 512 MB of memory is required. Similar to an operating system, the complexity of the I/O subsystem and the number of the virtual devices have a bearing on

the amount of memory required, such as when using multiple paths to SAN storage devices.

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

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

Virtual devices exported to client partitions by the Virtual I/O Server must be attached through supported adapters. An updated list of supported adapters and storage devices is found at:

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

Careful planning is recommended before you begin the configuration and installation of your Virtual I/O Server and client partitions. Depending on the type of workload and needs of an application, it is possible to mix virtual and physical devices in the client partitions.

3.1.3 Software packaging and support

Installation of the Virtual I/O Server partition is performed from a special DVD-ROM that is provided to clients that order the PowerVM feature. For the p5-590 and p5-595, this feature is included. The Virtual I/O Server software is only supported in Virtual I/O Server partitions.

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

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

Example 3-3 shows the steps to copy the DVD mksysb image of the Virtual I/O Server on to a NIM repository.

Example 3-3 Copying the Virtual I/O Server DVD media on to a NIM server

```
# mount /cdrom
# cd /cdrom
# ls
.Version      RPMS          installp      nimol         sbin
OSLEVEL       bosinst.data  ismp          ppc           udi
README.vios   image.data    mkcd.data     root          usr
# cd usr/sys/inst.images
# ls -l
total 3429200
-rw-r--r--   1 root     system   1755750400 Jun 06 2007  mksysb_image
# cp mksysb_image /nim/images/vios1.mksysb
# cp /cdrom/bosinst.data /nim/resources
```

Important:

- Copy the bosinst.data file from the Virtual I/O Server DVD to the NIM repository and define it as a NIM resource. Specify this resource when you set up your NIM installation. The bosinst.data script will perform the SSH For IVM, the DVD/CD drive is automatically virtualized from the Virtual I/O Server to be available to partitions.
- The **installios** command is not applicable for IVM-managed systems.
- installation from the Virtual I/O Server mksysb image.

For more information about the installation of the Virtual I/O Server, refer to 3.4, “Virtual I/O Server software installation” on page 173.

3.2 Virtual I/O Server enhancements and updates

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

To learn how to use the Virtual I/O Server in detail, see the following Web page for additional information:

http://www14.software.ibm.com/infocenter/systems/scope/hw/index.jsp?topic=/iphb1/iphb1kickoff.htm

Installing a fixpack effectively updates an installed Virtual I/O Server to the new level.

3.2.1 Tracking the latest virtualization enhancements

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 and existing features:

http://techsupport.services.ibm.com/webapp/set2/sas/f/vios/documentation/home.html

New features in Version 1.2 of the Virtual I/O Server

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

- Virtual optical device support
- Shared Ethernet Adapter failover
- Integrated Virtualization Manager
- Storage pool commands
- HMC enhancements for easier configuration and maintenance

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

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

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

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

- Virtual I/O Server

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

  - Newly added software: SSH now comes preinstalled with the Virtual I/O Server.
  - Improved monitoring with additional **topas** and **viostat** command performance metrics.

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

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

    ```
    $ chdev -dev sys0 -attr iostat=true
    ```
  - Enabled Performance (PTX) agent (PTX is a separately purchased LPP.)
  - Increased performance for virtual SCSI and virtual Ethernet

    Due to improvements in the code, the performance of both is increased.
  - Enhanced command-line interface

    Several new commands have been added:

    - The **wklmgr** and **wkldagent** commands for handling Workload Manager. If enabled, they can be used to record performance data that can then be analyzed by **wkldout** and viewed.
    - The **chtcpip** command for online management of TCP/IP parameters.
    - The **crontab** command for handling cron jobs.
    - The **viosecure** command for handling the security settings. This is in fact a central point to control security settings of the Virtual I/O Server, also handling the local firewall settings to secure the Virtual I/O Server based either on standard values or customized on a per-port basis.

    Other commands that received enhancements are **startnetsvc**, **stopnetsvc**, **lstcpip**, **lsdev**, and **mkbdsp**.
  - Enabled additional storage solutions

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

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

- Virtual SCSI and shared Fibre Channel adapters

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

  - Support for iSCSI TOE adapter and virtual devices backed by System Storage™ N series iSCSI disks
  - Virtual SCSI (VSCSI) functional enhancements
    - Support for SCSI reserve/release for limited configurations. Enables virtual SCSI for certain clustered/distributed configurations.
    - Support on the AIX 5L client for a changeable queue depth attribute for virtual disks. Allows the system administrator to tune the size of the I/O queue for each virtual disk.

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

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

      where N is the number of the hdisk and X denotes the desired depth of the queue. Therefore it is now possible to have values larger than three, which was the previous non-changeable default. Improvements for altering virtual device capacity without disruption.

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

      ```
      # chvg -g volume_group
      ```

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

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

      The vscsi_path_to attribute of the virtual SCSI client adapter was added. A value of 0 (default) disables it, while any other value defines the number of seconds the VSCSI client adapter will wait for commands issued to the VSCSI server adapter and not serviced. If that time is exceeded, the VSCSI client adapter attempts the commands again and waits up to 60 seconds until it fails the outstanding

commands, writes an error to the error log and, if MPIO is used, tries another path to the disk. Therefore, this parameter should only be set for MPIO installations with dual Virtual I/O Servers.

- VSCSI device error log enhancements.

► Virtual Ethernet and shared Ethernet adapter

Ethernet support was enhanced in the following ways:

– TCP Segmentation Offload

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

You can enable or disable TCP large send on the SEA using the CLI **chdev** command. To enable it, use the -attr largesend=1 option. To disable it, use the -attr largesend=0 option. For example, the following command enables large send for Shared Ethernet Adapter ent1:

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

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

LPARs can now offload TCP segmentation to Virtual I/O Server Ethernet adapter, benefitting CPU intensive applications running on the LPAR.

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

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

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

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

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

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

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

Note: To modify large send on a physical Ethernet adapter, use the command option:
large_send=yes or large_send=no
To modify large send on a Shared Ethernet Adapter (SEA), use the command option:
largesend=1 or largesend=0

- Monitoring

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

- IVM enhancements

  Industry leading functions in this release include:

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

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

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

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

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

`mksp` Creates a storage pool, using the physical volumes represented by the physical volume parameter.

`chsp` Adds or removes physical volumes from a storage pool, or sets the default storage pool.

lssp	Lists information about storage pools.
mkbdsp	Attaches storage from a storage pool to a virtual SCSI adapter.
rmbdsp	Removes storage from a virtual SCSI adapter and returns it to the storage pool.

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

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

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

New features in Version 1.4 of the Virtual I/O Server

- Enablement for POWER6 System p
- Lightweight Directory Access Protocol client (LDAP) to facilitate centralized user management
- Simple Network Management Protocol (SNMP)
- nSeries and NetApp® storage subsystems
- iSCSI and Fibre Channel attachments
- SAS and SATA interconnects
- Integrated Virtual Ethernet (IVE)
- Virtual I/O Server CLI enhancements
- Added support for dynamic registration and deregistration of VLANs (Standard IEEE 802.1Q)

Client support for:

- Tivoli Storage Manager (TSM)
- IBM Tivoli Usage and Accounting Manager (ITUAM)
- Tivoli Application Dependency Discovery Manager (TADDM)
- IBM Tivoli Monitoring enhancements (ITM)

New CLI commands:

- save_base
- repphyvol
- su
- alt_root_vg
- cfgassist

New features in Version 1.5 of the Virtual I/O Server

Virtual I/O Server V1.5 delivers several key enhancements in the area of PowerVM. Visit this Web site for more information:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/download/fixpack.html

The following functions are enabled or enhanced for Virtual I/O Server 1.5:

- File-backed virtual SCSI devices. Provides additional flexibility for virtual SCSI device provisioning and management. In addition to backing a Virtual SCSI device (disk or optical) by physical storage, a virtual SCSI device can now be backed by a file. All virtual SCSI devices, including file-backed virtual SCSI devices, are accessed as standard SCSI-compliant LUNs.
- Virtual I/O Server installation media includes an expansion pack that delivers additional Virtual I/O Server Security functionality. These include Kerberos (Network Authentication Service for users and Client and Server Applications), LDAP (Lightweight Directory Access Protocol client functionality) and SNMP V3 (Simple Network Management Protocol Version 3).
- Support for POWER6 BladeCenter JS22.
- Partition Mobility enables System p users to move partitions easily from one POWER6 server to another. Partition Mobility is licensed via the optional PowerVM Enterprise Edition.
  - Live Partition Mobility:
    - Move a running LPAR and all of its running applications from one server to another with no application downtime.
    - Avoid application and partition downtime by moving running AIX and Linux partitions to other systems prior to upgrades/maintenance.
    - Balance resource utilization across multiple servers.
    - Limit the impact of planned server outages.

- Inactive Partition Mobility:
  - Move a powered-off LPAR from one system to another.
  - Move a partition and all virtual devices to the destination server, ready for activation.
- Partition Mobility requirements:
  - POWER6 servers and PowerVM Enterprise Edition.
  - Virtual I/O Server for virtualizing resources and providing mobility infrastructure on source and destination servers.
  - AIX/Linux partitions with OS levels supporting Partition Mobility.
  - HMC managing both source and destination servers or IVM-managed to IVM-managed server migrations supported.
  - Partitions can be diskless, dataless, or use SAN-attached storage via the Virtual I/O Server. Any virtual disks must be SAN-attached and accessed through Virtual I/O Server.
- Partition Mobility minimum requirements:
  - HMC V7R3.2.0, or later
  - System Firmware EM320_31 (for a 570), EL320_040 (for 520 and 550) or higher
  - AIX V5.3 TL7, or later, or AIX V6.1, or later
  - Virtual I/O Server V1.5, or later
  - Red Hat Enterprise Linux (RHEL) 5.1, or later
  - SUSE Linux Enterprise Server (SLES) 10 SP1

- The System Planning Tool (SPT) and the Workload Estimator are designed to ease the deployment of a virtualized infrastructure.

  The SPT consists of tools to assist clients with deploying a virtualized environment and is available as a no-charge download from:

  `http://www.ibm.com/systems/support/tools/systemplanningtool/`

  Use the SPT for designing System p and System i servers and partitions. The resulting design, represented by a System Plan, can then be imported onto HMC V5.2, or later, and the IVM, where via the new System Plan feature, clients can automate configuration of the partitions designed via the SPT. The System Plan feature of the HMC also allows generation of system plans via the **`mksysplan`** command.

  A major new feature in this version of the SPT is support for Virtual I/O Server resource planning, allowing clients to seamlessly size, plan, and deploy their Virtual I/O Server partitions.

Highlights of SPT include:

- Export system configuration to IBM for order processing via the IBM Sales Configurator (eConfig)
- Export system configuration to HMC or IVM for support of Power Architecture-based systems configuration
- Workload Estimator (WLE) integration to aid in sizing workloads
- Performance Monitor and WLE integration to aid in sizing workloads
- Conversion of HMC-generated system plans to obtain partition configuration, system units, and expansion units within SPT

Support for System p:

- Virtual I/O Server Resource Planning based on Performance Analysis and Workload Sizing Tool
- SPT integration with Performance Analysis and Sizing (PTX/WLE), which allows VIO simulation and adds validation of resulting configuration
- Virtual I/O Server LPAR Deployment, which automates deployment in a seamless manner via predefined, validated plan (SPT)
- Complex configurations that can be planned and deployed, including Redundancy, Shared Ethernet, SAN connections and Internal Storage, and MPIO attachments
- Make system plan, which can capture VIO configuration, enabling configuration recovery
- Deployment Planning and Configuration of IVM-managed partitions
- Automated configuration of partitions from SPT system plan via deployment wizard on the IVM partition

System Plan Viewer:

- Allows viewing and printing of the System Plan, which contains the hardware inventory and partition profile attributes for the system
- Allows viewing of sysplan report generated by mksysplan on HMC or IVM
- Provides reporting enhancements; report shows both HTML and text view, system diagram

For more information, visit

http://www-304.ibm.com/jct01004c/systems/support/tools/systemplanningtool

- Added new SMIT menus to Install and Update Virtual I/O Server software.
- Added support for large receive for SEA with new SEA attribute large_receive.

- Added support for the large receive feature for SEA.
- Added support to identify HEA and PCI-E type devices that can be used to create SEA using the **lsdev** command.
- Added support to invoke ssh and scp from the restricted shell by default.
- IBM Systems Director and IBM Tivoli Management Tools. The IBM Systems Director Family is a suite of products consisting of a no-charge base (IBM Director) and for-a-fee extensions. Software support is optionally available for a fee. Using IBM Director and IVM, System p clients can perform basic monitoring and management of their System p (as well as cross-platform) virtualized and nonvirtualized environments from a single console view. Using these tools, clients can view and perform base management across IBM Systems in their enterprise. More information about IBM Systems Director can be found at:

  http://www.ibm.com/systems/virtualization/systemsdirector/

  – ADDM provides deep-dive discovery for the System p servers (POWER5, or later), including its dependencies on the network and applications, along with its configuration data, subsystems, and virtualized LPARs. The discovery includes the HMC, Virtual I/O Server, IVM, System p servers, and each of the LPARs. For more information, visit: http://www.ibm.com/software/tivoli/products/taddm/

  – IBM Tivoli Monitoring

    IBM Tivoli Monitoring V6.1 now includes new System p agents, including an enhanced AIX monitoring agent, as well as new monitoring agents for the System p Central Electronics Complex, Virtual I/O Server, and HMC. These agents provide improved monitoring of System p performance, throughput, health, and availability, for both physical and LPARs. Monitoring of virtual environments on System p servers is greatly enhanced with this release. Best practices are built into these monitors to provide quicker time to value and improved manageability of System p servers. These enhancements deliver additional performance and throughput monitoring for AIX and the Virtual I/O Server beyond that previously provided by the IBM Tivoli Monitoring System Edition for System p V6.1 product, which is available *at no charge* to System p clients.

    Clients interested in adding these capabilities may purchase IBM Tivoli Monitoring V6.1 and, because both products use the same infrastructure, can upgrade from IBM Tivoli Monitoring System Edition for System p V6.1 with minimal effort.

    For further information, visit:

    http://www.ibm.com/software/tivoli/products/monitor-systemp/

To access or download the Readme document, you must sign in with your IBM ID and password, and then accept license terms.

Tivoli Identity Manager (TIM)

Virtual I/O Server is enabled for user administration from TIM.

When user administration is being managed in the enterprise using TIM, the client can use the Virtual I/O Server-specific TIM adapter (part of TIM software packages) to manage Virtual I/O Server-related users, allowing clients to manage Virtual I/O Server users centrally and apply enterprise-specific security policies to the Virtual I/O Server administrators and users. For more information refer to the TIM documentation.

TIM home page:

http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliIdentityManager.html

TIM 4.6 adapters:

http://www.ibm.com/support/docview.wss?rs=644&context=SSTFWV&uid=swg21246156&loc=en_US&cs=utf-8<=en

IVM V1.5 content

IVM adds support for new POWER6 functions and provides significant enhancements to existing functions.

- POWER6 feature support
  - PowerVM Live Partition Mobility support: IVM has full support for migrating an active partition between two physical IVM managed systems with no application downtime. The partition must be using shared SAN storage or be running in diskless / dataless mode. Live Partition Mobility is licensed via PowerVM Enterprise Edition.
  - Integrated Virtual Ethernet (also known as Host Ethernet Adapter), allows multiple partitions to share a single physical Ethernet port without the need for the Shared Ethernet Adapter bridge.
  - Support for Dedicated I/O: IVM now offers support for both shared and dedicated I/O, thus offering clients a choice when configuring their system.
  - Shared Dedicated Capacity allows for the donation of unused dedicated CPU cycles to the shared processor pool, thus increasing overall system performance.
  - Partition recovery priorities manages a processor failure, designed so that the partitions with the highest priority continue running.

- Processor compatibility mode allows the operating system to run in POWER5, POWER6, or POWER6 enhanced mode.

- Existing feature enhancements
  - IVM now supports assigning any number of virtual Ethernet adapters to partitions, assigning virtual Ethernet IDs, and full support for IEEE 802.1Q adapters. Virtual optical support has been extended to support loading and unloading of optical media files to a virtual optical device in the partition. Read-only support enables multiple partitions to access the same ISO image simultaneously. Read-write support allows partitions to treat the device as a DVD-RAM drive. In addition to the optical support, files may now be used for virtual disk in addition to physical and logical volumes.
  - The virtual terminal or console of a partition may now be accessed via a secure encrypted (SSH) connection directly from the Web browser.
  - User interface enhancements include items such as a more intuitive tasks layout and improved virtual Ethernet management.

3.2.2 Updating the Virtual I/O Server using fix packs

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

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

Fix packs are typically a general update intended for all Virtual I/O Server installations. Fix packs can be applied to either HMC-managed or IVM-managed Virtual I/O Servers. All interim fixes applied must be manually removed before applying any fix pack.

To check the latest release and instructions for the installation of Virtual I/O Server, visit:

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

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

3.3 Creating a Virtual I/O Server partition

This section describes the steps to create the Virtual I/O Server logical partition on the HMC and to install the Virtual I/O Server software.

3.3.1 Defining the Virtual I/O Server partition

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

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

Tip: If you do not have access to a NIM server, you can configure NIM in a partition temporarily for easy installation of partitions.

Figure 3-2 on page 154 shows the HMC with two attached managed systems. For our basic Virtual I/O Server configuration setup, we use the managed system named p570_170. This is a chosen name based on part of the serial number for easy identification on the HMC.

Note: The user interface of HMC V7 has changed from earlier versions. The new Web browser-based interface is intuitive. See also *Hardware Management Console V7 Handbook*, SG24-7491 for more information.

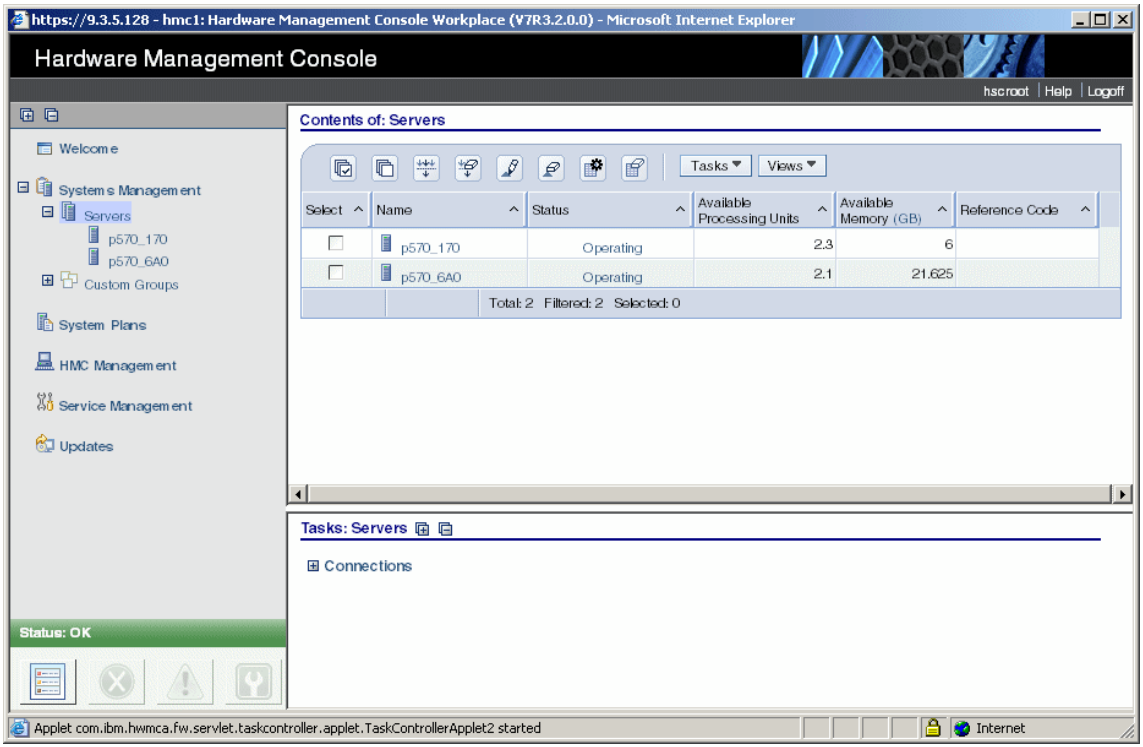


Figure 3-2 Hardware Management Console Server view

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

1. Select the managed system **p570_170**, then select **Configuration** → **Create Logical Partition** → **VIO Server** as shown in Figure 3-3, to start the Create Logical Partition Wizard.

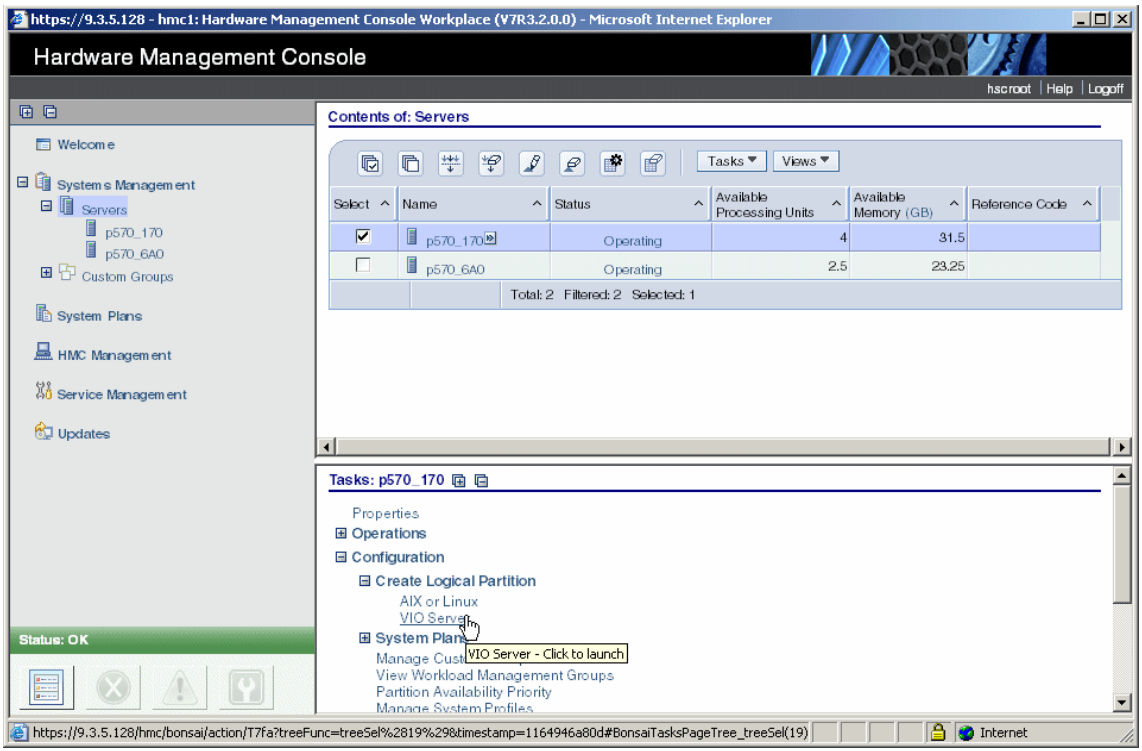


Figure 3-3 Starting the Create Logical Partition Wizard

Tip: You also have access to the Task area menus by clicking the button next to a selected item in the Contents area, as shown in Figure 3-4 on page 156.

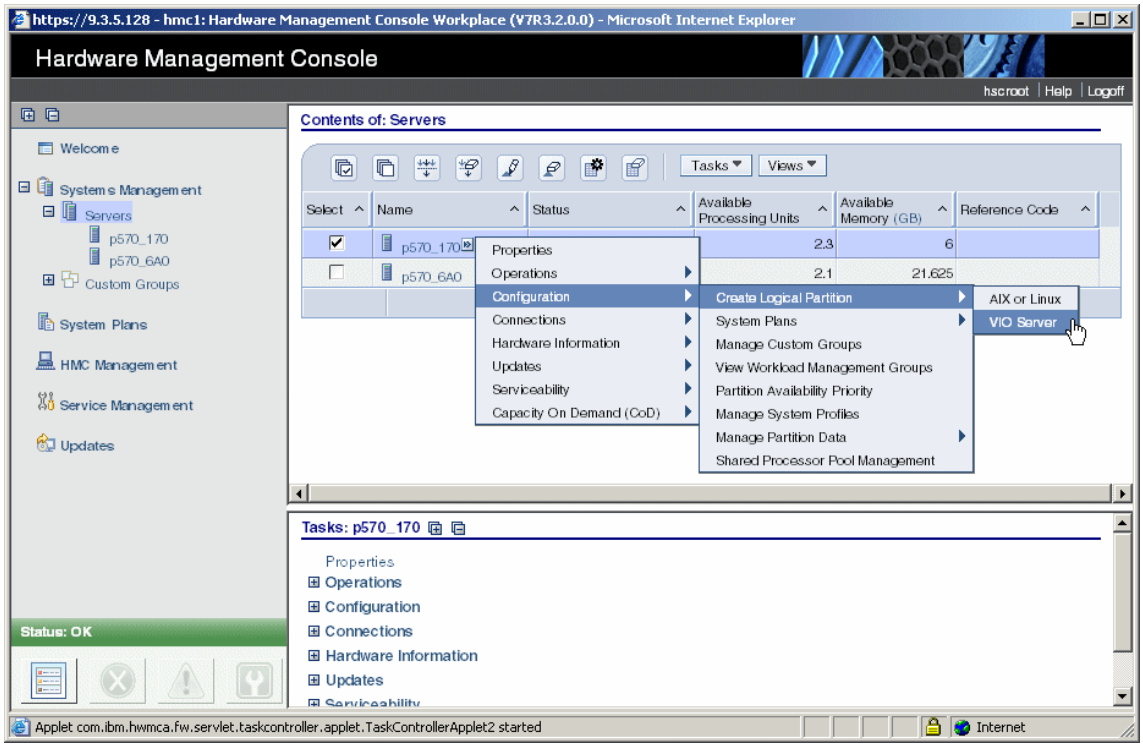


Figure 3-4 Using the pull-down menu in the HMC Contents area

2. Enter the partition name and ID as shown in Figure 3-5. (Or keep the ID selected by the HMC. IDs must be unique.) Check the **Mover service** box if the Virtual I/O Server partition to be created will be supporting Partition Mobility. Click **Next** to continue.

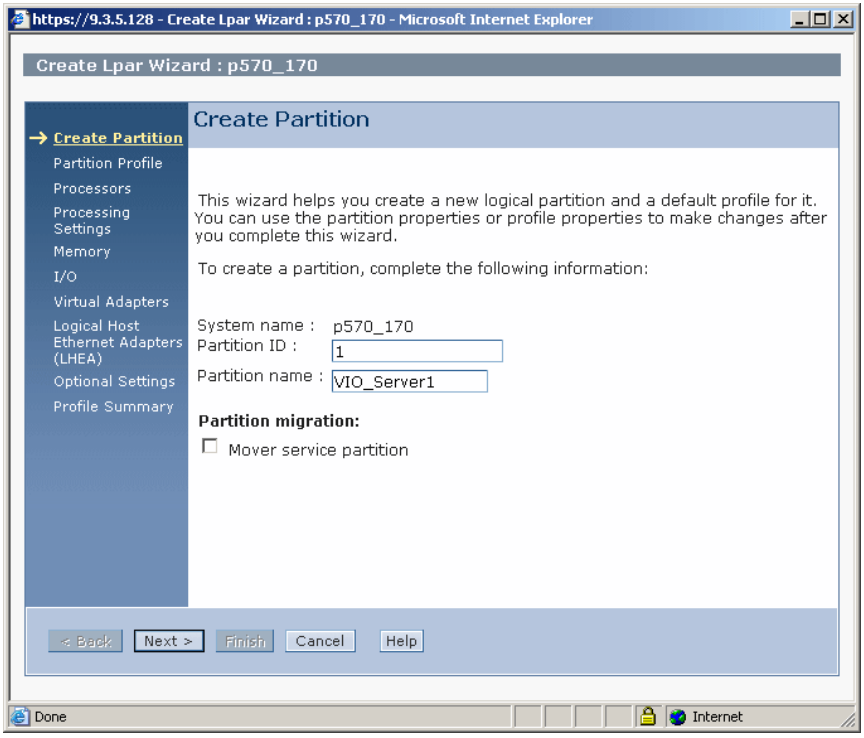


Figure 3-5 Defining the partition ID and partition name

3. Give the partition a profile name as shown in Figure 3-6. This becomes the default profile. A partition can have several profiles and you can change which profile is default. Click **Next** to continue.

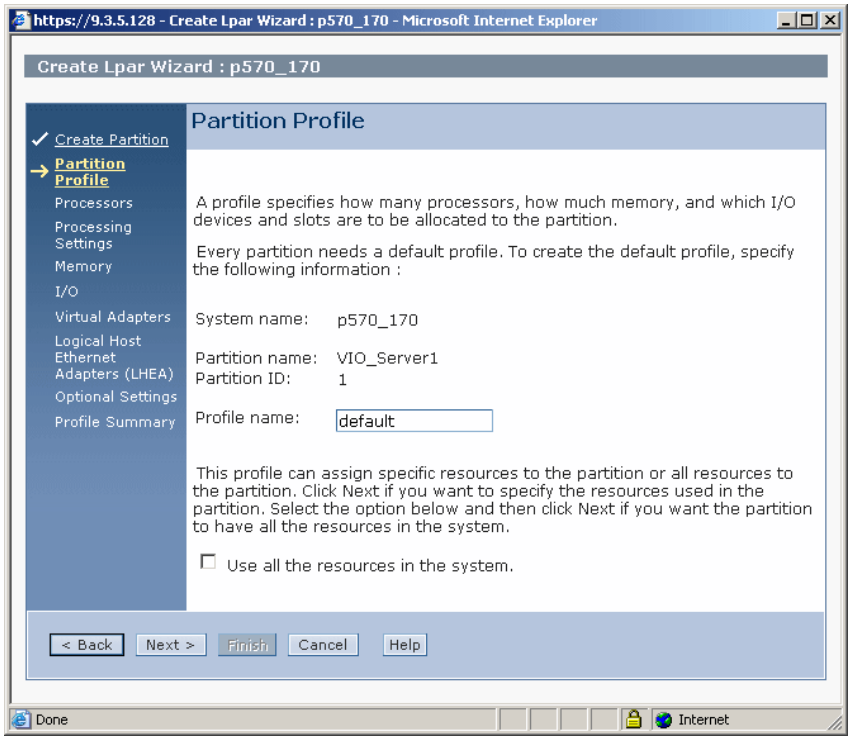


Figure 3-6 Naming the partition profile

Note: If the check box **Use all resources in the system** is checked, the logical partition being defined will get all the resources in the managed system and the partition will behave like a single server.

4. Select whether processors are to be part of a shared pool or dedicated for this partition. If shared is selected, it means that this will be a micro-partition. See Figure 3-7. Click **Next** to continue.

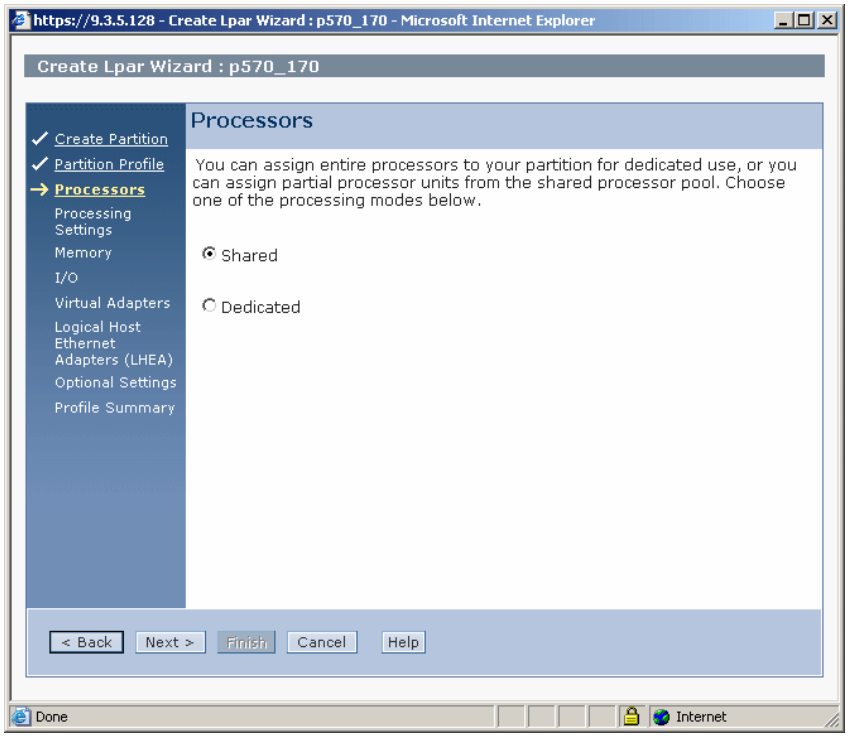


Figure 3-7 Select whether processors are to be shared or dedicated

5. Figure 3-8 shows the Processing Settings for micro-partitions. We increased the default weight of 128 to 191 since this is a Virtual I/O Server partition which should have priority.

   If you want to exploit the Multiple Shared Processor Pools capabilities and you have defined shared processor pools, you can specify here which pool this partition should belong to. See 4.12, “Configuring Multiple Shared-Processor Pools” on page 308.

   Click **Next** to continue.

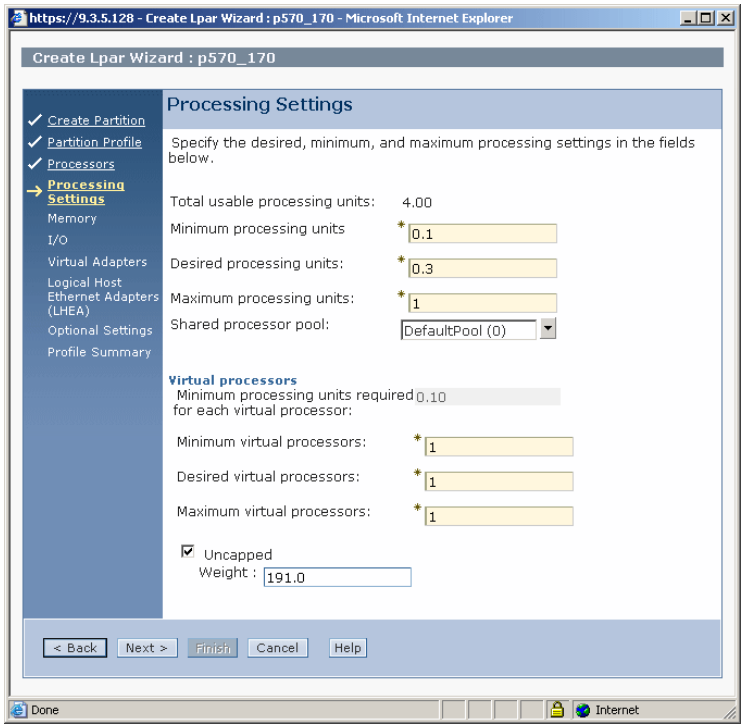


Figure 3-8 Processing Settings for a micro-partition

Note: The following rules apply to the processor settings:

- The system will try to allocate the desired values.
- The partition will not start if the managed system cannot provide the minimum amount of processing units.
- You cannot dynamically increase the amount of processing units to more than the defined maximum. If you want more processing units, the partition needs to be stopped and reactivated in order to read the updated profile (not just rebooted).
- The maximum number of processing units cannot exceed the total Managed System processing units.

Note: Refer to 2.3, “Overview of Micro-Partitioning technologies” on page 33 for more information about processing units, capped and uncapped mode, and virtual processors.

6. Choose the memory settings, as shown in Figure 3-9. Click **Next** to continue.

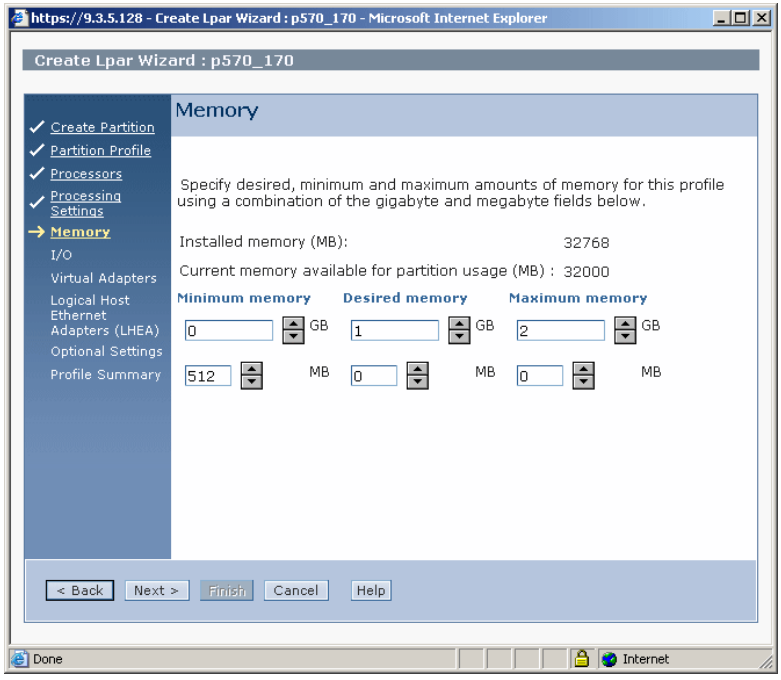


Figure 3-9 Memory setting

Note: The following rules apply to Figure 3-9:

- The system will try to allocate the desired values.
- If the managed system is not able to provide the minimum amount of memory, the partition will not start.
- You cannot dynamically increase the amount of memory in a partition to more than the defined maximum. If you want more memory than the maximum, the partition needs to be stopped and the profile updated and then restarted.
- The ratio between minimum amount of memory and maximum cannot be more than 1/64.

7. Select the physical I/O adapters for the partition as shown in Figure 3-10 on page 163. Required means that the partition will not be able to start unless these are available to this partition. Desired means that the partition can start also without these adapters.

   Click **Add as required**.

   We are creating a Virtual I/O Server partition, which in our case requires a fiber channel adapter to attach SAN disks for the client partitions. It also requires an Ethernet adapter for Shared Ethernet adapter bridging to external networks. The SAS adapter is attached to the internal disks on this POWER6 system. Click **Next** to continue.

   The latest editions of the VIOS require a minimum of 30 GB of disk space to store the installed contents of the installation media.

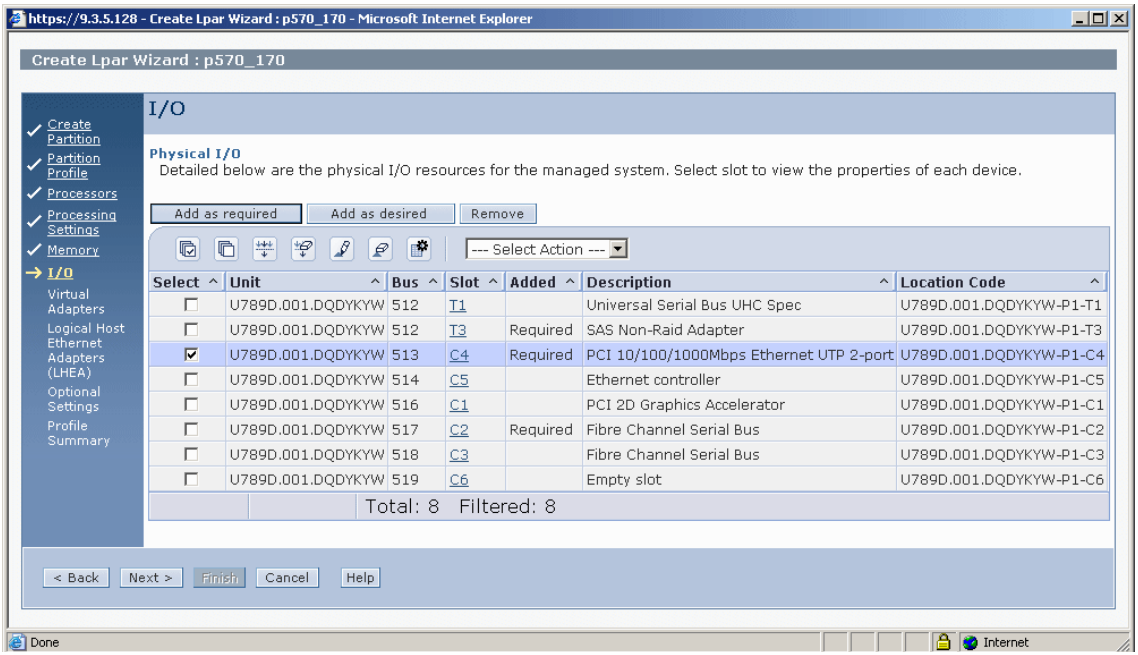


Figure 3-10 Selecting physical adapters for the partition

Note: The adapters for the Virtual I/O Server are set to `required` since they are needed to provide I/O to the client partitions. A required adapter cannot be moved in a dynamic LPAR operation.
To change the setting from *required* to *desired* for an adapter you have to change the profile, stop, and restart the partition.

Notes:

a. Do not set the adapter (if separate adapter) that holds the DVD to `required`, as it may be moved in a dynamic LPAR operation later.

b. The installed Ethernet adapter in this system is a 2-port adapter. Both ports are owned by the same partition.
In general, all devices attached to an adapter are owned by the partition that holds the adapter.

c. Virtualization will usually reduce the required number of physical adapters and cables.

d. If possible, use hot-plug Ethernet adapters for the Virtual I/O Server for increased serviceability.

8. Create virtual Ethernet and SCSI adapters. The start menu is shown in Figure 3-11. We increased the maximum number of virtual adapters to 100 to allow for a flexible numbering scheme.

Important: The default serial adapters are required for console login from the HMC. Do not change these.

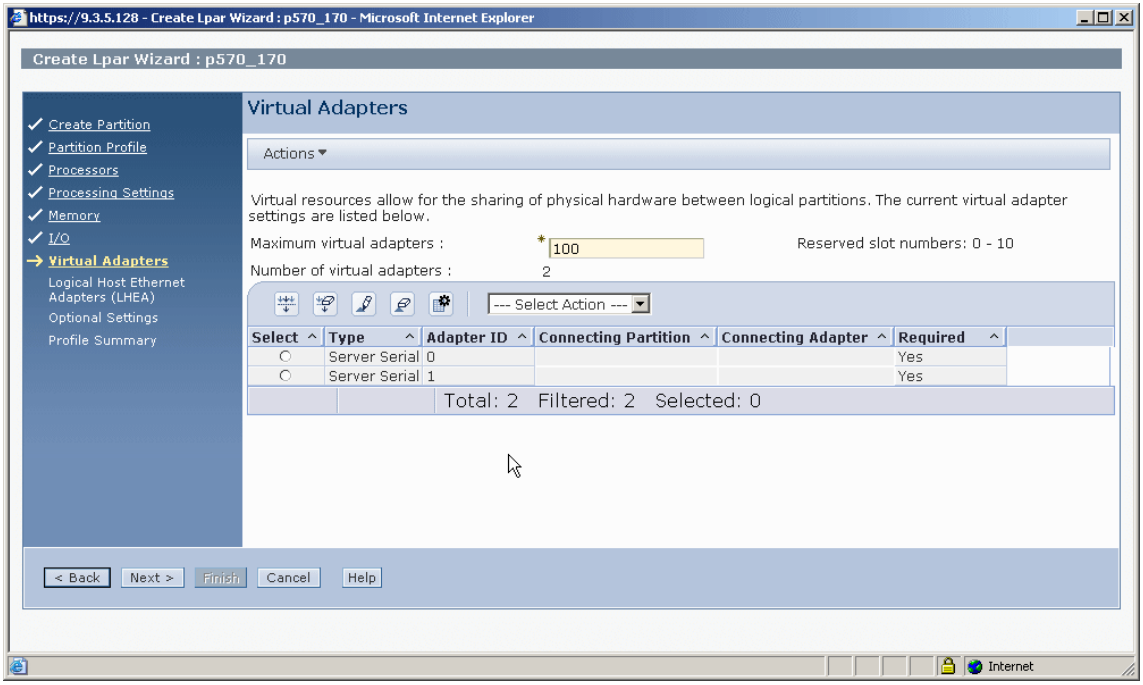


Figure 3-11 The start menu for creating virtual adpters.

Note: The maximum number of adapters must not be set above 1024.

9. Right-click **Actions** and select **Create** → **Ethernet Adapter** as shown in Figure 3-12 to open the Create Virtual Ethernet Adapter window.

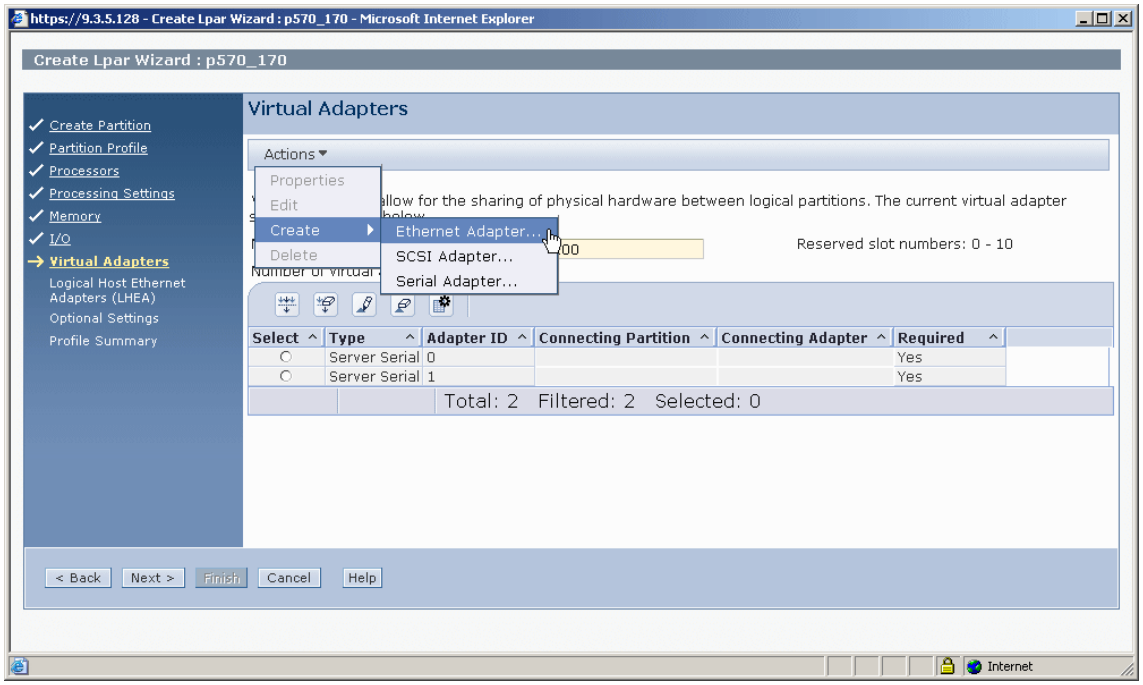


Figure 3-12 Selecting to create a virtual Ethernet adapter

10. Virtual Ethernet adapter is a logical adapter that emulates the function of a physical I/O adapter in a logical partition. Virtual Ethernet adapters enable communication to other logical partitions within the managed system without using physical hardware and cabling. A Virtual I/O Server is only required for communication to an external network. Input the Adapter ID (in this case 11) and a VLAN ID (in this case1) as shown in Figure 3-13. IEEE 802.1q is not needed here since VLAN tagging is not used. Select the **Access External network** check box to use this adapter as a gateway between an internal and an external network. This virtual Ethernet will be configured as part of a shared Ethernet adapter. You can select the **IEEE 802.1Q compatible adapter** check box if you want to add additional virtual LAN IDs.

    Click **OK** when finished.

    You can create more adapters if you need more networks.

    Note: For more information on Trunk Priority, see 4.3.1, “Shared Ethernet Adapter failover” on page 252.

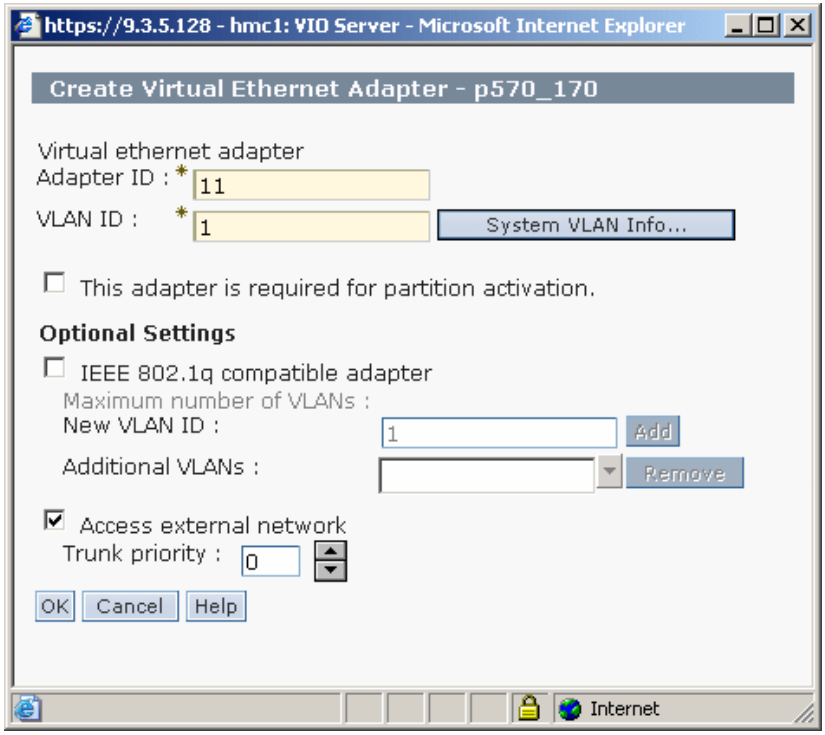


Figure 3-13 Creating Virtual Ethernet adapter

Notes:

- Adapter ID and slot ID are used interchangeably.
- Selecting the **Access External Networks** check box makes sense only for a Virtual I/O Server partition. Do not select this flag when configuring the client partitions virtual Ethernet adapters.

Tip: You can create an additional virtual Ethernet adapter for the Virtual I/O Server if you prefer to configure the IP address on a separate adapter instead of the SEA.

11. Right-click **Actions** and select **Create** → **SCSI Adapter** to open the Create Virtual SCSI Adapter window shown in Figure 3-14.

    Create a server adapter to be used by the virtual optical device (CD or DVD). Adapter number 90 is used here but you can use any unique number that fits your configuration. Note that this adapter is set to Any client partition can connect. Click **OK** when finished. This adapter is to make things easier from a system management point of view.

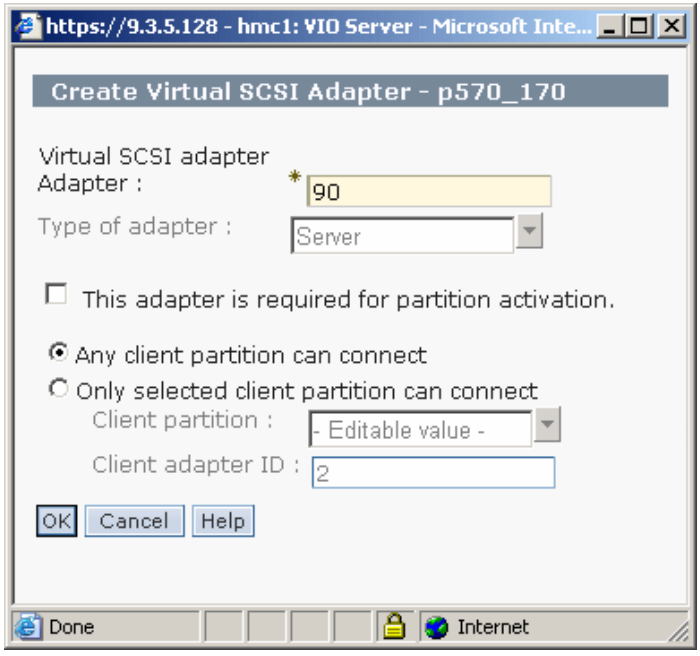


Figure 3-14 Creating the SCSI server adapter for the DVD

Note: For virtual adapters it is not necessary to check the box. This adapter is required for partition activation.

12. Right-click **Actions** and select **Create** → **SCSI Adapter** to open the Create Virtual SCSI Adapter window shown in Figure 3-15 to create additional virtual SCSI adapters for client partition disks. We will create 4 client partitions later so we need 4 server adapters with slot numbers 20, 30, 40, 50. Client slot is set to 21 for all clients.

    At this stage, the clients are not known to the HMC. If you create the SCSI server adapters now, you will have to specify the partition ID in the Client Adapter field or specify that **Any client can connect**, which means you will have to change this after you have created the clients. We plan to have partition ID of 2, 3, 4 and 5 (the Virtual I/O Server is 1). The HMC will use the partition names when the client partitions have been created. Click **OK** when finished.

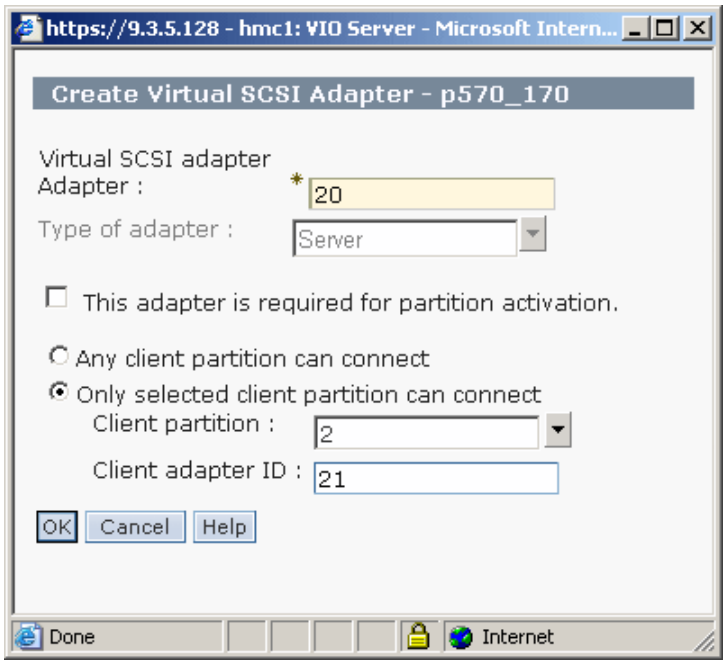


Figure 3-15 SCSI server adapter for the NIM_server

13. Repeat step 12 to create the SCSI server adapters for the rest of the client partitions according to Figure 3-25 on page 179. Page 170 shows the list of created virtual adapters and their slot numbers. Click **Next** to continue.

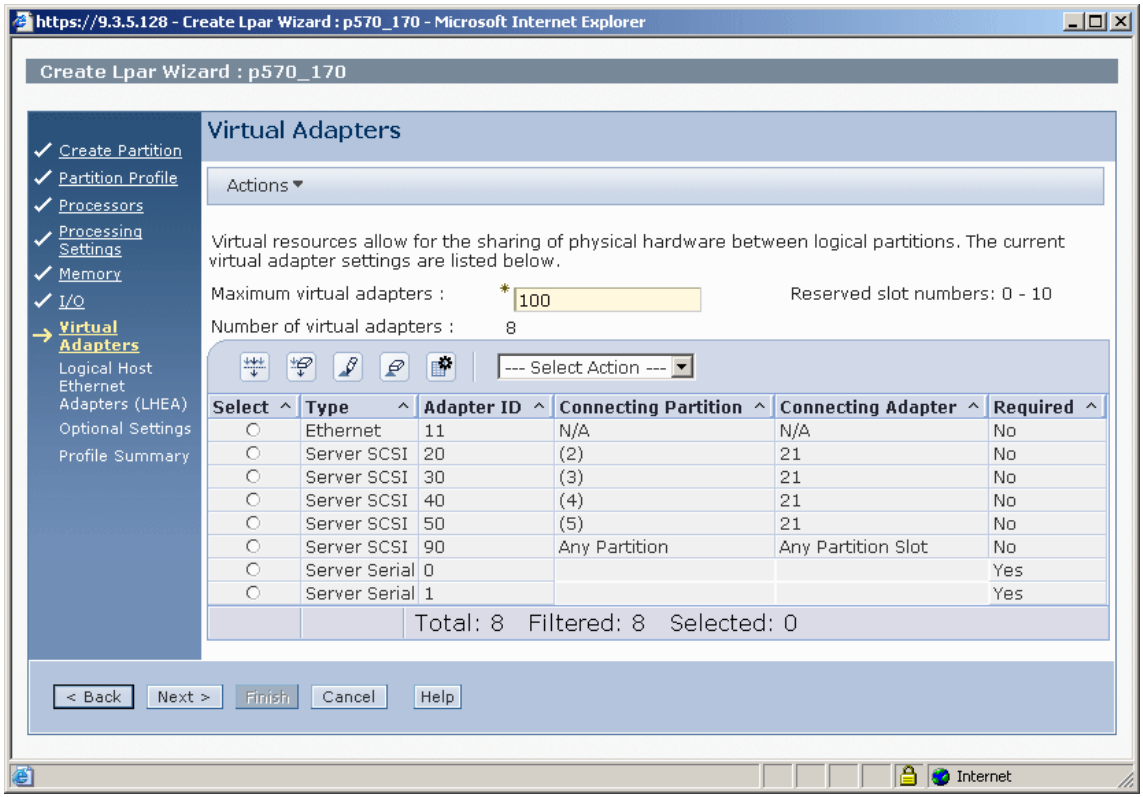


Figure 3-16 List of created virtual adapters

14. The Host Ethernet Adapter, HEA, is a new offering on the POWER6 system. It replaces the integrated Ethernet ports on selected systems and can provide logical Ethernet ports directly to the client partitions without using the Virtual I/O Server. We will not use any of these ports for the Virtual I/O Server. For information about HEA, see *Integrated Virtual Ethernet Adapter Overview and Introduction,* REDP-4340-00 on:

    http://www.redbooks.ibm.com

    Click **Next** to continue.

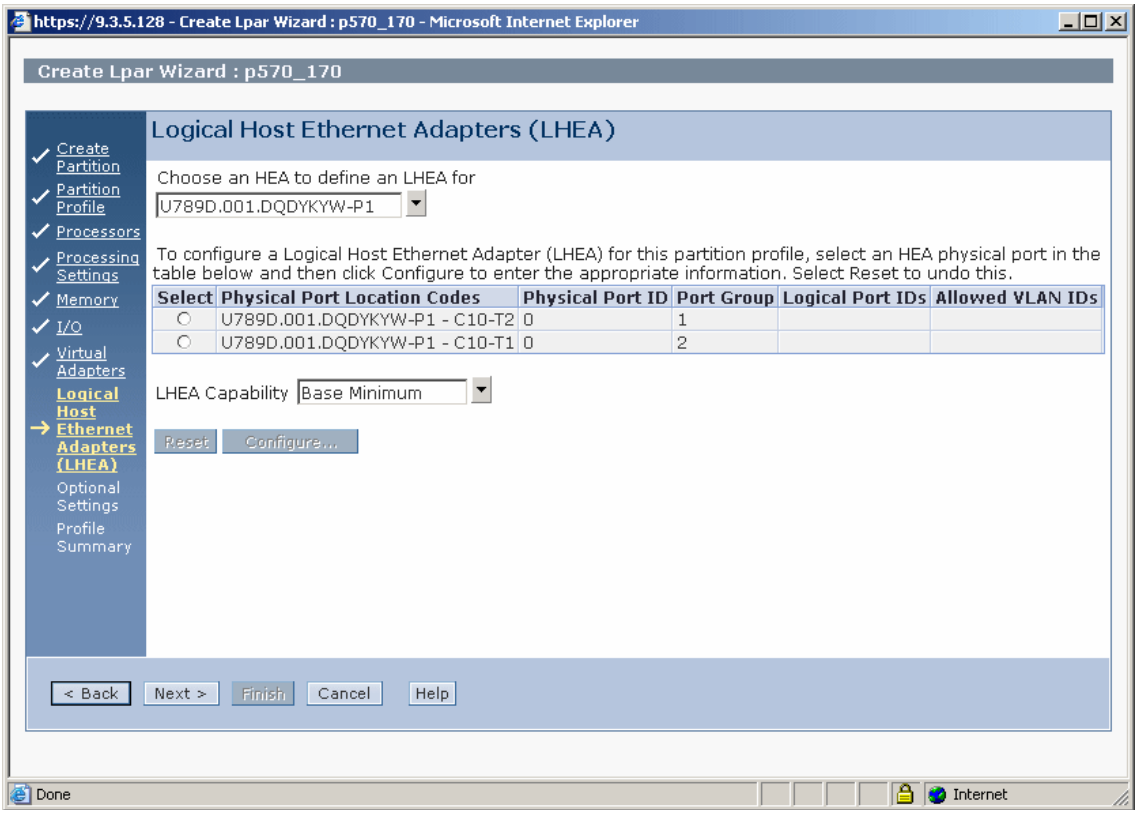


Figure 3-17 Menu for creating Logical Host Ethernet Adapters

15. Choose Optional Settings for the partition. The partition will boot in normal mode by default. “Enable connection monitoring” will alert any drop in connection to the HMC. Automatic start with managed system means that the partition will start automatically when the system is powered on with the “Partition auto start” option (selected at power-on). “Enable redundant error path reporting” allows for call-home error messages to be sent through the private network in case of open network failure. Click **Next** to continue.

> **Important:** Enable redundant error path reporting must *not* be set for partitions that will be moved using Partition Mobility.

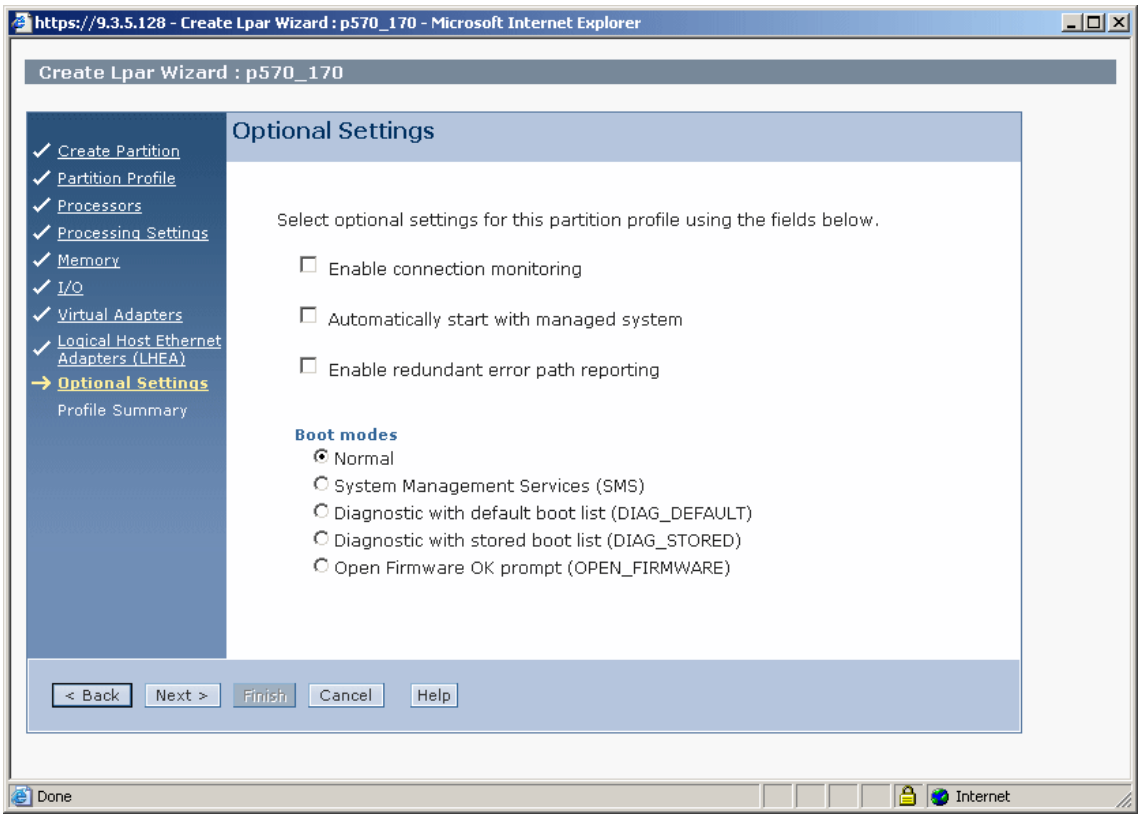


Figure 3-18 Optional Settings menu

16. The Profile Summary menu shows details about the partition. You can check details about the I/O devices by clicking **Details**, or click **Back** to go back and modify any of the previous settings. Click **Finish** to complete the profile.

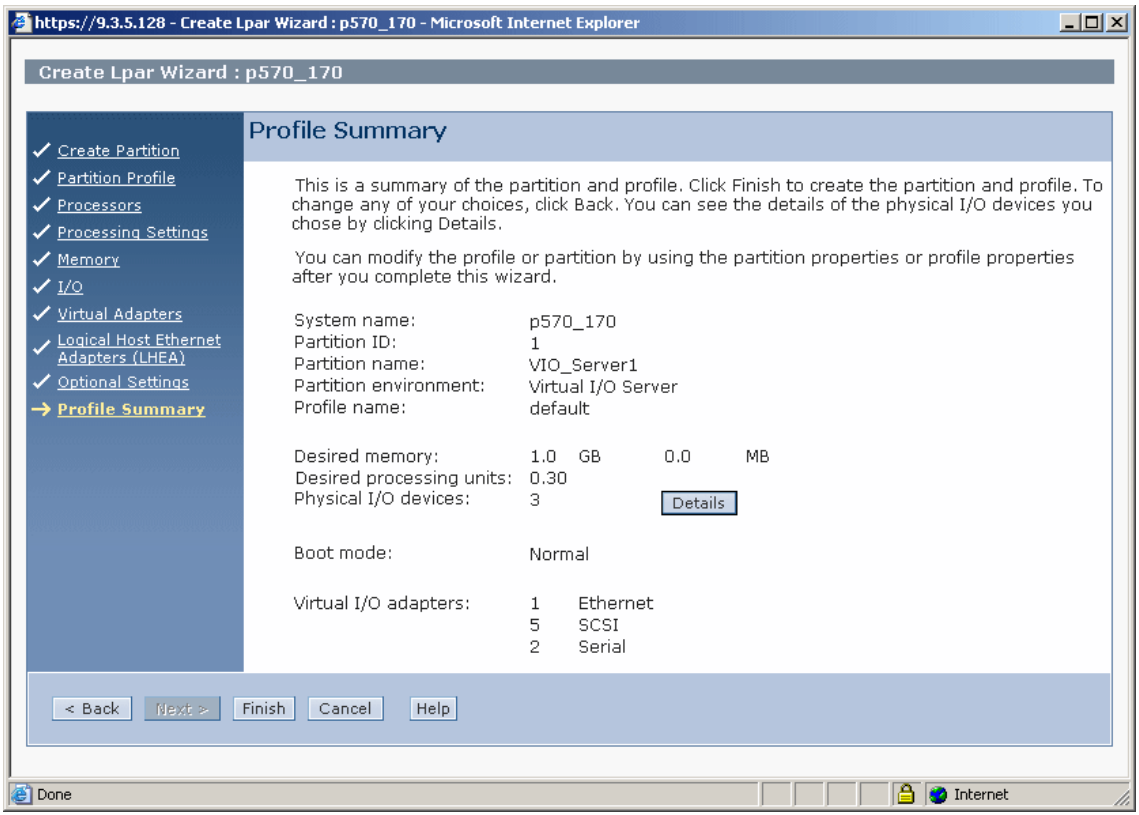


Figure 3-19 The Profile Summary menu

Figure 3-20 shows the created partition on the HMC.

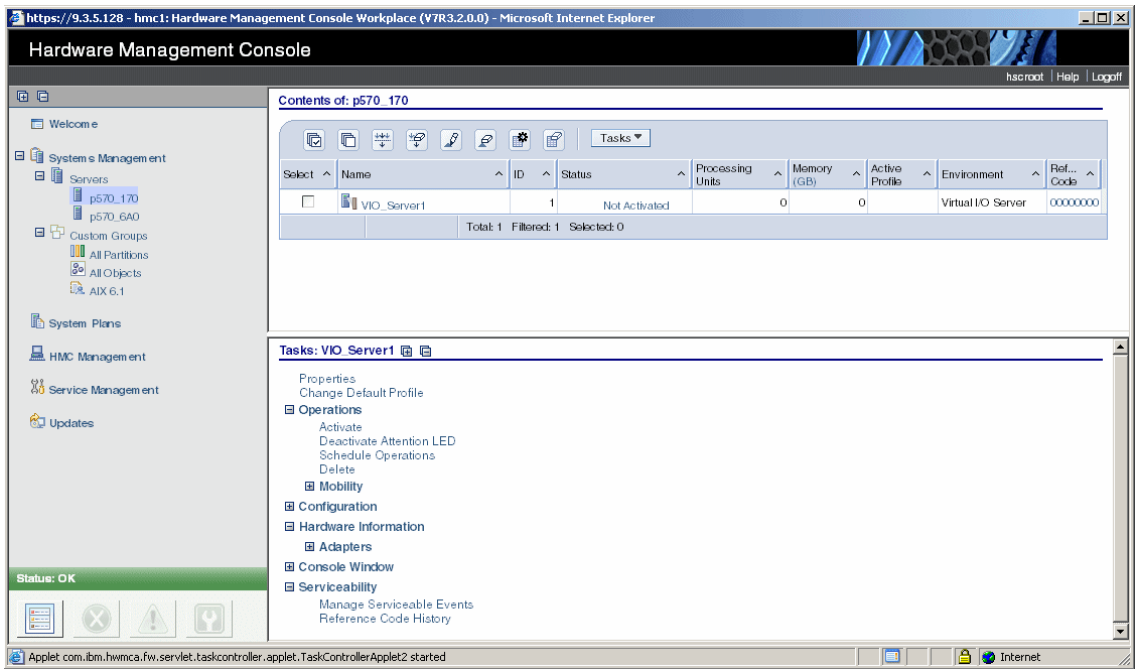


Figure 3-20 The created partition VIO_Server1

3.4 Virtual I/O Server software installation

This section describes the installation of the Virtual I/O Server Version 1.5 software into the previously created Virtual I/O partition named VIO_Server1. There are three supported methods of installing the Virtual I/O Server software:

- Using the DVD allocated to the Virtual I/O Server partition and booting from it.
- Installing the Virtual I/O Server software from the HMC using the **`installios`** command, which is using NIM for a network installation. If you just enter **`installios`** without any flags, a wizard will be invoked and the you will be prompted to interactively enter the information contained in the flags. The default is to use the optical drive on the HMC for the Virtual I/O Server installation media, but you can also specify a remote file system instead. See the HMC Information Center for details on how to use **`installios`** from the HMC at:

  `http://publib.boulder.ibm.com/infocenter/systems/index.jsp`

Note: A network adapter with connection to the HMC network is required for the Virtual I/O Server installation.

- When installing the media using NIM, the **installios** command is also available in AIX both for the NIM server and any NIM client. If you run the **installios** command on a NIM client, you are prompted for the location of the bos.sysmgt.nim.master fileset. The NIM client is then configured as a NIM master. Use the following link and search for **installios** for additional information:

  http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp

Tip: If you plan on using two Virtual I/O Servers (described in 4.1, “Providing continuous availability for virtual I/O” on page 244), you could install the first server, apply updates, multipath drivers and customization, then make a NIM backup and use the image for installing the second Virtual I/O Server.

Take Note: The architecture of POWER processor-based systems may allow swapping an optical media device used for installs between partitions by reassigning the Other SCSI Controller. The disks on these systems may be on their own SCSI Controller. On several systems, the media devices and internal storage may be part of a single “SAS Controller” group, or simply share a controller, preventing you from separating the install device and a subset of the internal disks.

With this in mind, there are two approaches to install a second VIOS.

- NIM install of the second VIOS from an AIX system using the **installios** command.
- Assign the install media to the first VIOS and install on the SAS disks that do not have the optical device as part of the group. Then unassign the installed disks and install on the disks that are part of the group.

Note: The VIOS has a minimum disk requirement of 30 GB.

The following steps show the installation using the optical install device:

1. Place the Virtual I/O Server DVD disk in the drive of the System p server.

2. Activate the VIO_Server1 partition by selecting the partition and the clicking on **Activate**, as shown in Figure 3-21. Select the default profile you used to create this partition.

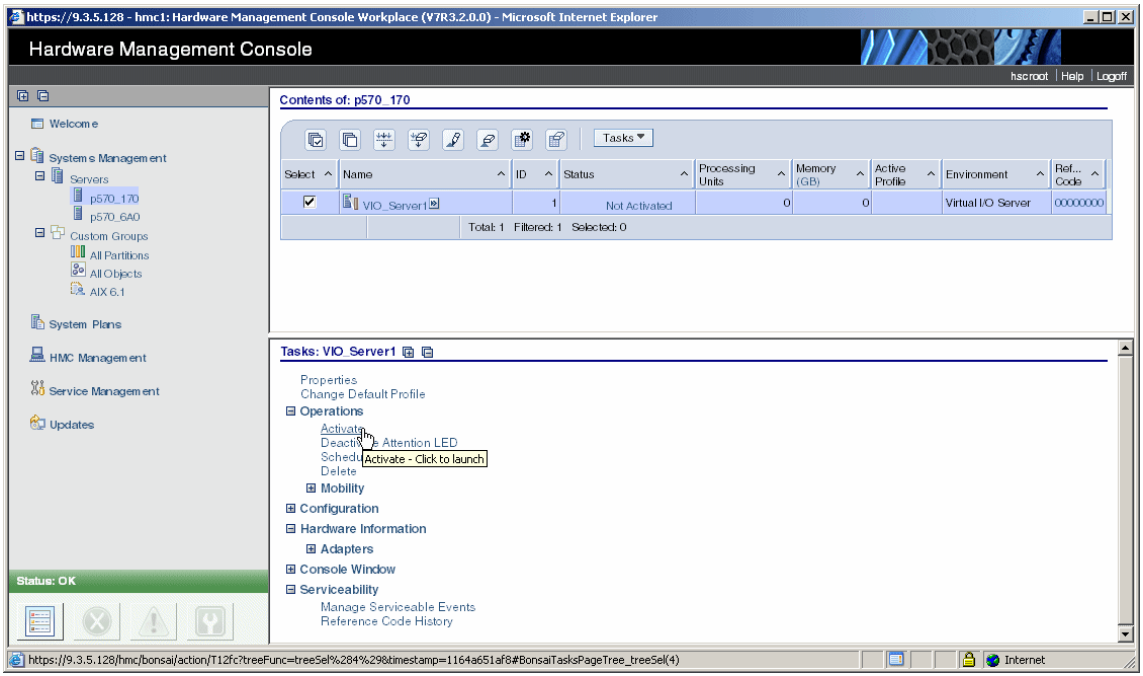


Figure 3-21 Activating a partition

3. Select the VIO_Server1 profile and then check the **Open a terminal window or console session** check box, as shown in Figure 3-22, and then click **Advanced**.

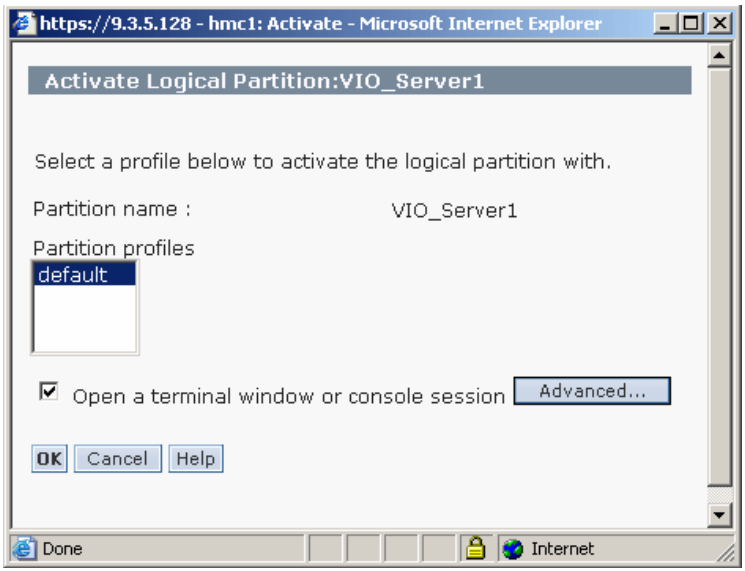


Figure 3-22 The Activate Logical Partition submenu

4. Under the Boot Mode drop-down list, choose **SMS**, as shown in Figure 3-23, and then click **OK**.

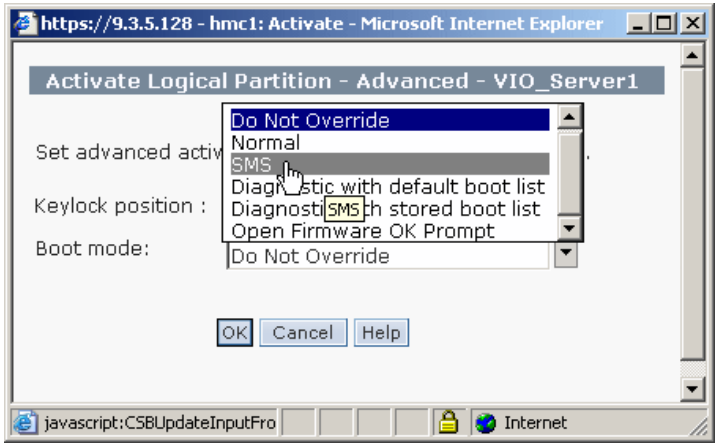


Figure 3-23 Selecting the SMS menu for startup

5. Figure 3-24 shows the SMS menu after booting the partition in SMS mode.

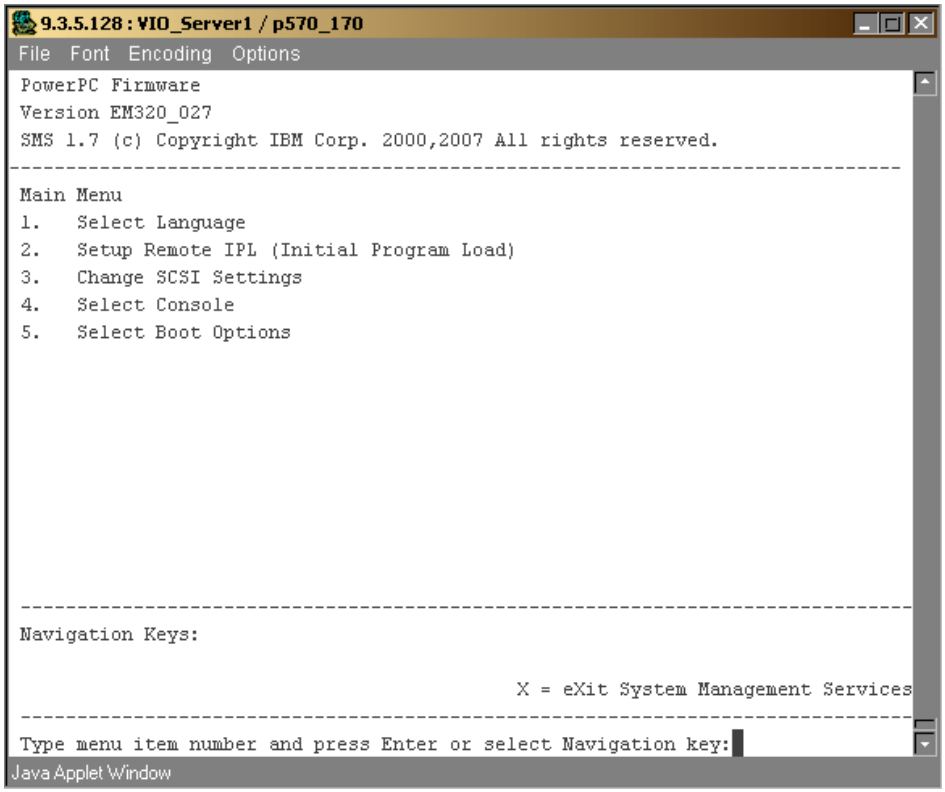


Figure 3-24 The SMS startup menu

6. Follow the following steps to continue and boot the Virtual I/O Server partition. The process is similar to installing AIX:
   a. Choose 5 for Select Boot Options and then press Enter.
   b. Choose 1 for Select Install/Boot Device and then press Enter.
   c. Choose 9 to list all devices, look for the CD-ROM (press n to get to next page if required), enter the number for the CD_ROM and then press Enter.

   Tip: It is useful to list all devices to check that the required devices are available. Only adapters will be visible at this point unless underlaying disks have a bootdevice on them.

   d. Choose 2 for Normal Mode Boot and then press Enter.
   e. Confirm your choice with 1 for Yes and then press Enter.

f. Next you are prompted to accept the terminal as console and then to select installation language.

g. You are then presented with the installation menu. It is recommended to check the settings (option 2) before proceeding with the installation. Check if the selected installation disk is correct.

Visit the IBM Systems Information Center for more information on the installation process:

http://publib.boulder.ibm.com/infocenter/systems/index.jsp

7. When the installation procedure has finished, use the padmin user name to log in. Upon initial login, you will be asked to supply the password. There is no default password.

After logging in successfully you will be placed under the Virtual I/O Server command line interface (CLI).

Enter a (and press Enter) to accept the Software Maintenance Agreement terms, then type in the following command to accept the license:

```
$ license -accept
```

You are now ready to use the newly installed Virtual I/O Server software.

3.5 Basic Virtual I/O Server scenario

In this section, a simple configuration consisting of a single Virtual I/O Server partition servicing virtual SCSI devices and a single network to four logical partitions is shown. Refer to Figure 3-25 on page 179 for the basic configuration scenario.

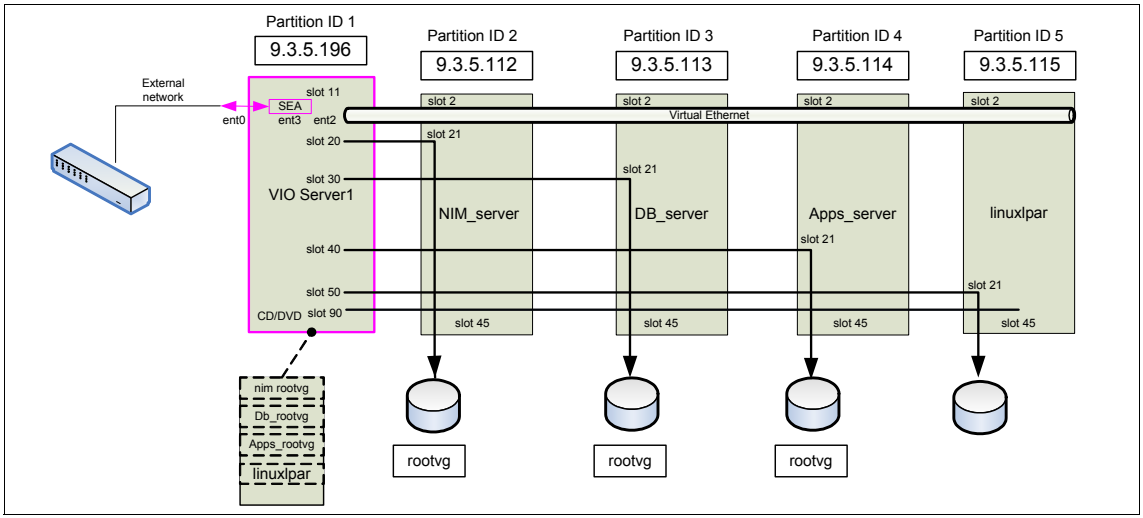


Figure 3-25 Basic Virtual I/O Server scenario

Based on Figure 3-25, the following sections will guide you through configuring client partitions, including setting up virtual CD/DVD.

3.5.1 Creating a Shared Ethernet Adapter

To create a Shared Ethernet adapter, perform the following steps:

1. Open a terminal window on the HMC.
2. Use the **lsdev** command on the Virtual I/O Server to verify that the Ethernet trunk adapter (adapter that had the **Access external network** box checked) is available (see Example 3-4).

Example 3-4 Check for virtual Ethernet adapter

```
$ lsdev -virtual
name             status     description
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
vasi0            Available  Virtual Asynchronous Services Interface
(VASI)
vhost0           Available  Virtual SCSI Server Adapter
vhost1           Available  Virtual SCSI Server Adapter
vhost2           Available  Virtual SCSI Server Adapter
vhost3           Available  Virtual SCSI Server Adapter
vhost4           Available  Virtual SCSI Server Adapter
vsa0             Available  LPAR Virtual Serial Adapter
```

3. Select the appropriate physical Ethernet adapter that will be used to create the Shared Ethernet Adapter. The **lsdev** command will show a list of available physical adapters (see Example 3-5). In this case it is ent0.

Example 3-5 Check for physical Ethernet adapter

```
$ lsdev -type adapter
name            status     description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
fcs0            Available  4Gb FC PCI Express Adapter (df1000fe)
fcs1            Available  4Gb FC PCI Express Adapter (df1000fe)
sissas0         Available  PCI-X266 Planar 3Gb SAS Adapter
vasi0           Available  Virtual Asynchronous Services Interface (VASI)
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vhost3          Available  Virtual SCSI Server Adapter
vhost4          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

You can use the **lsmap -all -net** command to check the slot numbers of the virtual Ethernet adapters. We use ent2 in slot 11 (see Example 3-6).

Example 3-6 Checking slot numbers

```
$ lsmap -all -net
SVEA   Physloc
------ --------------------------------------------
ent2   U9117.MMA.101F170-V1-C11-T1

SEA                 NO SHARED ETHERNET ADAPTER FOUND
```

4. Use the **mkvdev** command to create a new ent3 device as the Shared Ethernet Adapter. ent0 will be used as the physical Ethernet adapter and ent2 as the virtual Ethernet adapter (Example 3-7 on page 181). The -default flag specifies the default virtual adapter to be used by the SEA for non-VLAN-tagged packets. For more information on Virtual I/O Server commands you find the Virtual I/O Server Command Reference at:

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

   Note: Any interface with an IP address on the adapters used when defining the SEA must be detached.

Example 3-7 Create Shared Ethernet Adapter

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1
ent3 Available
en3
et3
```

5. Confirm that the newly created Shared Ethernet Adapter is available using the **lsmap -all -net** command (Example 3-8).

Example 3-8 Confirm that the Shared Ethernet Adapter is available

```
$ lsmap -all -net
SVEA   Physloc
------ --------------------------------------------
ent2   U9117.MMA.101F170-V1-C11-T1

SEA                  ent3
Backing device       ent0
Status               Available
Physloc              U789D.001.DQDYKYW-P1-C4-T1
```

The Shared Ethernet Adapter will form a bridge, allowing communication between the inter-partition VLAN and the external network.

We used the following values for our scenario (Table 3-1).

Table 3-1 Network settings

Settings	Value
hostname	VIO_Server1
IP-address	9.3.5.196
netmask	255.255.254.0
gateway	9.3.4.1

6. Use the **cfgassist** command (see Figure 3-26) or the **mktcpip** command (see Example 3-9) to configure the interface on the SEA, ent3.

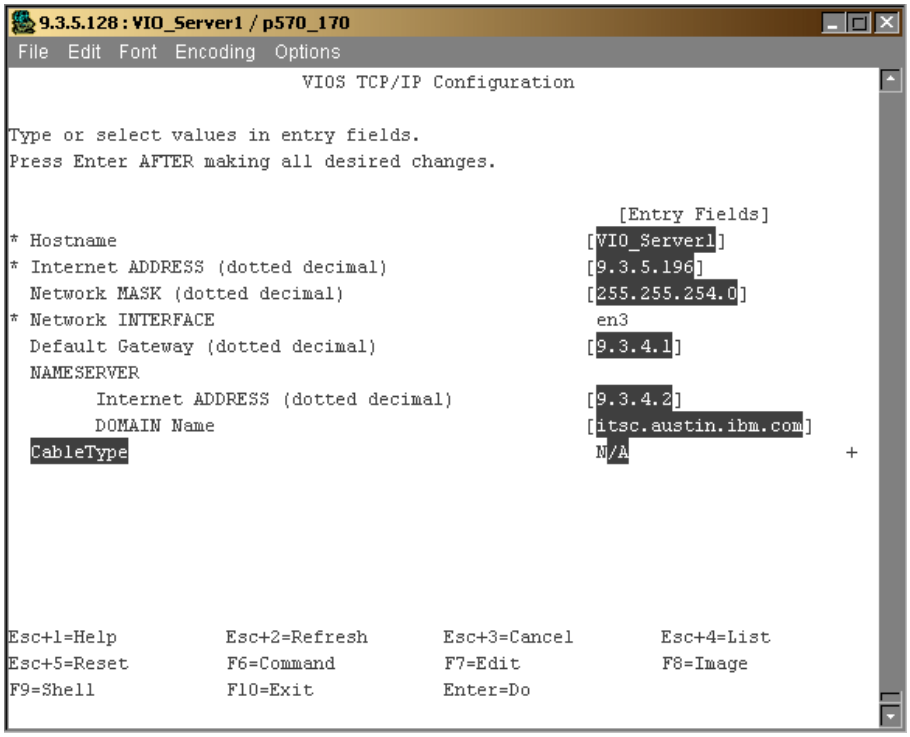


Figure 3-26 Setting TCP/IP parameters using the cfgassist command

Note: There is no performance penalty for adding the IP address to the SEA interface instead of keeping it on a separate virtual Ethernet adapter. However, the SEA can be redefined without having to detach the interface when the interface is kept on a separate virtual Ethernet adapter. This provides increased serviceability.

Example 3-9 Defining IP settings on the created SEA

```
$ mktcpip -hostname VIO_Server1 -inetaddr 9.3.5.196 -interface en3\
-netmask 255.255.254.0 -gateway 9.3.4.1
```

3.5.2 Defining virtual disks

Virtual disks can either be whole physical disks, logical volumes, or files. The physical disks can either be local disks or SAN attached disks. SAN disks can be

used both for the Virtual I/O Server rootvg (using SAN boot) and for virtual I/O client disks.

Tip: A virtual disk, physical volume, can be mapped to more than one partition by using the -f option of the **mkvdev** command for the second mapping of the disk. See 4.14.3, “Concurrent disks in client partitions” on page 345.

Use the following steps to build the logical volumes required to create the virtual disk for the client partition’s rootvg based on our basic scenario using logical volumes:

1. Log in with the padmin user ID and run the **cfgdev** command to rebuild the list of visible devices used by the Virtual I/O Server.

   The virtual SCSI server adapters are now available to the Virtual I/O Server. The name of these adapters will be vhost*x,* where x is a number assigned by the system.

2. Use the **lsdev -virtual** command to make sure that your 5 new virtual SCSI server adapters are available, as shown in Example 3-10.

   Example 3-10 Listing virtual devices

```
$ lsdev -virtual
name             status     description
ent2             Available  Virtual I/O Ethernet Adapter (l-lan)
vasi0            Available  Virtual Asynchronous Services Interface
(VASI)
vhost0           Available  Virtual SCSI Server Adapter
vhost1           Available  Virtual SCSI Server Adapter
vhost2           Available  Virtual SCSI Server Adapter
vhost3           Available  Virtual SCSI Server Adapter
vhost4           Available  Virtual SCSI Server Adapter
vsa0             Available  LPAR Virtual Serial Adapter
ent3             Available  Shared Ethernet Adapter
```

3. Use the **lsmap -all** command to check slot numbers and vhost adapter numbers as shown in Example 3-11:

   Example 3-11 Verifying slot numbers and vhost adapter numbers

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.MMA.101F170-V1-C20                    0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND

SVSA            Physloc                                     Client Partition ID
```

```
--------------- ------------------------------------------ ------------------
vhost1          U9117.MMA.101F170-V1-C30                   0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND

SVSA            Physloc                                    Client Partition ID
--------------- ------------------------------------------ ------------------
vhost2          U9117.MMA.101F170-V1-C40                   0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND

SVSA            Physloc                                    Client Partition ID
--------------- ------------------------------------------ ------------------
vhost3          U9117.MMA.101F170-V1-C50                   0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND

SVSA            Physloc                                    Client Partition ID
--------------- ------------------------------------------ ------------------
vhost4          U9117.MMA.101F170-V1-C90                   0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND
```

If the devices are not available, then there was a problem defining them. You can use the **rmdev -dev vhost0 -recursive** command for each device and then reboot the Virtual I/O Server if needed. Upon reboot, the configuration manager will detect the hardware and recreate the vhost devices. Also check the profile on the HMC.

Using logical volumes

In our basic scenario, we will create the volume group named rootvg_clients on hdisk2 and partition it into logical volumes to serve as boot disks to our client partitions.

Important: We do not recommend using the Virtual I/O Server rootvg disk for virtual client disks (logical volumes).

1. Create a volume group rootvg_clients on hdisk2 using the **mkvg** command, as shown in Example 3-12.

Example 3-12 Creating the rootvg_clients volume group

```
$ mkvg -f -vg rootvg_clients hdisk2
rootvg_clients
```

2. Define all the logical volumes that are going to be presented to the client partitions as hdisks using the **mklv** command. In our case, these logical volumes will be our rootvg for the client partitions (see Example 3-13).

Example 3-13 Create logical volumes

```
$ mklv -lv dbsrv_rvg rootvg_clients 10G
rootvg_dbsrv
$ mklv -lv appssrv_rvg rootvg_clients 10G
rootvg_apps
$ mklv -lv nimsrv_rvg rootvg_clients 10G
rootvg_nim
$ mklv -lv linux rootvg_clients 2G
rootvg_lnx
```

3. Define the SCSI mappings to create the virtual target device that associates to the logical volume you have defined in the previous step. Based on Example 3-14, we have four virtual host devices on the Virtual I/O Server. These vhost devices are the ones we are going to map to our logical volumes. Adapter C4 is the adapter for the virtual DVD. See 2.2.1, “POWER Hypervisor virtual processor dispatch” on page 28 for details on virtual optical devices.

Example 3-14 Create virtual device mappings

```
$ lsdev -vpd|grep vhost
vhost4 U9117.MMA.101F170-V1-C90 Virtual SCSI Server Adapter
vhost3 U9117.MMA.101F170-V1-C50 Virtual SCSI Server Adapter
vhost2 U9117.MMA.101F170-V1-C40 Virtual SCSI Server Adapter
vhost1 U9117.MMA.101F170-V1-C30 Virtual SCSI Server Adapter
vhost0 U9117.MMA.101F170-V1-C20 Virtual SCSI Server Adapter

$ mkvdev -vdev nimsrv_rvg -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev dbsrv_rvg -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev appssrv_rvg -vadapter vhost2 -dev vappssrv_rvg
vappssrv_rvg Available
$ mkvdev -vdev linux -vadapter vhost3 -dev vlinux
vlinux Available
$ mkvdev -vdev cd0 -vadapter vhost4 -dev vcd
vcd Available

$ lsdev -virtual
name            status     description
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vasi0           Available  Virtual Asynchronous Services Interface
(VASI)
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
```

```
vhost3          Available  Virtual SCSI Server Adapter
vhost4          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
vappssrv_rvg    Available  Virtual Target Device - Logical Volume
vcd             Available  Virtual Target Device - Optical Media
vdbsrv_rvg      Available  Virtual Target Device - Logical Volume
vlinux          Available  Virtual Target Device - Logical Volume
vnimsrv_rvg     Available  Virtual Target Device - Logical Volume
ent3            Available  Shared Ethernet Adapter
```

Tip: It is useful to give the virtual device a name using the **-dev** flag with the **mkvdev** command for easier identification.

Note: Based on the **lsdev -vpd** command, the mappings exactly correspond to the slot numbering we intended (refer to Figure on page 179). For example, the vhost0 device is slot number 20 (U9117.MMA.101F170-V1-C20) on the Virtual I/O Server, which is then being shared to the NIM_server partition. The NIM_server partition has its virtual SCSI device slot set to 21. This is for easy association between virtual SCSI devices on the server and client side.

4. Use the **lsmap** command to ensure that all logical connections between newly created devices are correct, as shown in Example 3-15.

Example 3-15 Checking mappings

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.MMA.101F170-V1-C20                    0x00000000

VTD                   vnimsrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        nimsrv_rvg
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost1          U9117.MMA.101F170-V1-C30                    0x00000000

VTD                   vdbsrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        dbsrv_rvg
Physloc
```

```
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost2          U9117.MMA.101F170-V1-C40                    0x00000000

VTD                   vappssrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        appssrv_rvg
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost3          U9117.MMA.101F170-V1-C50                    0x00000000

VTD                   vlinux
Status                Available
LUN                   0x8100000000000000
Backing device        linux
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost4          U9117.MMA.101F170-V1-C90                    0x00000000

VTD                   vcd
Status                Available
LUN                   0x8100000000000000
Backing device        cd0
Physloc               U789D.001.DQDYKYW-P4-D1
```

Tips:

1. The same concept applies when creating virtual disks that are going to be used as data volume groups.
2. You can map several disks through the same client-server adapter pair.

The mapped virtual disks will now appear to client partitions as SCSI disks.

Using whole disks

SAN disks can be assigned to the Virtual I/O Server and mapped to client partitions as whole disks in appropriate sizes for the clients.

Notes:

- When whole disks are going to be mapped, they cannot belong to a volume group.
- Both local disks and SAN disks can be mapped to client partitions as whole disks.
- It is recommended to map whole LUNs when using SAN disks rather than to split a LUN into separate logical volumes.

You can verify the available hdisks with the **lspv** command as shown in Example 3-16. We will use the newly defined hdisk6 to 9.

Example 3-16 Listing hdisks

```
$ lspv
NAME            PVID                                VG              STATUS
hdisk0          00c1f170d7a97dec                    rootvg          active
hdisk1          00c1f170e170ae72                    None
hdisk2          00c1f170e170c9cd                    None
hdisk3          00c1f170e170dac6                    None
hdisk4          00c1f170e170e91a                    None
hdisk5          00c1f170e170fbb2                    None
hdisk6          none                                None
hdisk7          none                                None
hdisk8          none                                None
hdisk9          none                                None
hdisk10         00c1f17064cbb91d                    None
```

It is useful to be able to map the LUNs to hdisk numbers in a SAN environment. The **fget_config -Av** command is provided on the IBM DS4000 series for a listing of LUN names, as shown in Example 3-17.

This command is part of the storage device driver and you will have to use the **oem_setup_env** command to access it. Use the same SCSI Server adapters as with the logical volumes.

Tip: A similar function is provided on DS8000 and DS6000 with the **lssdd** command.

Example 3-17 Listing of LUN to hdisk mapping

```
$ oem_setup_env
# fget_config -Av

---dar1---
```

```
User array name = 'FAST200'
dac0 ACTIVE dac1 ACTIVE

Disk     DAC   LUN Logical Drive
utm              1
hdisk6   dac1    0 1
hdisk7   dac0    2 2
hdisk8   dac0    3 4
hdisk9   dac1    4 3
hdisk10  dac1    5 5
```

You can also find the LUN number for an hdisk with the **lsdev -dev hdisk*n* -vpd** command, where *n* is the hdisk number as shown in Example 3-18.

Example 3-18 Finding the LUN number

```
$ lsdev -dev hdisk7 -vpd
  hdisk7
U789D.001.DQDYKYW-P1-C2-T1-W200500A0B8110D0F-L2000000000000  3542
(200) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:  block
```

These are the steps to map whole disks in the same way as in the previous section:

1. You can use the **lsdev -vpd** command to list the virtual slot numbers corresponding to vhost numbers as in Example 3-19.

   Example 3-19 Listing of slot number to vhost mapping

   ```
   $ lsdev -vpd|grep vhost
   vhost4 U9117.MMA.101F170-V1-C90 Virtual SCSI Server Adapter
   vhost3 U9117.MMA.101F170-V1-C50 Virtual SCSI Server Adapter
   vhost2 U9117.MMA.101F170-V1-C40 Virtual SCSI Server Adapter
   vhost1 U9117.MMA.101F170-V1-C30 Virtual SCSI Server Adapter
   vhost0 U9117.MMA.101F170-V1-C20 Virtual SCSI Server Adapter
   ```

2. Define the SCSI mappings to create the virtual target devices that associate to the logical volume you have defined in the previous step. Based on Example 3-14 on page 185, we have four virtual host devices on the Virtual

I/O Server. These vhost devices are the ones we are going to map to our disks. We also add the virtual DVD and call it vcd. See Example 3-20.

Example 3-20 Mapping SAN disks and the DVD drive, cd0

```
$ mkvdev -vdev hdisk6 -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev hdisk7 -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev hdisk8 -vadapter vhost2 -dev vappssrv_rvg
vappssrv_rvg Available
$ mkvdev -vdev hdisk9 -vadapter vhost3 -dev vlinux
vlinux Available
$ mkvdev -vdev cd0 -vadapter vhost4 -dev vcd
vcd Available
```

Tips:

1. The same concept applies when creating disks that are to be used as data disks.
2. You can map data disks through the same vhost adapters that are used for rootvg. VSCSI connections operate at memory speed and each adapter can handle a large number of target devices.

You are now ready to install AIX or Linux in each of the partitions. The disks should be available for client partitions to use at this point.

3.5.3 Virtual SCSI optical devices

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

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

The setup steps are as follows:

1. Let the DVD drive be assigned to a Virtual I/O Server.

2. Create a server SCSI adapter using the HMC where any partition can connect as shown in Figure 3-14 on page 167.

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

3. Run the **cfgdev** command to get the new vhost adapter. You can find the new adapter number with the **lsdev -virtual** command.
4. In the Virtual I/O Server, VIO_Server, you create the virtual device with the following command:

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

   where n is the number of the vhost adapter. See Example 3-21.

   Example 3-21 Making the virtual device for the DVD drive

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

5. Create a client SCSI adapter in each LPAR using the HMC. The client adapter should point to the server adapter created in the previous step. In our basic setup we used slot 90 for the server adapter and slot 45 for all client adapters.

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

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

Moving the drive:

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

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

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

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

Notes:

- You can also use the virtual CD or DVD to install an LPAR when selected in the SMS startup menu, provided the drive is not assigned to another LPAR.
- In IVM, the optical device is moved using the graphical user interface.

You can use the **dsh** command to find the LPAR currently holding the drive, as shown in Example 3-22. **dsh** is installed by default in AIX. You can use **dsh** with **rsh**, **ssh** or Kerberos authentication as long as **dsh** can run commands without being prompted for a password.

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

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

Example 3-22 Finding which LPAR is holding the optical drive using dsh

```
# dsh lsdev -Cc cdrom|dshbak
HOST: Apps_server
---------------
cd0 Available  Virtual SCSI Optical Served by VIO Server
HOST: DB_server
-------------
cd0 Defined  Virtual SCSI Optical Served by VIO Server
HOST: NIM_server
---------
cd0 Defined  Virtual SCSI Optical Served by VIO Server
```

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

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

Or use the **ssh** command. See Example 3-23 on page 193.

Example 3-23 Finding which LPAR is holding the optical drive using ssh

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

Tip: You can also find the Partition ID of the partition holding the drive from the **`lsmap -all`** command on the Virtual I/O Server.

Note: AIX6 offers a new graphical interface to system management called IBM Systems Console for AIX. This has a menu setup for dsh.

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

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

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

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

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

Note: Care must be taken not to unconfigure recursively when production disks share the same adapter as the optical drive.

1. Release the drive from the partition holding it.
2. Unconfigure the virtual device in the Virtual I/O Server.
3. Unconfigure the PCI or SAS adapter recursively.

4. Use the HMC to move the adapter to the target partition.
5. Run the **cfgmgr** command (or the **cfgdev** command for a Virtual I/O Server partition) to configure the drive.
6. When finished, remove the PCI adapter recursively
7. Use the HMC to move the adapter back.
8. Run the **cfgmgr** command (or the **cfgdev** command for a Virtual I/O Server partition) to configure the drive.

Use the **cfgdev** command on the Virtual I/O Server to reconfigure the drive when it is reassigned to the original partition in order to make it available as a virtual optical drive again. See Example 3-24 for unconfiguring and configuring the drive (disregard the error message from our test system).

Example 3-24 Unconfiguring and reconfiguring the DVD drive

```
$ rmdev -dev vcd -ucfg
vcd Defined
$ lsdev -slots
# Slot                     Description       Device(s)
U787B.001.DNW108F-P1-C1    Logical I/O Slot  pci3 ent0
U787B.001.DNW108F-P1-C3    Logical I/O Slot  pci4 fcs0
U787B.001.DNW108F-P1-C4    Logical I/O Slot  pci2 sisioa0
U787B.001.DNW108F-P1-T16   Logical I/O Slot  pci5 ide0
U9113.550.105E9DE-V1-C0    Virtual I/O Slot  vsa0
U9113.550.105E9DE-V1-C2    Virtual I/O Slot  ent1
U9113.550.105E9DE-V1-C3    Virtual I/O Slot  ent2
U9113.550.105E9DE-V1-C4    Virtual I/O Slot  vhost0
U9113.550.105E9DE-V1-C20   Virtual I/O Slot  vhost1
U9113.550.105E9DE-V1-C22   Virtual I/O Slot  vhost6
U9113.550.105E9DE-V1-C30   Virtual I/O Slot  vhost2
U9113.550.105E9DE-V1-C40   Virtual I/O Slot  vhost3
U9113.550.105E9DE-V1-C50   Virtual I/O Slot  vhost4
$ rmdev -dev pci5 -recursive -ucfg
cd0 Defined
ide0 Defined
pci5 Defined

$ cfgdev
Method error (/usr/lib/methods/cfg_vt_optical -l vcd ):
$ lsdev -virtual
name            status
description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
```

```
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vhost3          Available  Virtual SCSI Server Adapter
vhost4          Available  Virtual SCSI Server Adapter
vhost6          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
apps_rootvg     Available  Virtual Target Device - Disk
db_rootvg       Available  Virtual Target Device - Disk
linux_lvm       Available  Virtual Target Device - Disk
nim_rootvg      Available  Virtual Target Device - Disk
vcd             Available  Virtual Target Device - Optical Media
vtscsi0         Available  Virtual Target Device - Logical Volume
ent3            Available  Shared Ethernet Adapter
```

3.5.4 Mirroring the Virtual I/O Server rootvg

Once the installation of the Virtual I/O Server is complete, the following commands can be used to mirror the Virtual I/O Server rootvg volume group to a second physical volume. The following steps show how to mirror the Virtual I/O Server rootvg:

1. Use the **extendvg** command to include hdisk1 as part of the rootvg volume group. The same LVM concept applies; you cannot use an hdisk that belongs to another volume group and the disk needs to be of equal size or greater.
2. Use the **lspv** command, as shown in Example 3-25, to confirm that rootvg has been extended to include hdisk2.

Example 3-25 lspv command output before mirroring

```
$ extendvg -f rootvg hdisk2
0516-1162 extendvg: Warning, The Physical Partition Size of 128
requires the creation of 2235 partitions for hdisk2. The limitation for
volume group rootvg is 1016 physical partitions per physical volume.
Use chvg command with -t option to attempt to change the maximum
Physical Partitions per Physical volume for this volume group.
0516-792 extendvg: Unable to extend volume group.

$ chvg -factor 6 rootvg
0516-1164 chvg: Volume group rootvg changed.  With given
characteristics rootvg can include upto 5 physical volumes with 6096
physical partitions each.
$ extendvg -f rootvg hdisk2
$
```

```
$ lsp
NAME            PVID                                VG
STATUS
hdisk0          00c1f170d7a97dec                    rootvg
active
hdisk1          00c1f170e170ae72                    rootvg_clients
active
hdisk2          00c1f170e170c9cd                    None
hdisk3          00c1f170e170dac6                    None
```

3. Use the **mirrorios** command to mirror the rootvg to hdisk1, as shown in Example 3-26. With the -f flag, the **mirrorios** command will automatically reboot the Virtual I/O Server partition.

Note: SAN disks are usually RAID protected in the storage subsystem. If you use a SAN disk for the rootvg of the Virtual I/O Server, mirroring may not be required.

Example 3-26 Mirroring the Virtual I/O Server rootvg volume group

```
$ extendvg rootvg hdisk2

$ lspv
NAME            PVID                                VG
STATUS
hdisk0          00c1f170d7a97dec                    rootvg
active
hdisk1          00c1f170e170ae72                    rootvg_clients
active
hdisk2          00c1f170e170c9cd                    rootvg
active
hdisk3          00c1f170e170dac6                    None

$ mirrorios -f hdisk2
SHUTDOWN PROGRAM
Fri Nov 23 18:35:34 CST 2007
0513-044 The sshd Subsystem was requested to stop.

Wait for 'Rebooting...' before stopping.
```

4. Check if logical volumes are mirrored and if the normal boot sequence has been updated, as shown in Example 3-27 on page 197. Both mirrored boot devices should appear in the bootlist, when correctly configured.

Example 3-27 Logical partitions are mapped to two physical partitions

```
$ lsvg -lv rootvg
rootvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT
POINT
hd5                 boot       1       2       2    closed/syncd  N/A
hd6                 paging     4       8       2    open/syncd    N/A
paging00            paging     8       16      2    open/syncd    N/A
hd8                 jfs2log    1       2       2    open/syncd    N/A
hd4                 jfs2       2       4       2    open/syncd    /
hd2                 jfs2       23      46      2    open/syncd    /usr
hd9var              jfs2       5       10      2    open/syncd    /var
hd3                 jfs2       18      36      2    open/syncd    /tmp
hd1                 jfs2       80      160     2    open/syncd    /home
hd10opt             jfs2       6       12      2    open/syncd    /opt
lg_dumplv           sysdump    8       8       1    open/syncd    N/A

$ bootlist -mode normal -ls
hdisk0 blv=hd5
hdisk2 blv=hd5
```

3.5.5 Creating client partitions

This section shows you how to create the four client partitions for our basic Virtual I/O scenario. The definitions are similar to the creation of our Virtual I/O Server partition, but instead of selecting **Virtual I/O**, choose **AIX** or **Linux**. For the client partitions we are creating, refer to Figure 3-3 on page 155. Follow these steps to create the client partitions:

1. Restart the Create Logical Partition Wizard, as you did at the beginning of the previous process 3.3, “Creating a Virtual I/O Server partition” on page 153.

Enter the name of the partition as shown in Figure 3-27. Note that Partition ID is 2. This ID was used when the server SCSI adapters were created. Once you have defined the Partition name (as shown on next page), the HMD will update the definitions of the server SCSI adapters to add that Partition name.

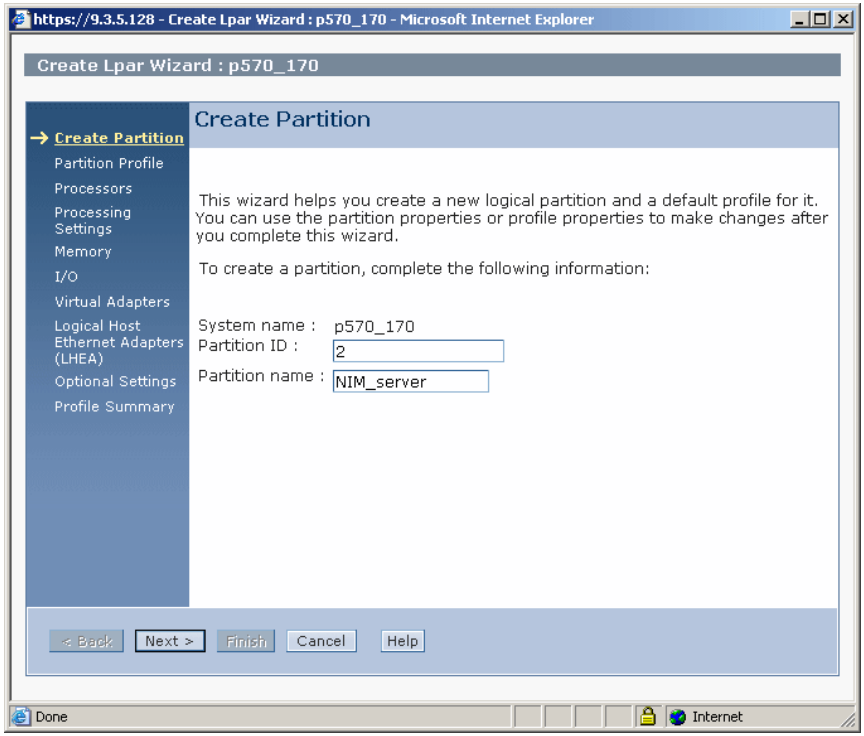


Figure 3-27 Creating a client partition

2. Repeat steps 1 to 6 of 3.3.1, “Defining the Virtual I/O Server partition” on page 153 with the following exceptions:
   a. Use 256/512/768 MB for minimum/desired/maximum memory settings on step 6, or appropriate values for your configuration.
   b. Use 0.1/0.5/1.0 processing units for minimum/desired/maximum settings on step 8, or appropriate values for your configuration.
   c. Use 1/1/4 for minimum/desired/maximum virtual processor settings on step 9. We set the maximum to four since we use a system with four processors and the system cannot utilize more than the number of physical processors.

3. Skip step 7 by clicking **Next** instead of choosing to create a physical I/O component selection since we are not using physical adapters in our client partitions.
4. Create virtual Ethernet and SCSI client adapters. The start menu for creating virtual adapters is shown in Figure 3-28. The default serial adapters are required for console login from the HMC and should not be modified or removed.

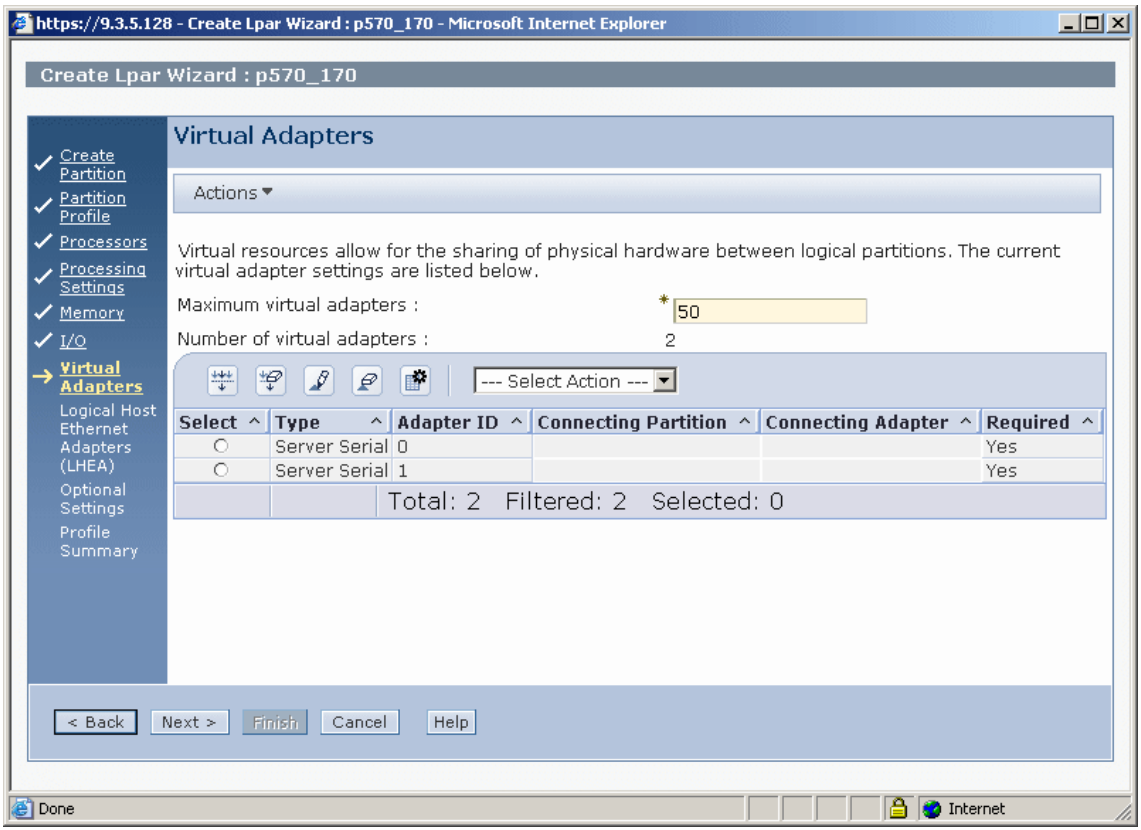


Figure 3-28 The start menu for creating virtual adapters window

5. Right-click **Actions** and select **Create** → **Ethernet Adapter** to open the Create Virtual Ethernet Adapter window. Create one virtual Ethernet adapter, as shown in Figure 3-29. Click **OK** when finished.

Important: Do not check the **Access external network** box for client adapters.

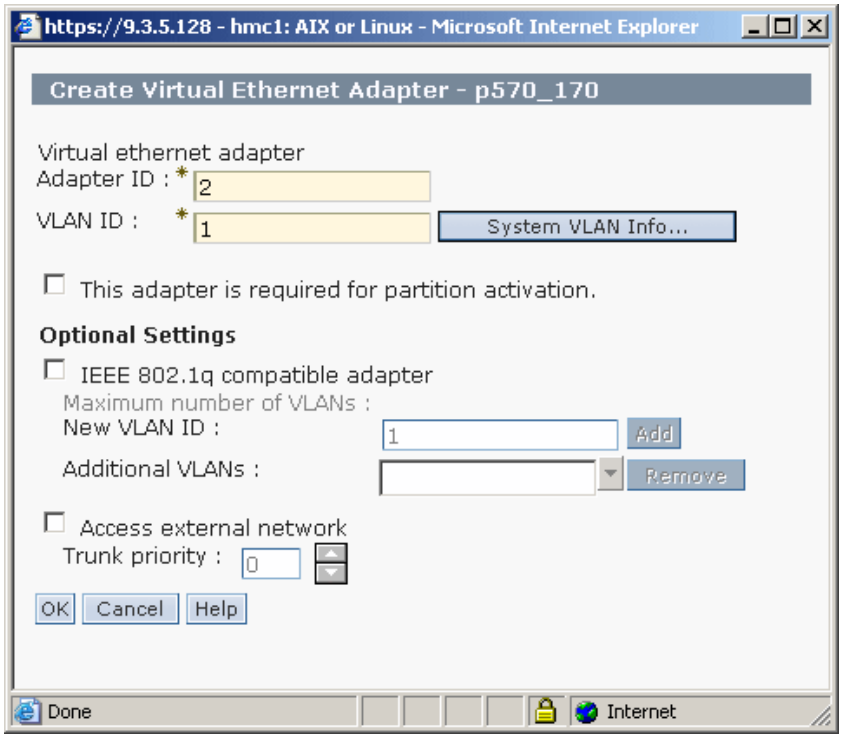


Figure 3-29 Creating a client Ethernet adapter

Note: We increased the maximum number of adapters to 50. Use any number that fits your configuration as long as it is less than 1024.

6. Right-click **Actions** and select **Create** → **SCSI Adapter** to open the Create Virtual SCSI Adapter window and create the client SCSI adapter. We want to create one SCSI adapter for disk and one SCSI adapter for the virtual optical device as shown in Figure 3-30 and Figure 3-31. Use Figure 3-25 on page 179 to select the correct client and server slot number. Make sure you select the correct Server partition in the menu. You can click **System VIOS Info** for more information about the slot numbers and their client-server relation.

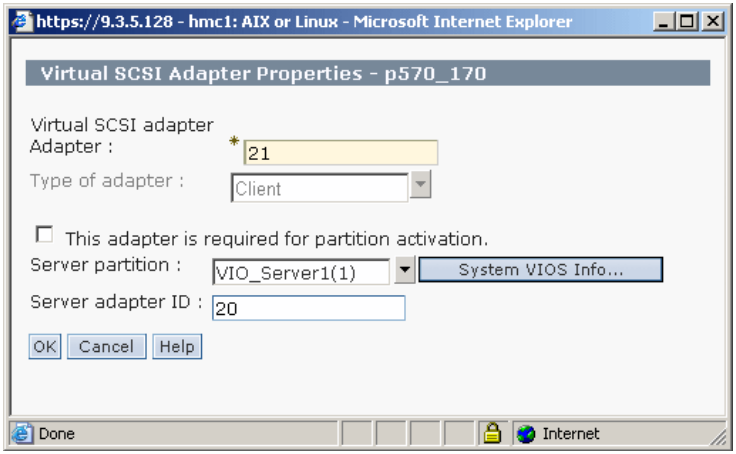


Figure 3-30 Creating the client SCSI disk adapter

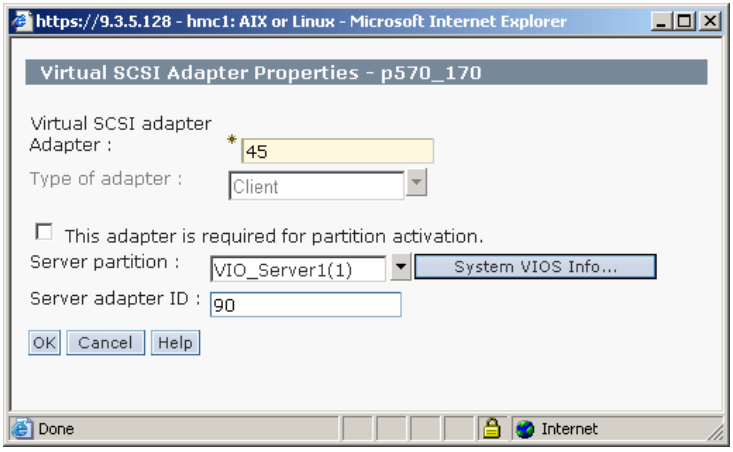


Figure 3-31 Creating the client SCSI DVD adapter

7. The list of created virtual adapters is shown in Figure 3-32. Click **Next** to continue.

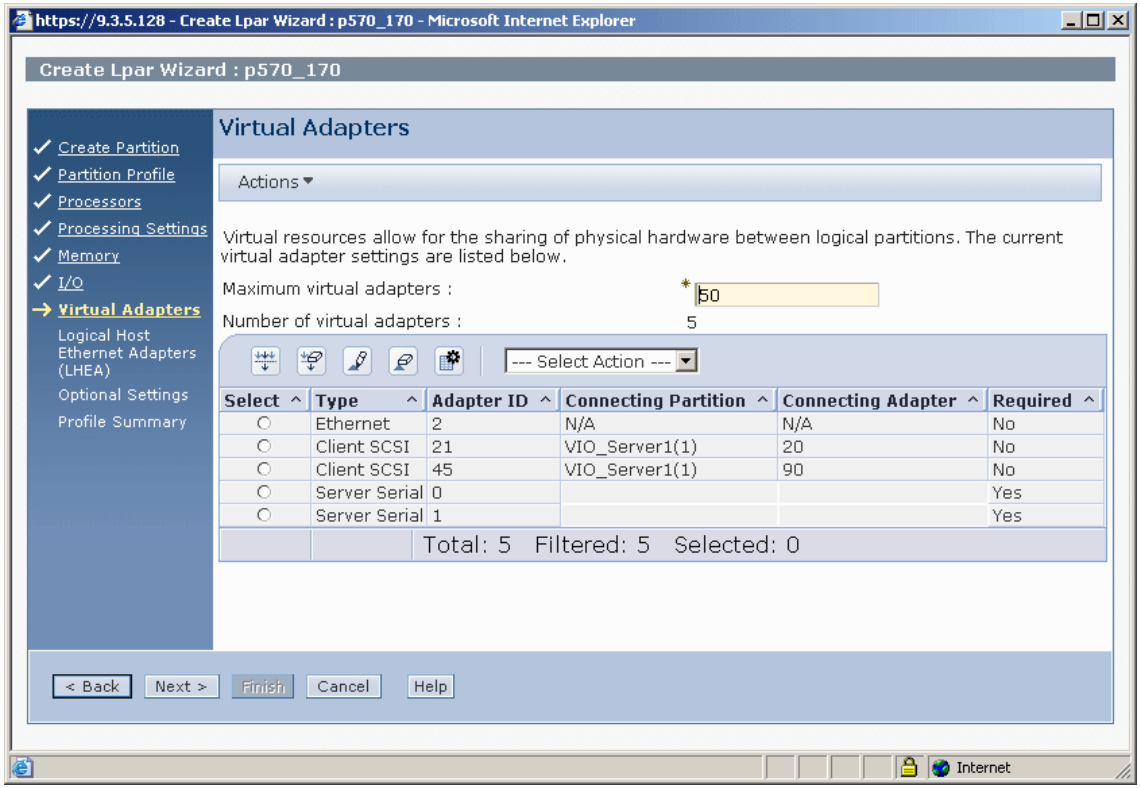


Figure 3-32 List of created virtual adapters

8. The Host Ethernet Adapter, HEA, is a new offering on the POWER6 system. It replaces the integrated Ethernet ports on previous systems and can provide logical Ethernet ports directly to the client partitions without using the Virtual I/O Server. For information about HEA, see *Integrated Virtual Ethernet Adapter Overview and Introduction*, REDP-4340, on:

   http://www.redbooks.ibm.com

   We will not use any of these ports for our basic setup. The setup window is shown in Figure 3-33. Click **Next** to continue.

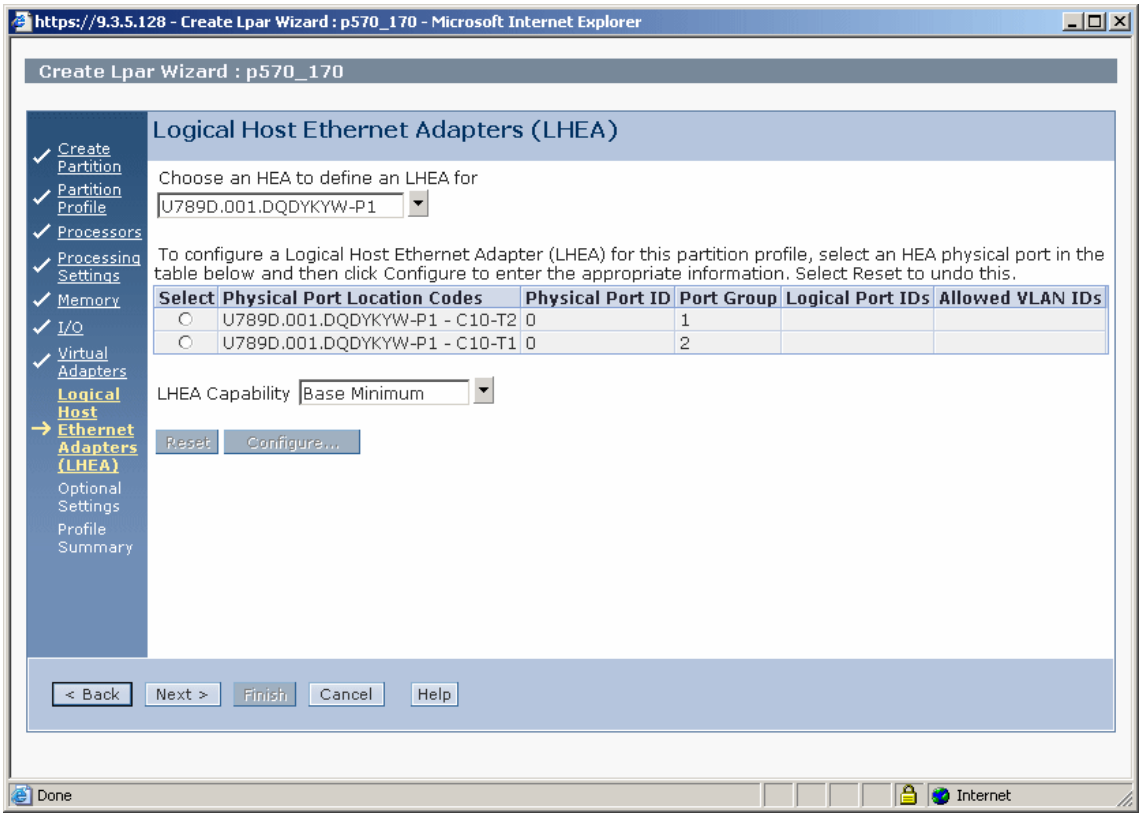


Figure 3-33 The Logical Host Ethernet Adapters menu

9. Choose **Optional Settings** for the partition (Figure 3-34). The partition will boot in normal mode by default.

   “Enable connection monitoring” will alert any drop in connection to the HMC. “Automatic start with managed system” means that the partition will start automatically when the system is powered on with the Partition auto start option (selected at power-on). “Enable redundant error path reporting” allows for call-home error messages to be sent through the private network in case of open network failure.

   Click **Next** to continue.

   Important: Enable redundant error path reporting must *not* be set for partitions that will be moved using Partition Mobility.

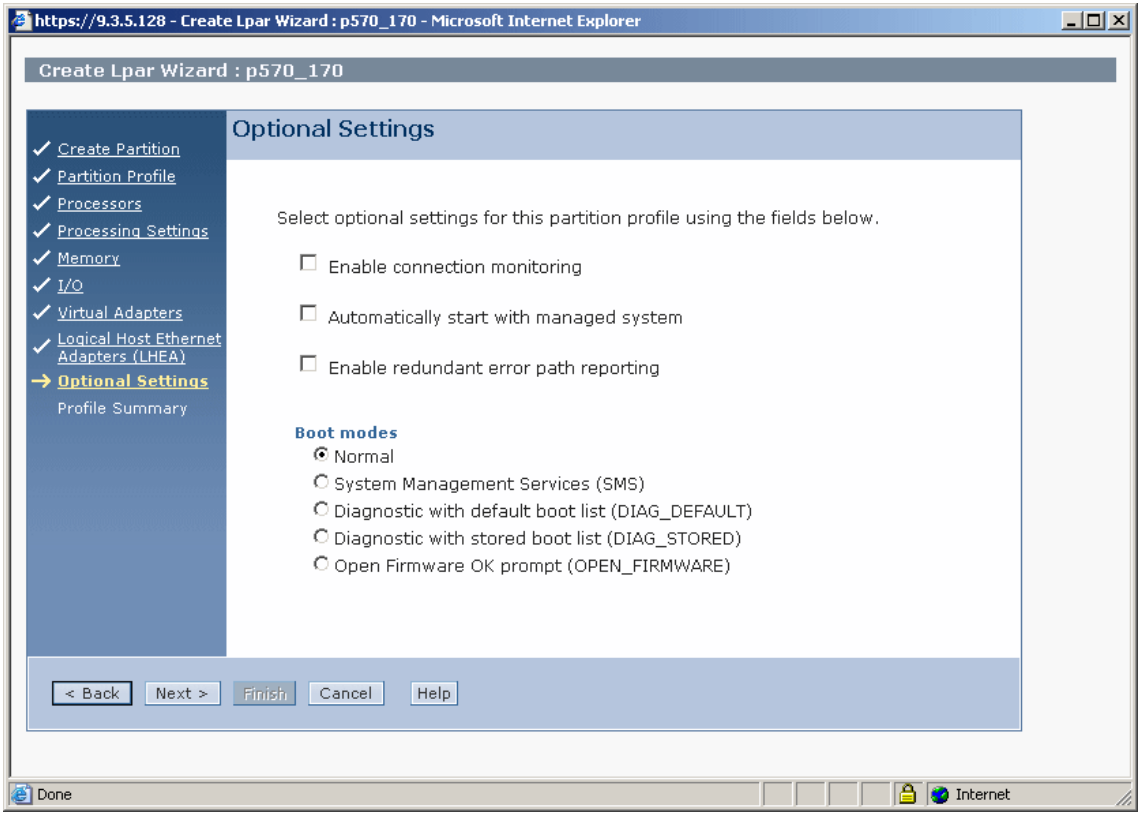


Figure 3-34 The Optional Settings menu

10. The Profile Summary menu shows details about the partition. You can check details about the I/O devices by clicking **Details** or click **Back** to go back and modify any of the previous settings. Click **Finish** to complete the profile.

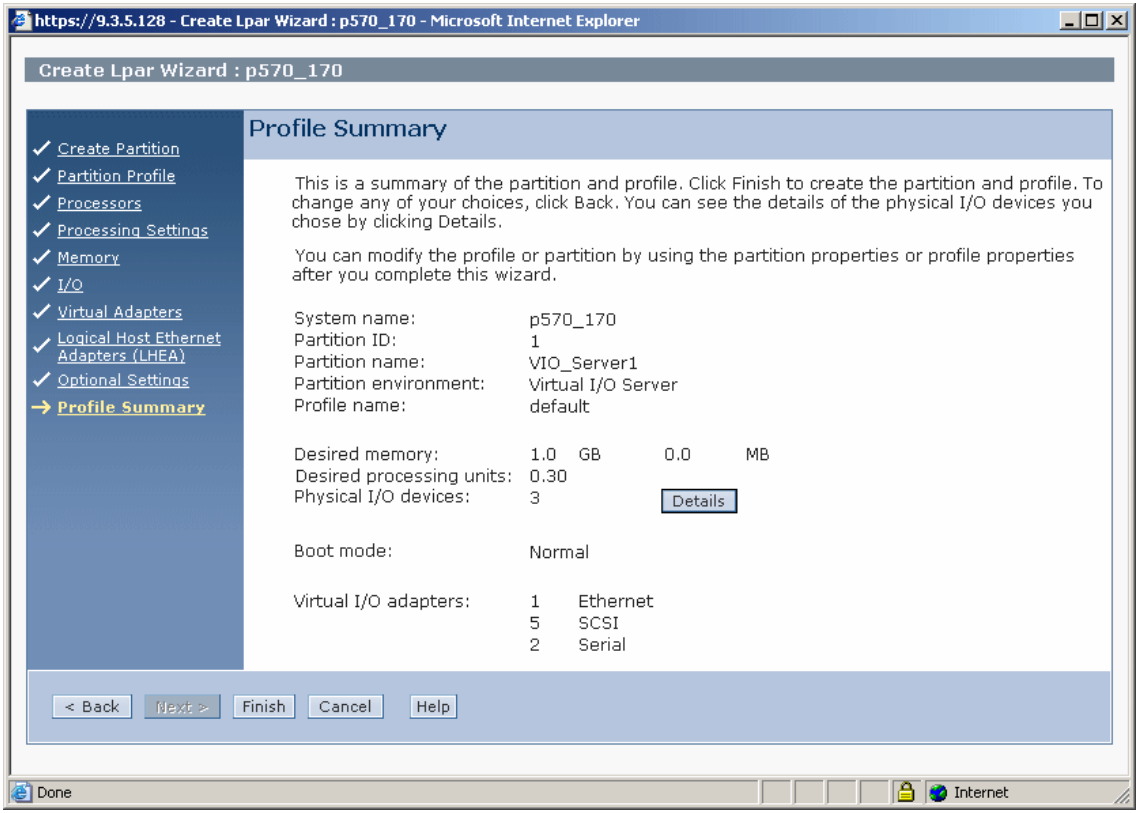


Figure 3-35 The Profile Summary menu

11. The complete list of partitions for the basic setup is shown in Figure 3-36. The NIM_server is selected for installation.

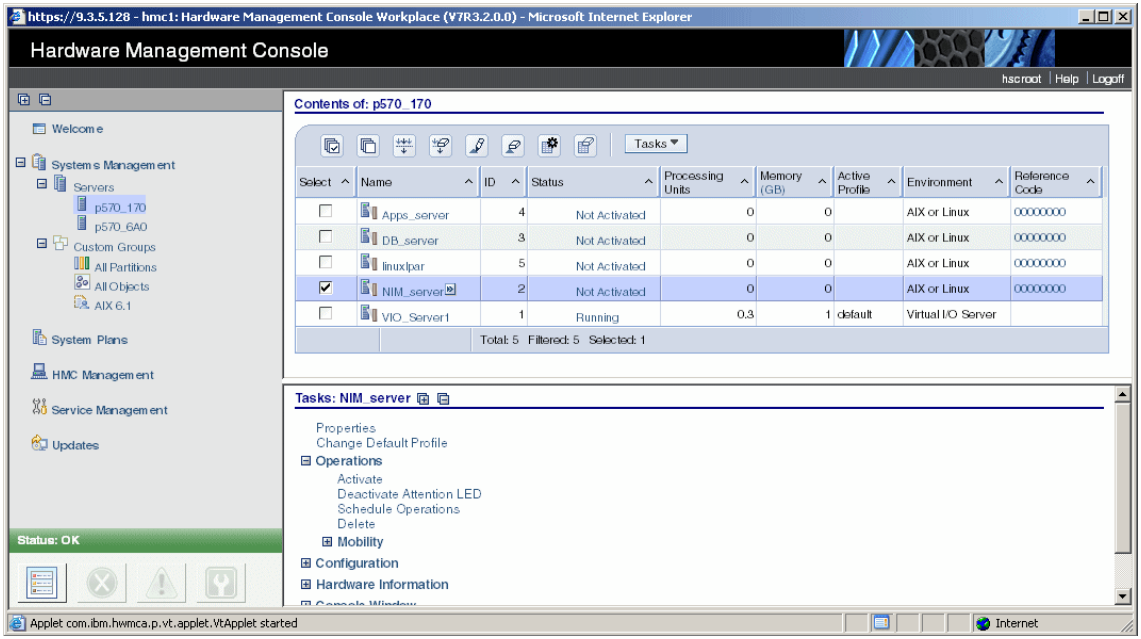


Figure 3-36 The list of partitions for the basic setup

12. It is best practice to back up the profile definition in case you want to restore it later. Activating the backup is shown in Figure 3-37. A menu is opened where you specify the name of the backup. Click **OK** to complete.

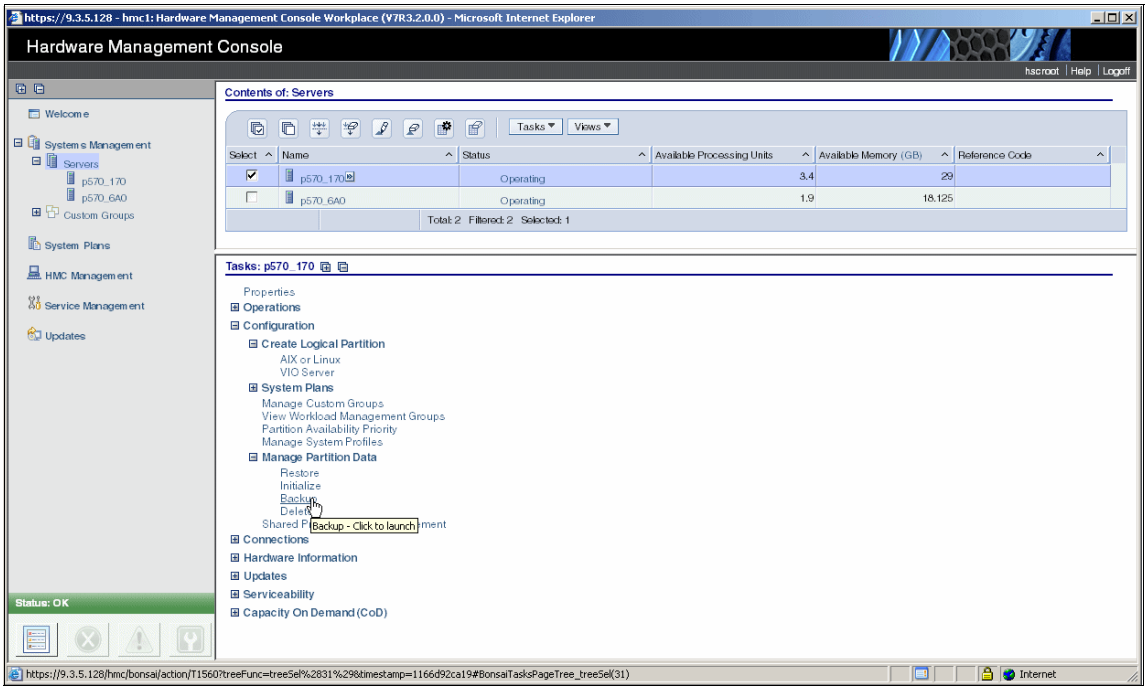


Figure 3-37 Backing up the profile definitions

3.5.6 AIX client partition installation

This section describes the method to install AIX onto a previously defined client partition. You can choose your preferred method but for our basic scenario we opted to install the NIM_server from CD and then install DB_server partition and Apps_server partition using the Network Installation Manager (NIM) from the NIM_server. We are also going to use the virtual Ethernet adapters for network booting and the virtual SCSI disks that were previously allocated to client partitions for rootvg.

Tip: A virtual optical device can be used for a CD or DVD installation as long as it is not already assigned to another client partition.

Assuming that a NIM master is configured, the following are the basic steps required to perform an AIX installation using NIM:

1. Create the NIM machine client dbserver and definitions on your NIM master. Example 3-28 shows how to check for allocated resources.

Example 3-28 Check if resources had been allocated

```
# lsnim -l dbserver
dbserver:
   class          = machines
   type           = standalone
   connect        = nimsh
   platform       = chrp
   netboot_kernel = 64
   if1            = network1 dbserver 0
   cable_type1    = N/A
   Cstate         = BOS installation has been enabled
   prev_state     = ready for a NIM operation
   Mstate         = not running
   boot           = boot
   lpp_source     = aix61_lppsource
   nimscript      = nim_script
   spot           = aix61_spot
   control        = master

# tail /etc/bootptab
#      T170 -- (xstation only) -- server port number
#      T175 -- (xstation only) -- primary / secondary boot host
indicator
#      T176 -- (xstation only) -- enable tablet
#      T177 -- (xstation only) -- xstation 130 hard file usage
#      T178 -- (xstation only) -- enable XDMCP
#      T179 -- (xstation only) -- XDMCP host
#      T180 -- (xstation only) -- enable virtual screen
dbserver:bf=/tftpboot/dbserver:ip=9.3.5.113:ht=ethernet:sa=9.3.5.197
:sm=255.255.254.0:
```

2. Initiate the install process by activating the DB_server client partition in SMS mode (see Figure 3-38). Figure 3-22 on page 176 and Figure 3-23 on page 176 show how to activate a partition in SMS mode.

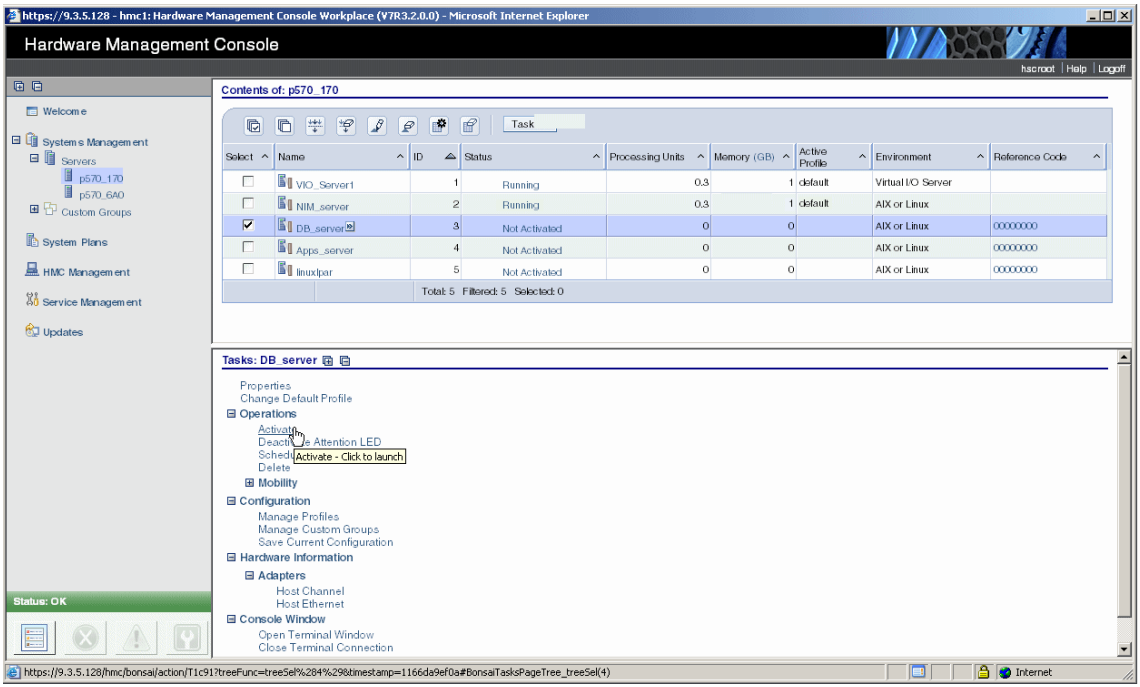


Figure 3-38 Activating the DB_server partition

Continue with the following steps in SMS:

3. Set up the network boot information by choosing option 2, Setup Remote IPL (see Figure 3-39).

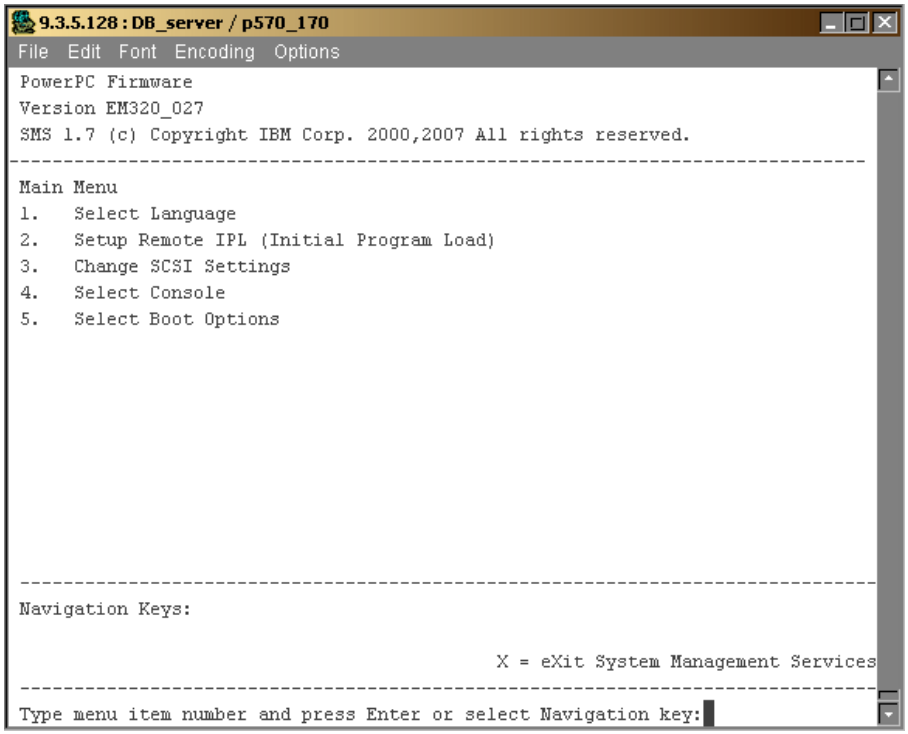


Figure 3-39 The SMS menu

4. Choose option 1, as shown in Figure 3-40.

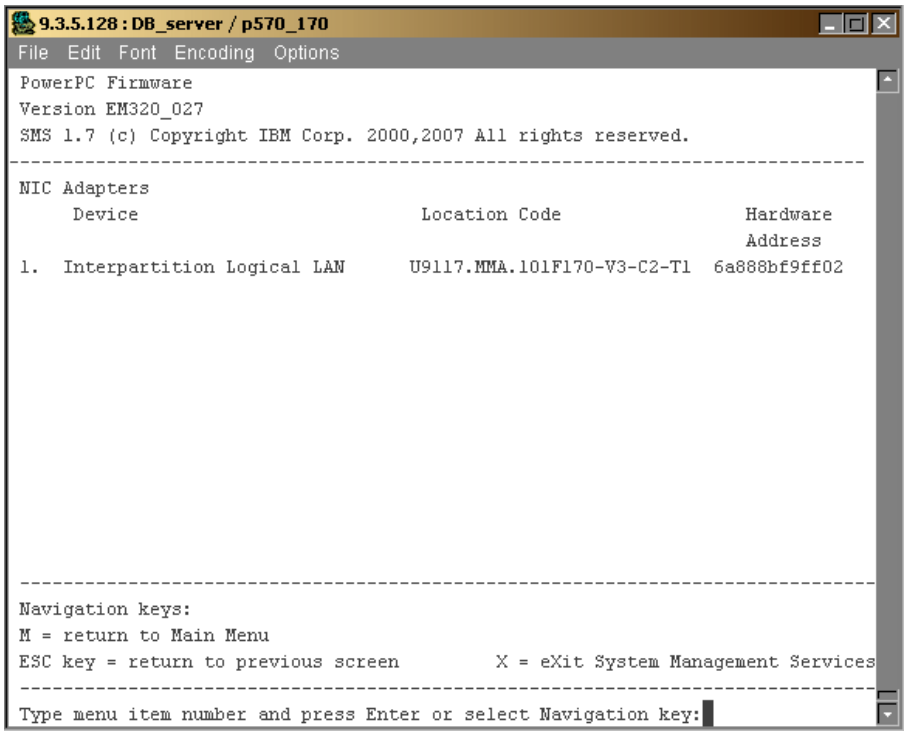


Figure 3-40 Selecting adapter

Note: Interpartition Logical LAN number 1 is the virtual Ethernet adapter that was defined on the NIM master for the DB_server client:

```
dbserver:
   class          = machines
   type           = standalone
   connect        = nimsh
   platform       = chrp
   netboot_kernel = 64
   if1            = network1 dbserver 0
```

5. Choose option 1 for IP Parameters, then go through each of the options and supply the IP address, as shown in Figure 3-41.

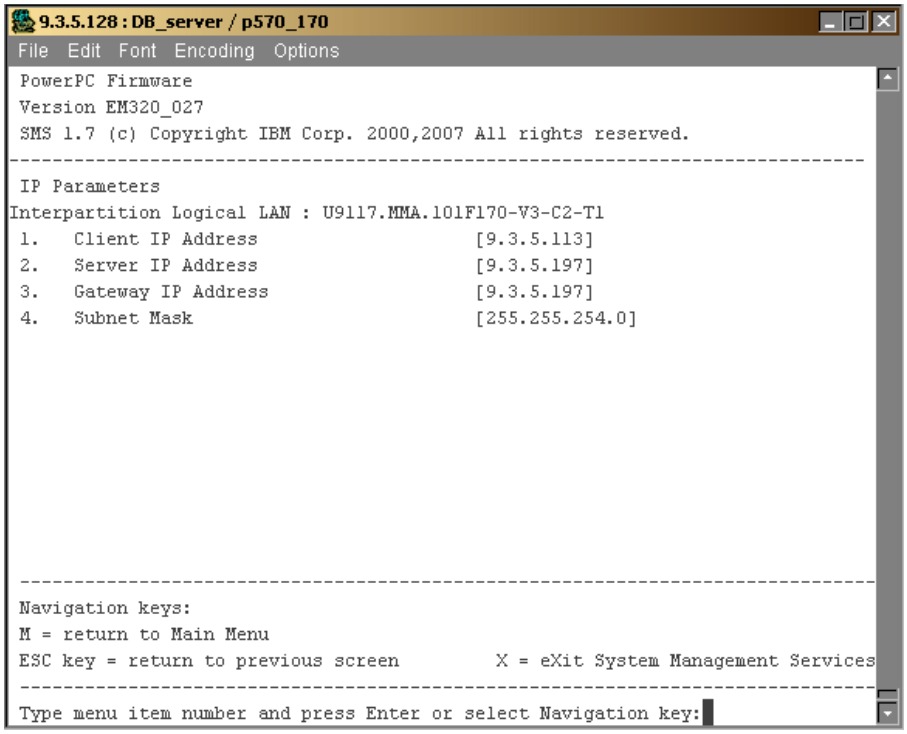


Figure 3-41 IP settings

6. Press Esc to go one level up for the **Ping test** as shown in Figure 3-42.

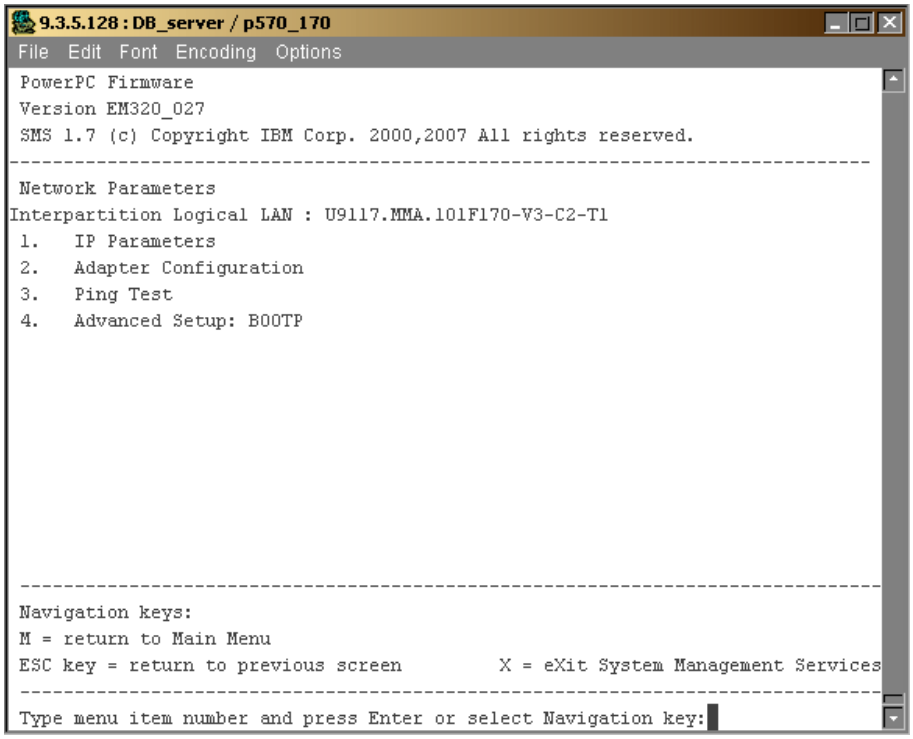


Figure 3-42 Ping test

7. Select 3 to execute a ping test and, provided it is successful, you are ready to do the NIM installation.
8. Press **M** to get back to the main menu.
9. Select 5 to verify that boot is set to the correct network adapter.
10. Select 2 for boot order.
11. Select the first boot device, option 1.

12. It is useful to list all devices with option 8. In Figure 3-43 you can see that no device is listed in Current Position.

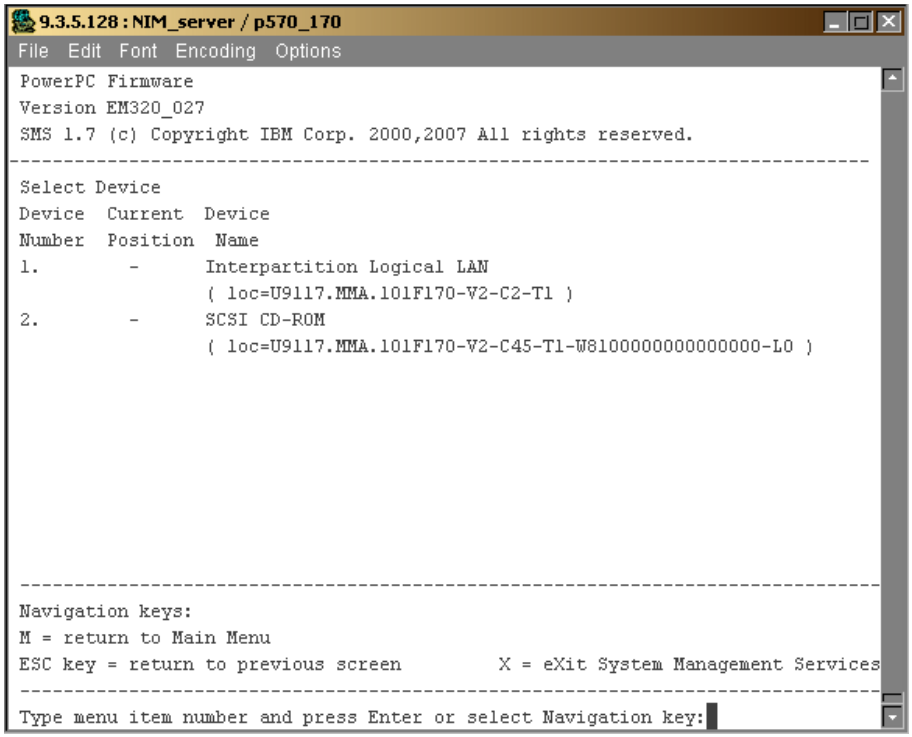


Figure 3-43 Setting the install device

13. Select option 1. We want the Interpartition Logical LAN to be our boot device.
14. Select option 2 to Set Boot Sequence.
15. .Press X to exit from SMS and confirm with 1 to start the network boot.

Note: It is best practice to use separate volume groups for applications and user data in order to keep the rootvg volume group reserved for the AIX operating system. This makes rootvg more compact and easier to manipulate if required.

3.5.7 Dynamic partitioning for virtual SCSI devices

Virtual SCSI resources can be assigned and removed dynamically using the HMC.

Virtual SCSI server and client adapters can be assigned to a partition using dynamic logical partitioning.

A virtual adapter can only be removed from a partition after it has been released from the operating system (removed or defined state).

Moving a virtual disk between partitions involves reassigning the Virtual I/O Server vhost adapter to a new client partition followed by creating a new client partition VSCSI client adapter. On the client, the **cfgmgr** command must be run to make the VSCSI disk available.

Important: Dynamic LPAR operations rely on a network connection between HMC and the involved partitions.

3.5.8 Dedicated donating processors

When using dedicated processors there are options to donate unused processor cycles to Virtual Shared Processor Pools on POWER6 systems.

These options are set when editing the profile after it is created.

To set this option, do the following:

1. Select the partition with dedicated processors.
2. Select **Configuration** → **Manage Profiles** to open the Managed Profiles window.

3. Select a profile and click **Actions** → **Edit** as shown in Figure 3-44.

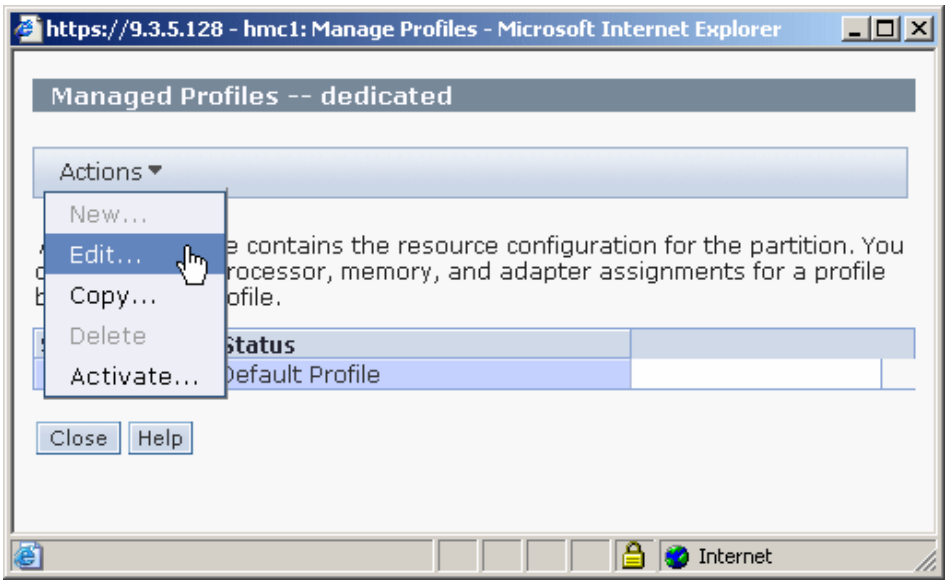


Figure 3-44 The edit Managed Profile window

4. The Logical Partition Profile Properties is opened. Open the Processors tab as shown in Figure 3-45 where the Processor Sharing options can be set.
   - The option **Allow when partition is inactive** is set by default and indicates whether the dedicated processors are made available to shared processor partitions when the logical partition that is associated with this partition profile is shut down.
   - The option **Allow when partition is active** indicates whether the dedicated processors are made available to shared processor partitions when the logical partition that is associated with this partition profile is active.

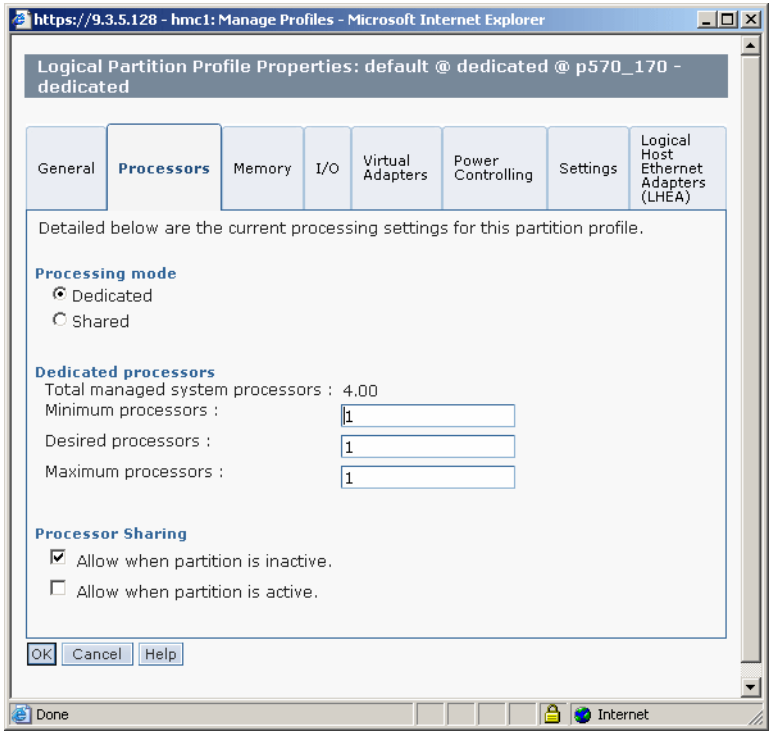


Figure 3-45 Setting the Processor Sharing options

3.6 Interaction with UNIX client partitions

This section describes how the Virtual I/O Server provides resources to the AIX or Linux partitions. These resources can be storage and optical devices through virtual SCSI or network connectivity through virtual Ethernet.

Note: For Linux on POWER systems, Hardware service diagnostic aids and productivity tools, as well as installation aids for IBM servers running the Linux operating systems on POWER6, POWER5, and POWER4-based processors, are available from:

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

3.6.1 Virtual SCSI resources

For virtual SCSI, the interaction between a Virtual I/O Server and an AIX or Linux client partition is enabled when the configurations of both virtual SCSI server adapter and virtual SCSI client adapter have mapped slot numbers in their partition profiles, and both operating systems recognize their virtual adapter (with the **cfgmgr** command for dynamically added virtual SCSI adapters).

Once the interaction between virtual SCSI server and client adapters is enabled, mapping storage resources from the Virtual I/O Server to the client partition is needed. The client partition configures and uses the storage resources when it starts up or when it is reconfigured at runtime.

The process runs as follows:

- The HMC maps interaction between virtual SCSI adapters.
- The mapping of storage resources is performed in the Virtual I/O Server.
- The client partition uses the storage after it has been booted or a **cfgmgr** command is run.

For more information about the technical details of virtual SCSI implementation, refer to 2.9, “Software licensing in a virtualized environment” on page 109.

Figure 3-46 shows the flow needed to enable virtual SCSI resources to AIX or Linux clients. Note that Virtual I/O Server and client partitions do not need to restart when new virtual SCSI server and client adapters are created using the dynamic LPAR menus in the HMC and dynamic LPAR operations are enabled in the operating systems.

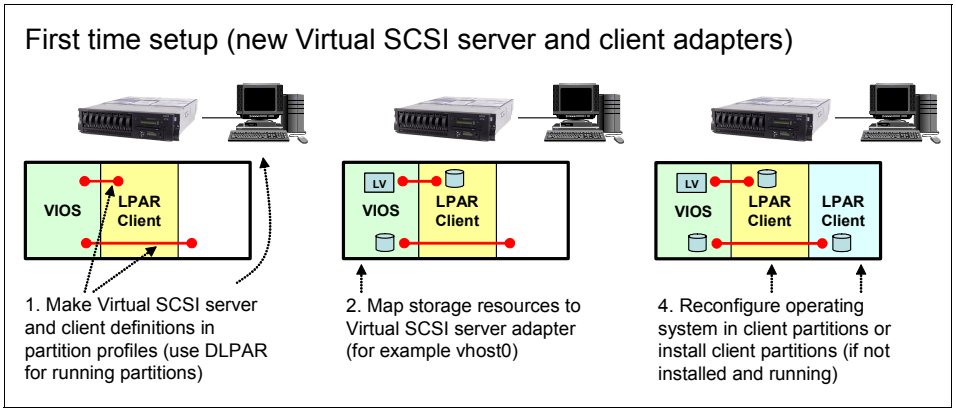


Figure 3-46 Basic configuration flow of virtual SCSI resources

To enable the AIX or Linux client partitions to interact with virtual SCSI resources, the following steps are necessary:

1. Plan which virtual slot will be used in the Virtual I/O Server for the virtual SCSI server adapter and which slot in the AIX or Linux client partition for the virtual SCSI client adapter (each partition has its own pool of virtual slots). In the Virtual I/O Server, consider ranges of slot numbers for virtual adapters that serve a specific partition (for example, slots 20 through 29 for virtual SCSI server adapters for an AIX 5L client partition).
2. Define the virtual SCSI server adapter on the Virtual I/O Server.
3. Define the SCSI client adapter on the AIX or Linux client partition.
4. Map the desired SCSI resources using the **mkvdev** command, as described in 3.5.2, “Defining virtual disks” on page 182.

Once both the Virtual I/O Server and AIX or Linux client partitions are interacting with each other, virtual devices can be mapped online on the Virtual I/O Server and the AIX 5L or Linux client partitions can be reconfigured online to use them, as shown in the Figure 3-47.

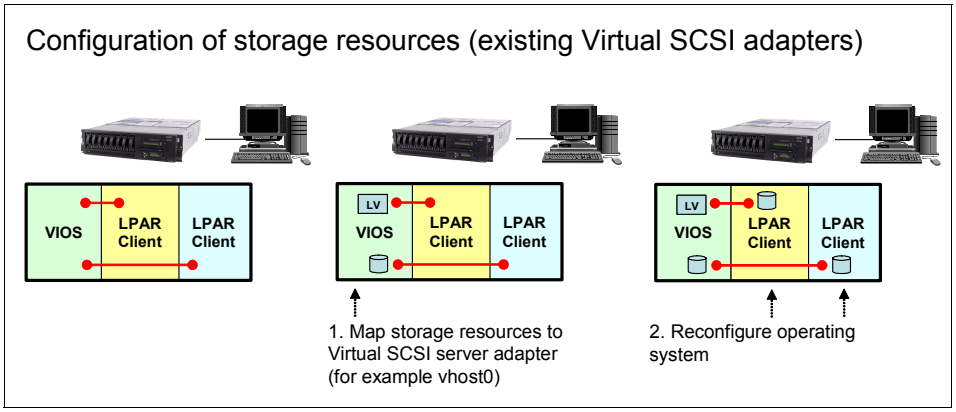


Figure 3-47 Basic configuration flow of virtual SCSI resources

Virtual SCSI services for AIX partitions

AIX client partitions can use virtual SCSI devices currently mapped from the Virtual I/O Server once the operating system starts up or the system is reconfigured with the **cfgmgr** or **mkdev** commands.

Mapped disk-type resources (physical volumes or logical volumes in the Virtual I/O Server scope) appear on the AIX client partition as an hdisk type of device (for example, hdisk0, hdisk1). The virtual SCSI client adapter can use these devices like any other physically connected hdisk device for boot, swap, mirror, or any other supported AIX feature. Optical-type resources (DVD-ROM and DVD-RAM) appear as a CD type of device (for example, cd0).

Virtual SCSI services for Linux partitions

The configuration for Linux partitions to enable them to use virtual SCSI adapters and devices is similar to the configuration for AIX partitions. Linux partitions handle virtual SCSI devices as SCSI disc drives. Clients should be aware of the following considerations in the Linux client partitions:

- Virtual SCSI adapters in the Linux client partition are listed in the /sys/class/scsi_host directory; such a directory contains the control files for each of the virtual adapters. The virtual SCSI client adapters can be requested to re-scan for virtual SCSI resources recently mapped from the Virtual I/O Server using the same interfaces as any other SCSI adapter.
- Virtual SCSI disk drives in the Linux client partition can be seen in the device tree as an sd*x* type of device (for example, sda). Note that Linux shows in the device tree both virtual disk drives (virtual SCSI storage resources mapped from Virtual I/O Server) and disk partitions created inside those virtual disk drives.

3.6.2 Virtual Ethernet resources

The second type of resource that interacts between AIX or Linux client partitions and the Virtual I/O Server is the Shared Ethernet Adapter in the Virtual I/O Server. This feature allows the AIX partitions to connect to external networks without a physical adapter.

The implementation of virtual Ethernet adapters is based on the definition of network interfaces that connect through the POWER Hypervisor to an IEEE VLAN-aware virtual Ethernet switch in the system. All partitions talking on the virtual Ethernet network are peers. Up to 4,096 separate IEEE VLANs can be defined in the system. Each partition can have up to 65,533 virtual Ethernet adapters connected to the virtual Ethernet switch and each adapter can be connected to 21 different IEEE VLANs (20 VID and 1 PVID).

The enablement and setup of a virtual Ethernet adapter in an AIX or Linux client partition does not require any special hardware or software (could require a special release of the software). After a specific virtual Ethernet is enabled for a partition, a network device is created within the partition. The user can then set up the TCP/IP configuration appropriately to communicate with other partitions.

In order to allow the AIX or Linux client partition to communicate with external Ethernet networks, the Virtual I/O Server acts like a bridge and forwards the IP packets from the virtual Ethernet client adapter. This is done by creating a Shared Ethernet Adapter (SEA) in the Virtual I/O Server that connects the virtual Ethernet server adapter and a physical Ethernet adapter in the server. For more information about the Shared Ethernet Adapter, refer to 2.7, “Virtual and Shared Ethernet introduction” on page 83.

Note: Remember that virtual SCSI or virtual Ethernet adapters created with dynamic LPAR operations are not reflected in the partition's profiles and exist at partition runtime only. You can either update your profile or save your current configuration to a new profile to make your changes permanent across restarts.

3.6.3 Simplification by using system plans and System Planning Tool

This section describes how the HMC system plans and the PC-based System Planning Tool, SPT, can document your configuration and simplify partition deployment. SPT can be downloaded for free from the SPT Web site.

In-depth information can be found at:

http://www-304.ibm.com/jct01004c/systems/support/tools/systemplanningtool

SPT is introduced in order to simplify planning and deployment of LPAR configurations.

- SPT allows for planning of new systems before delivery and deployment of system plans on the HMC or IVM, which can speed up the process.
- Automatic checking in SPT when you create a system plan and validation during the deployment process on the HMC can help to install a correct configurations faster than by manual input on the HMC.

The Web site also has references to documentation and *IBM System i and p System Planning and Deployment: Simplifying Logical Partitioning,* SG24-7487.

Note: The version of SPT used at the time of writing is 2.07.320.

Creating a configuration using SPT and deploying on the HMC

A new system plan (sysplan) can be created using SPT in several ways:

- Create a new system (advanced)
- Import from another system plan
- Based on IBM-supplied sample systems
- Based on existing performance data based on IBM Systems Workload Estimator, WLE
- Based on new workloads you want to run

A new system plan can be created to be the basis for a new system or a new configuration of an existing system.

The Basic Virtual I/O Server scenario in 3.5, “Basic Virtual I/O Server scenario” on page 178 was created using SPT, imported on the HMC and deployed as shown in Figure 3-61 on page 236.

Creating a system plan using SPT is outside the scope of this book. The tool is intuitive; you can refer to the SPT Web site for more information.

These are the steps to deploy the Basic scenario in 3.5, “Basic Virtual I/O Server scenario” on page 178 using a POWER5 p520:

1. Create a valid system plan using the SPT. Figure 3-48 shows the Edit Virtual Slots window in SPT. This menu is very useful for ensuring correct virtual adapter slot numbering and correct pairing of server-client virtual SCSI adapters. You can edit slot numbers and a total (maximum) slot number to fit your configuration and planned numbering

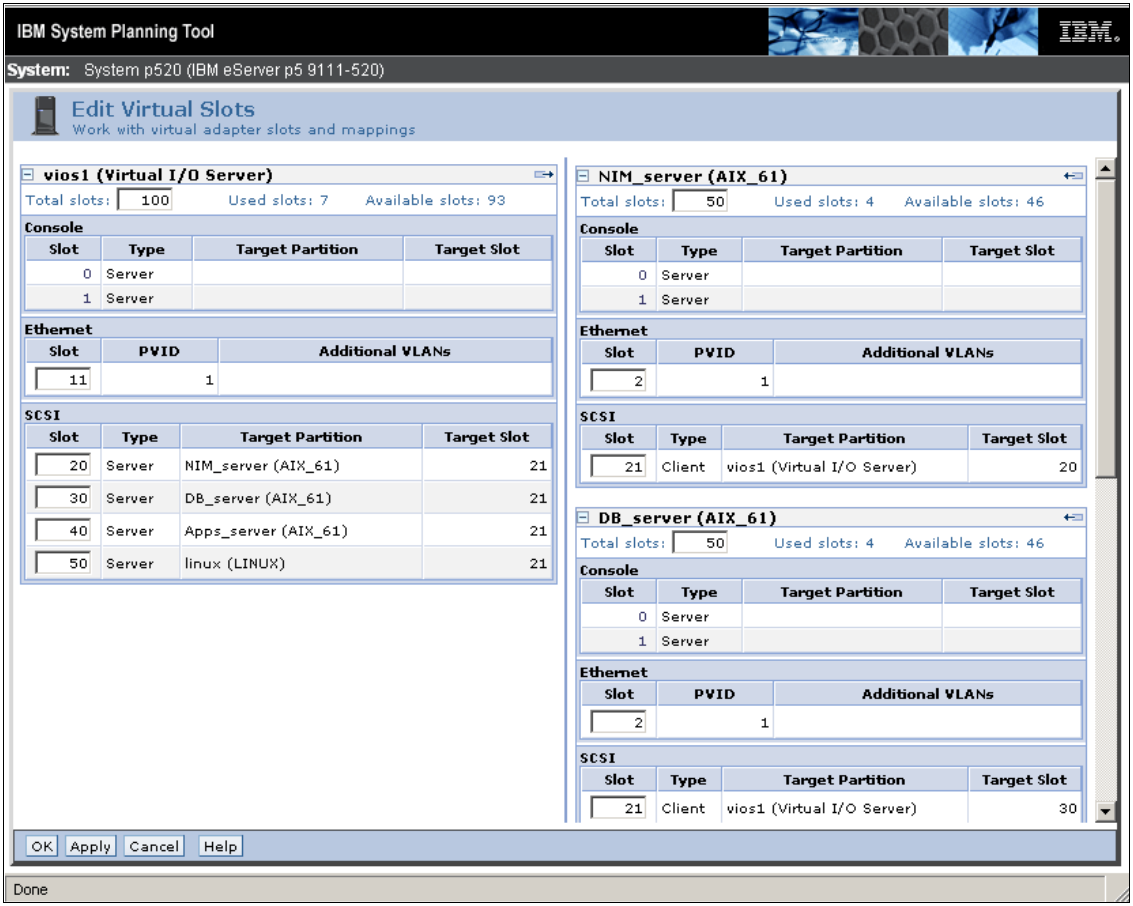


Figure 3-48 Edit Virtual Slots in SPT

2. Select System Plans as shown in Figure 3-49.

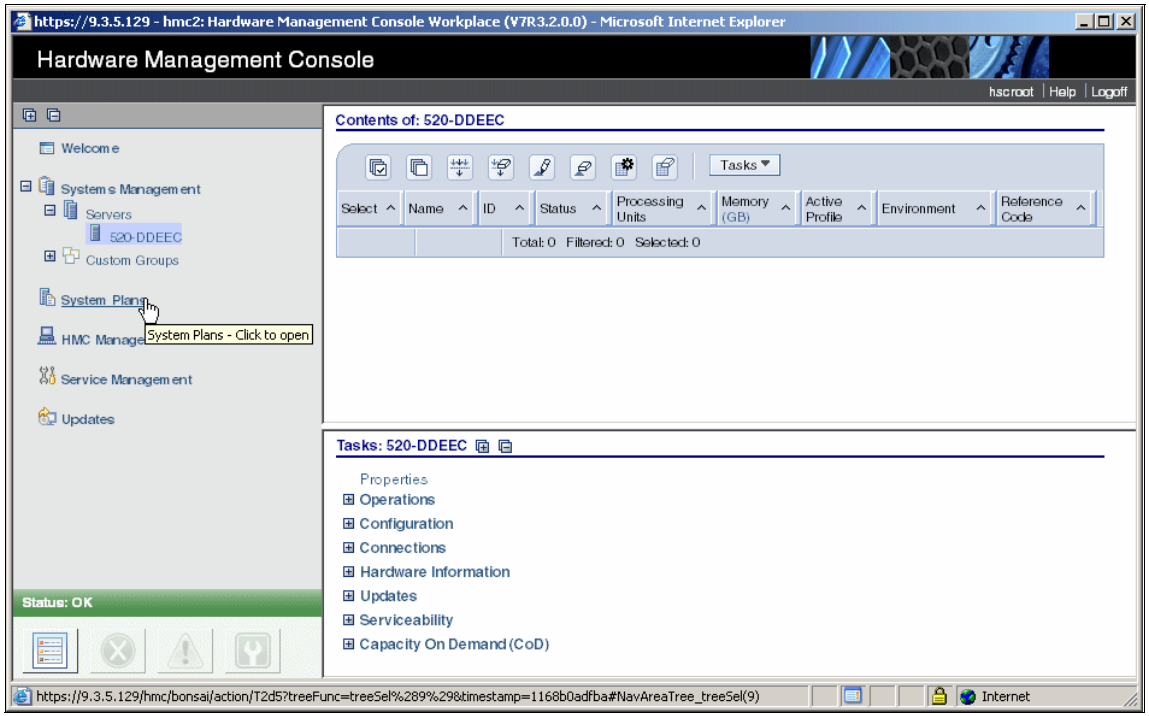


Figure 3-49 Selecting to work with System Plans

d. Import the system plan on the HMC by clicking **Import System Plan** to open the Import System Plan window and selecting the Import from this computer to the HMC option as shown in Figure 3-50 on page 225. Select the system plan file to be imported from your PC.

Note: All system plan file names on the HMC and SPT must have the extension .sysplan.

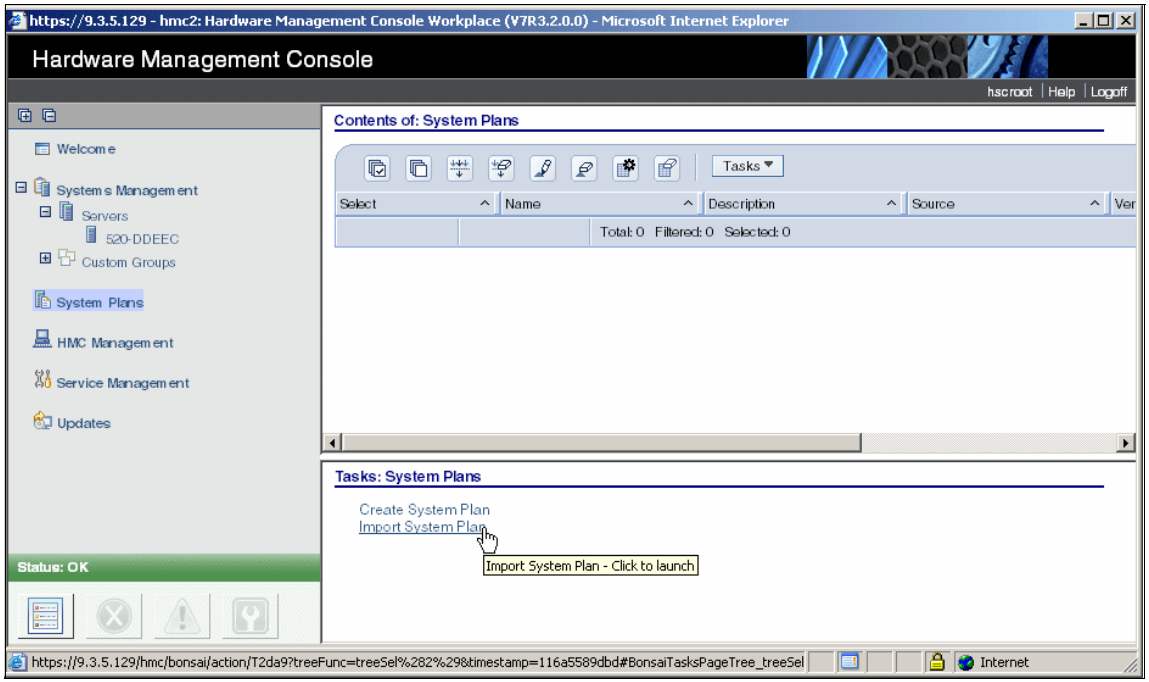


Figure 3-50 Importing a system plan

3. Select the system plan on the HMC as shown in Figure 3-51 and Click **Deploy System Plan** to open the Deploy System Plan Wizard, as shown in Figure 3-52. Click **Next** to continue.

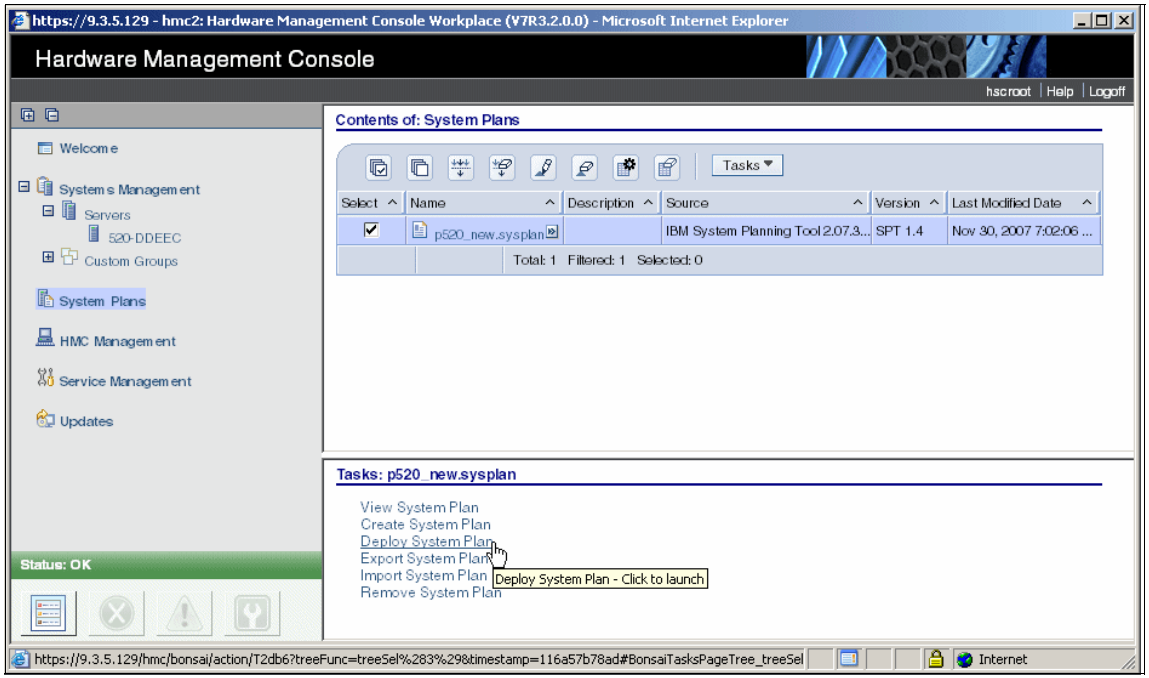


Figure 3-51 Deploying a system plan

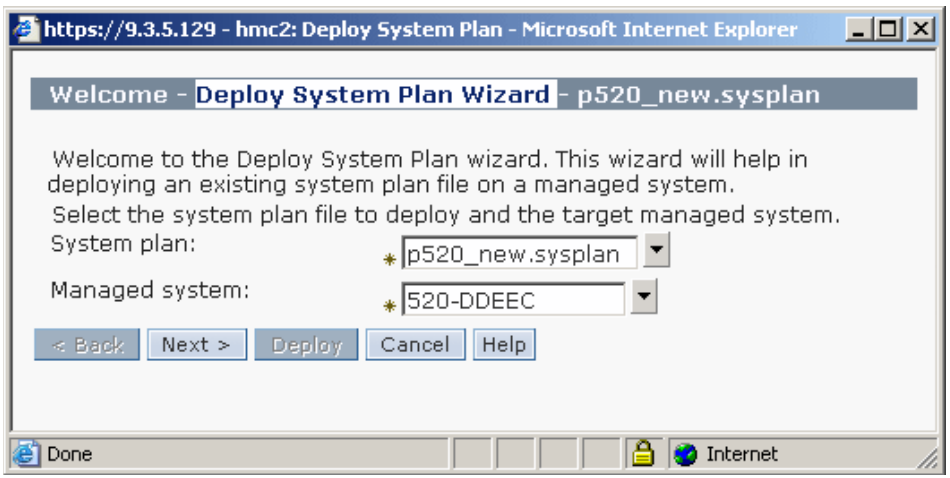


Figure 3-52 Opening the Deploy System Plan Wizard

4. The system plan is validated as shown in Figure 3-53. If it fails you can usually find to cause in the Validation Messages section. Click **Next** if successful.

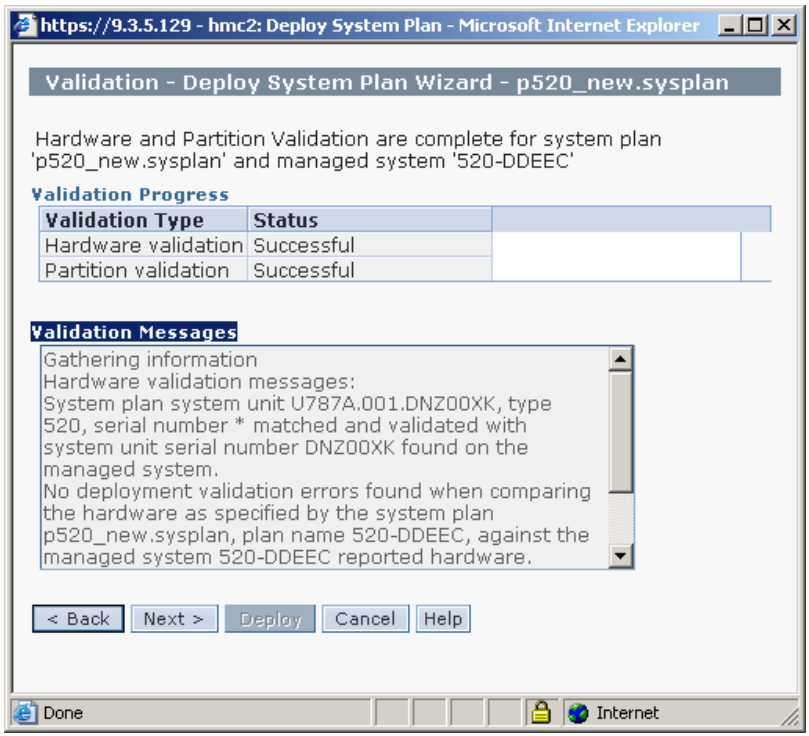


Figure 3-53 System plan validation

5. In this panel you can select which partitions to deploy. Figure 3-54 shows the Partition Deployment window. Click **Next** to continue.

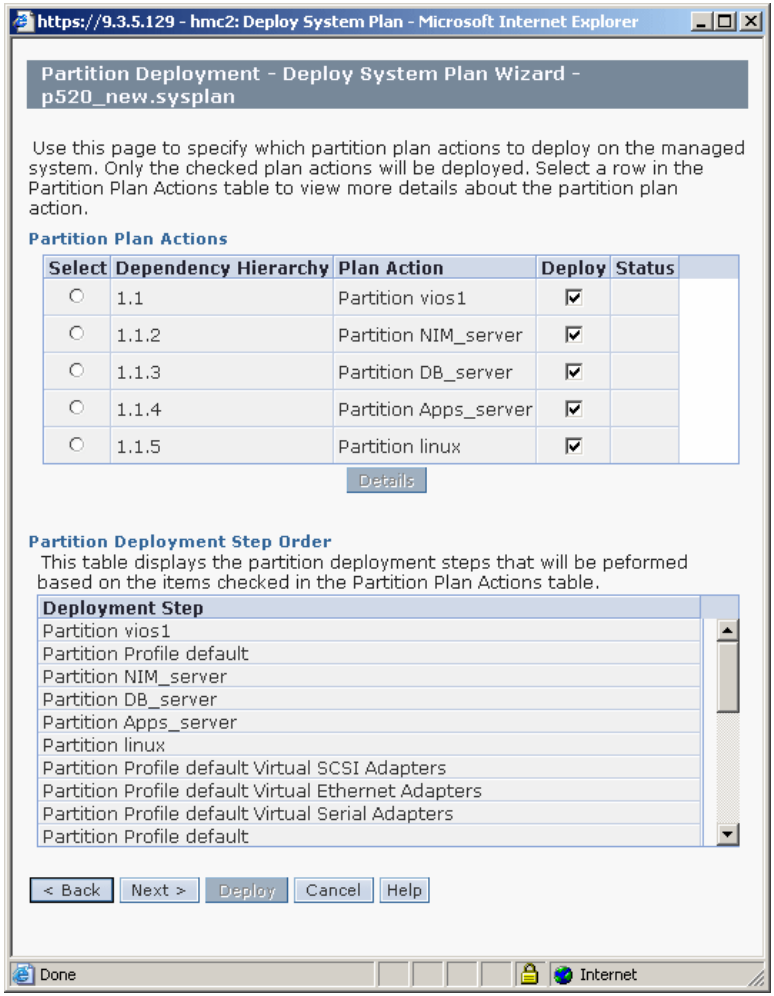


Figure 3-54 The Partition Deployment window

Note: If you deselect the Virtual I/O Server, all client partitions are also deselected.

6. In the Operating Environment Install Deployment window, shown in Figure 3-55, you have the option of installing the Virtual I/O Server from the HMC repository or skip the installation to just create the partition profiles.

   If you wish to install the Virtual I/O Server, the installation software must reside on the HMC. There is currently no GUI for installing the Virtual I/O Server DVD media on the HMC. To install using the command line you need to open a terminal window to the HMC. Example 3-29 shows an example of a command to copy the DVD on to the HMC repository. The DVD media must be in the DVD drive of the HMC. You can use any name instead of vios_res_15.

   The default is to install the Virtual I/O Server. If you want to skip the installation and only have the partition profiles generated, you have to uncheck the **Deploy** box. For the basic scenario it was decided to not install the Virtual I/O Server at deployment. If you decide not to install the Virtual I/O Server at this point, skip the next step, 8. Click **Next** to continue.

Example 3-29 HMC command to copy Virtual I/O Server media to the HMC

```
hscroot@hmc2:~> OS_install -o define_resource -a type=AIX -a version=1.5.1.0\
-a location=/export/resources/vios15 -a source=/dev/cdrom vios_res_15
```

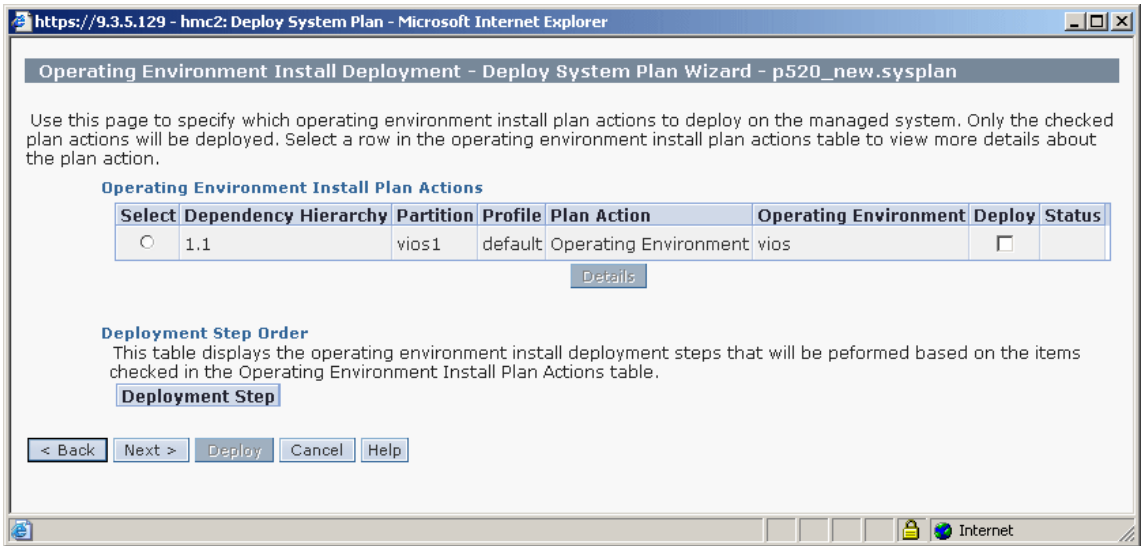


Figure 3-55 Operating Environment installation window

7. If you want the Virtual I/O Server to be installed you must leave the Deploy box ticked (default) and the Customizing Operating Environment window is opened as shown in Figure 3-56. In this menu you provide information about the Virtual I/O Server source on the HMC and the network information to be set for the partition. The first time you install from the HMC resource you must open the New Install Image Resource menu as shown in Figure 3-57 on page 231 to specify which resource to use. You also have to open the Modify Install Settings menu as shown in Figure 3-58 on page 231 to set the network parameters for network installation. Click **Next** to continue.

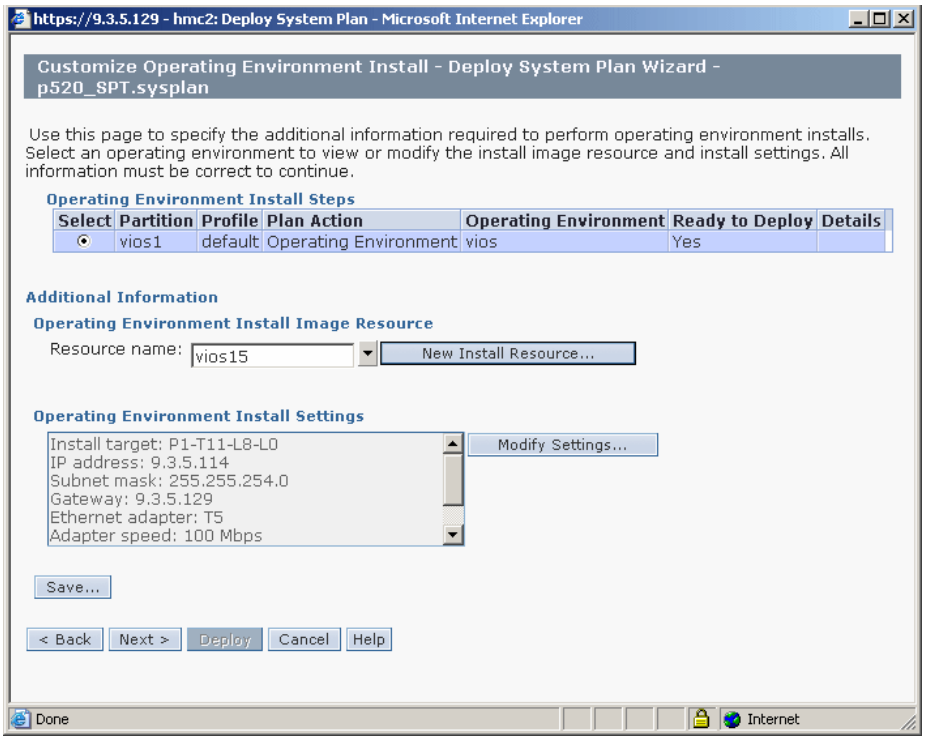


Figure 3-56 The Customize Operating Environment menu

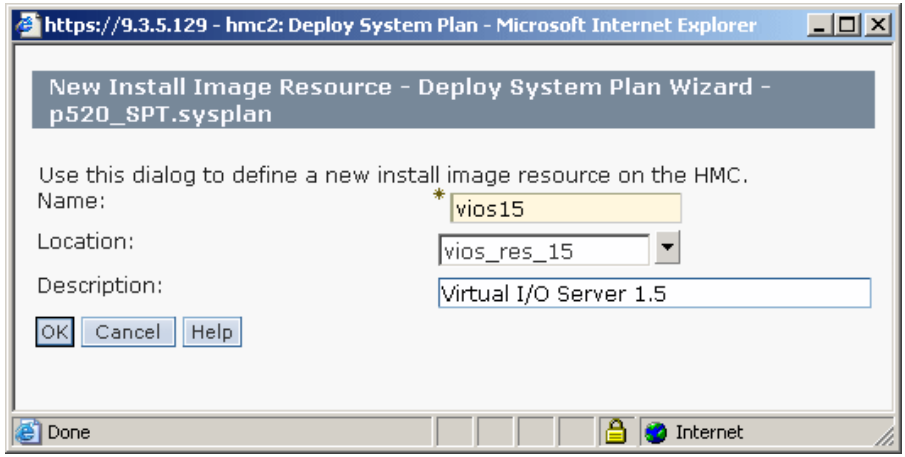


Figure 3-57 The New Install Image Resource menu

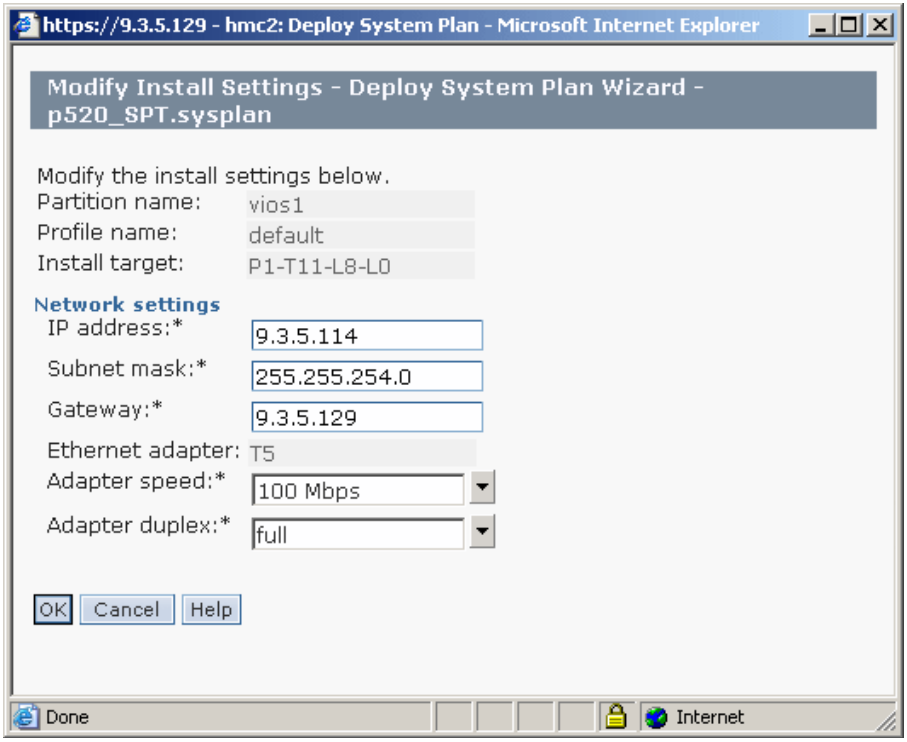


Figure 3-58 Virtual I/O Server installation setup

Important: Verify that the Ethernet adapter is correct and connected to the network. The adapter displayed in the menu is the adapter you selected for installation in SPT.

Installing the Virtual I/O Server image using installios

You can also use the **installios** command on the HMC to install the Virtual I/O Server image (stored on the HMC or the Virtual I/O Server DVD in the HMC drive). Example 3-30 shows the dialog from the **installios** command without options.

Example 3-30 Running installios on the HMC

```
hscroot@hmc2:~> installios
The following objects of type "managed system" were found.  Please
select one:
1. 520-DDEEC
Enter a number: 1
The following objects of type "virtual I/O server partition" were
found.  Please select one:
1. vios1
Enter a number: 1
The following objects of type "profile" were found.  Please select one:
1. default
Enter a number: 1
Enter the source of the installation images [/dev/cdrom]:
/export/resources/vios15
Enter the client's intended IP address: 9.3.5.114
Enter the client's intended subnet mask: 255.255.254.0
Enter the client's gateway: 9.3.5.129
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 U9111.520.10DDEEC-V1-C2-T1 b60f90001002 /vdevice/l-lan@30000002
virtual
2. ent U787A.001.DNZ00XK-P1-T5 00096b6b05a3
/pci@800000020000002/pci@2,4/ethernet@1  physical
3. ent U787A.001.DNZ00XK-P1-T6 00096b6b05a4
/pci@800000020000002/pci@2,4/ethernet@1,1  physical
```

```
4. ent U787A.001.DNZ00XK-P1-C2-T1 000d600a5926
/pci@800000020000002/pci@2,2/ethernet@1  physical

Enter a number (1-4): 2
Here are the values you entered:

managed system = 520-DDEEC
virtual I/O server partition = vios1
profile = default
source = /export/resources/vios15
IP address = 9.3.5.114
subnet mask = 255.255.254.0
gateway = 9.3.5.129
speed = auto
duplex = auto
configure network = yes
ethernet adapters = 00:09:6b:6b:05:a3

Press enter to proceed or type Ctrl-C to cancel...
```

Note: If you answer yes to configure the client network and you want to use this physical adapter to be part of a shared Ethernet adapter, SEA, you will have to detach this interface to configure SEA.

Tip: If you plan on using dual Virtual I/O Servers, it is practical to install and upgrade the first Virtual I/O Server to the desired level, then make a NIM backup and install the second Virtual I/O Server using NIM installation.

8. The system plan is ready to deploy as shown in the Summary window in Figure 3-59. Click **Deploy** and confirm by clicking **Yes** to start deploying the system plan.

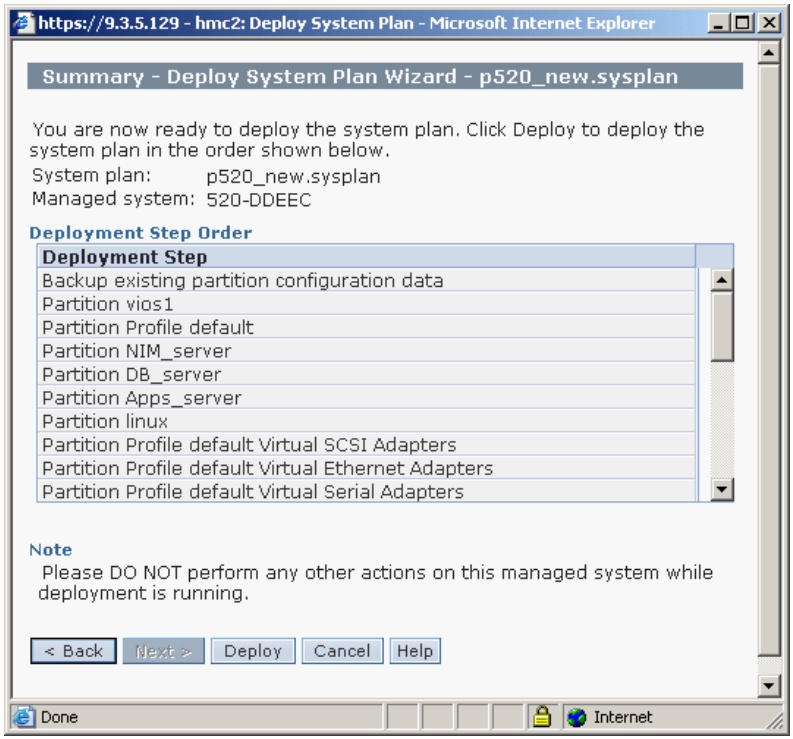


Figure 3-59 The Summary window

9. Deployment complete is shown in Figure 3-60.

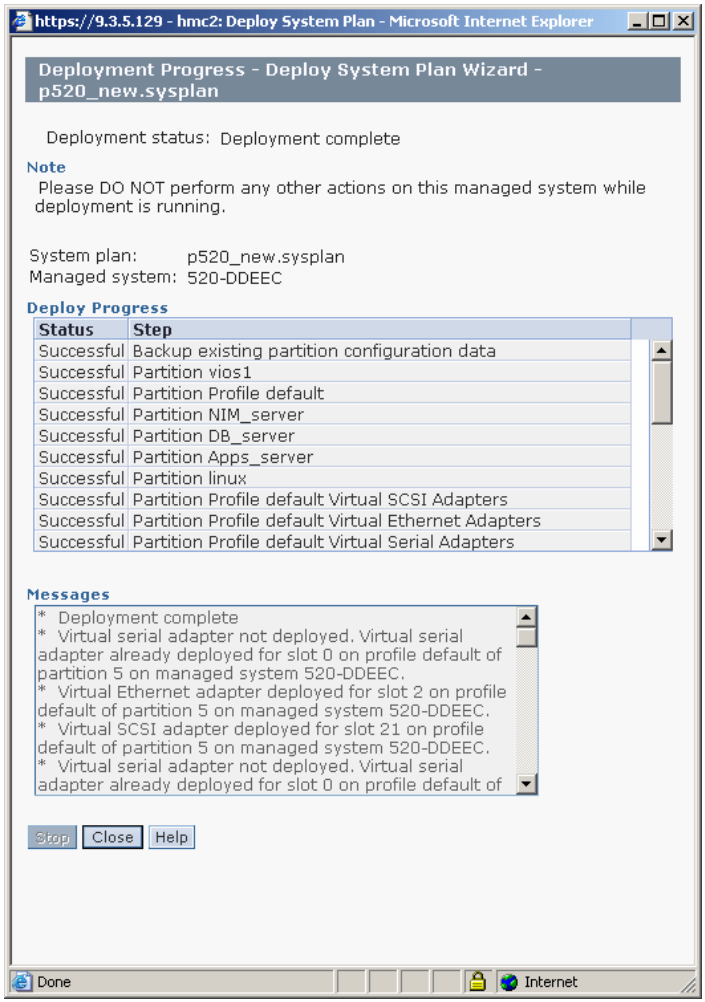


Figure 3-60 Deployment complete

Figure 3-61 shows the basic scenario deployed using SPT. The Virtual I/O Server is ready for installation and subsequent configuration of virtual devices.

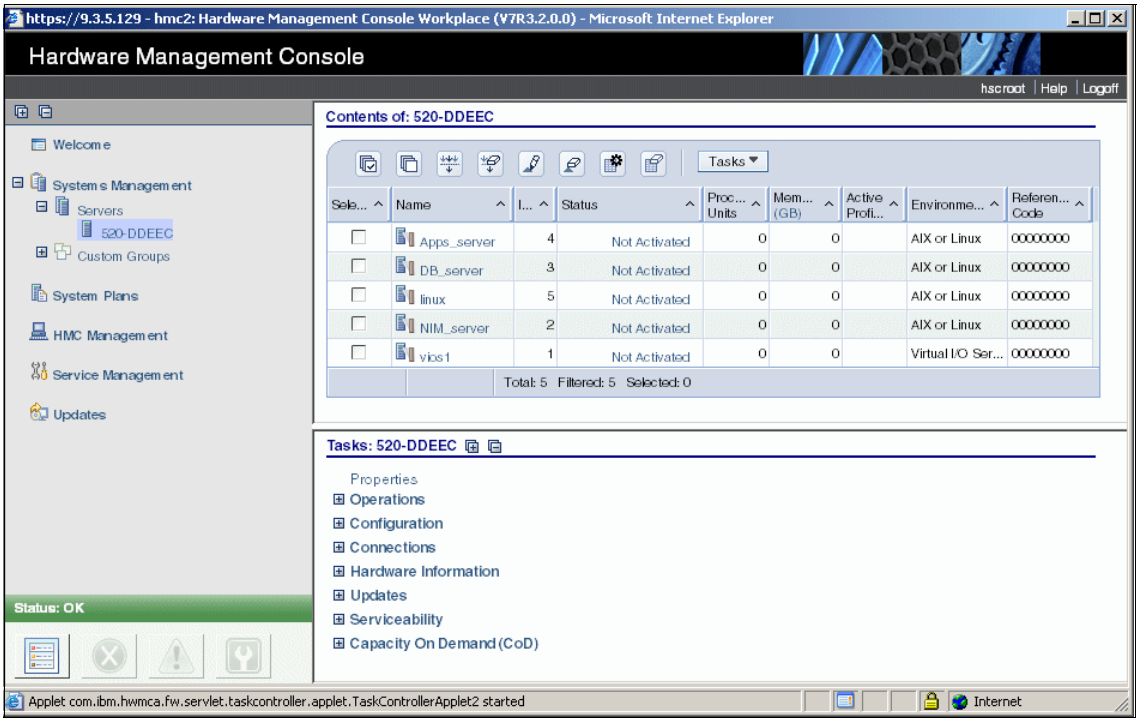


Figure 3-61 Basic scenario deployed from the system plan created in SPT

The Virtual I/O Server is now ready for installation and customization.

Note: At the time of writing the SPT does not have the possibility to create a server SCSI adapter with an “any client can connect” setting (one to many) like the SCSI adapter used for the virtual DVD. This adapter must be added after deployment.

For more information on how to deploy a system plan to a system, see “Partitioning the server” in the IBM Systems Hardware Information Center at:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphat/iphblsysplanoverview.htm

Creating an HMC system plan

On the HMC (or from IVM) you can generate a system plan to document your configuration. In the System Plan window, shown in Figure 3-62, you see the option to create a system plan.

Click **Create System Plan** to launch the menu shown in Figure 3-63 on page 238 and give the system plan a name. Click **Create** to start.

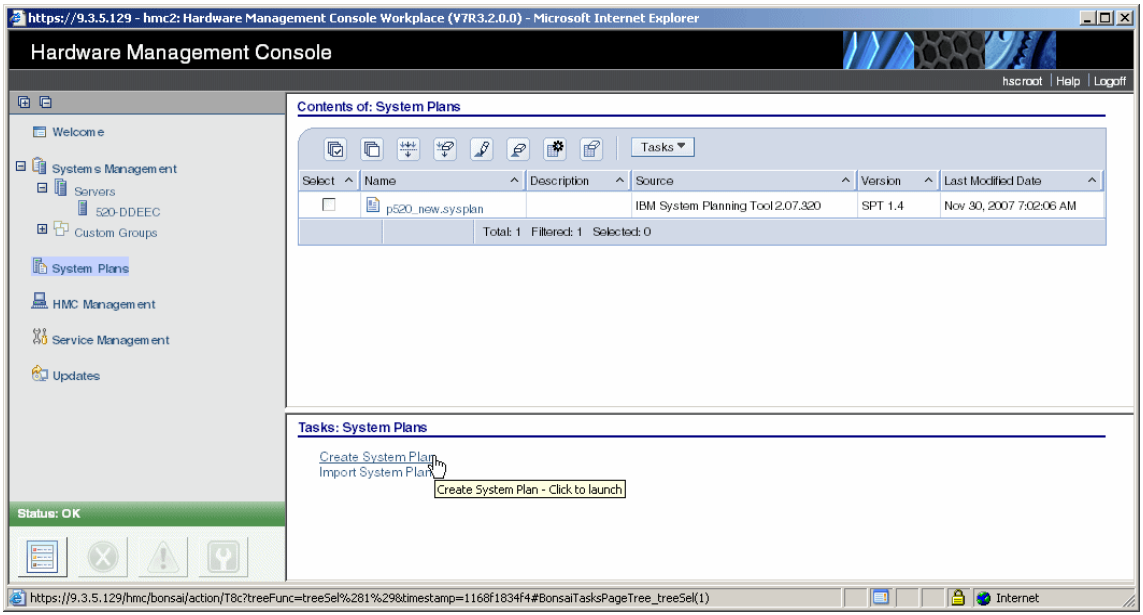


Figure 3-62 Creating an HMC system plan for documentation

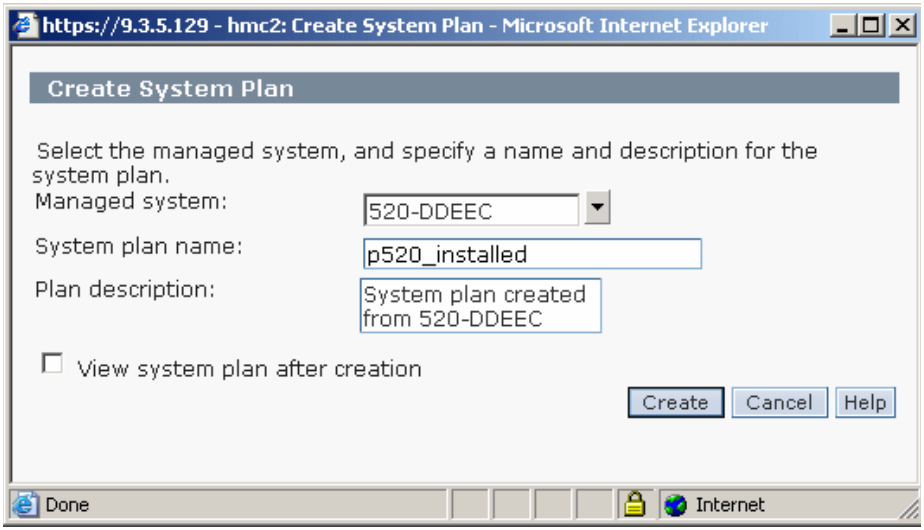


Figure 3-63 Giving a name to the system plan being created

A view of the created system plan is shown in Figure 3-64.

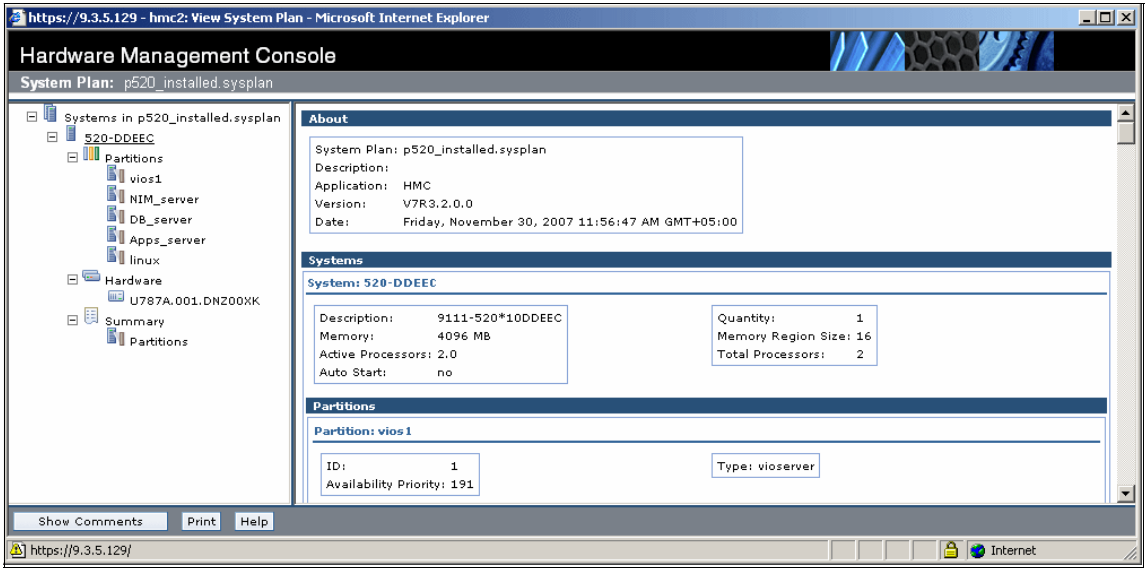


Figure 3-64 The created system plan

Figure 3-65 shows the back view of a POWER6 p570 server and its installed adapters.

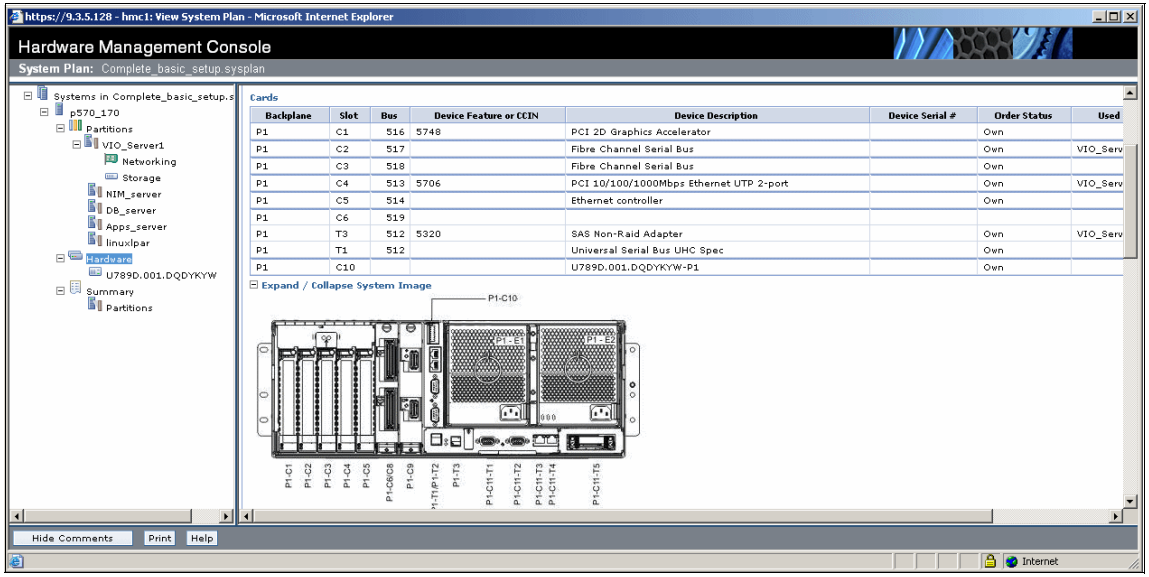


Figure 3-65 The back of the server and its installed adapters

Important: The HMC system plan can also be redeployed serving as a backup of the configuration.

Notes:

- It is still recommended to back up profile data in the Servers menu: **Configuration** → **Manage Partition Data** → **Backup**. Also, you should back up HMC data in the HMC Management menu.
- There is currently no discovery of devices belonging to an adapter such as disks on a SCSI adapter.

When the process is finished, you can select the newly created system plan; Figure 3-66 shows the menu options.

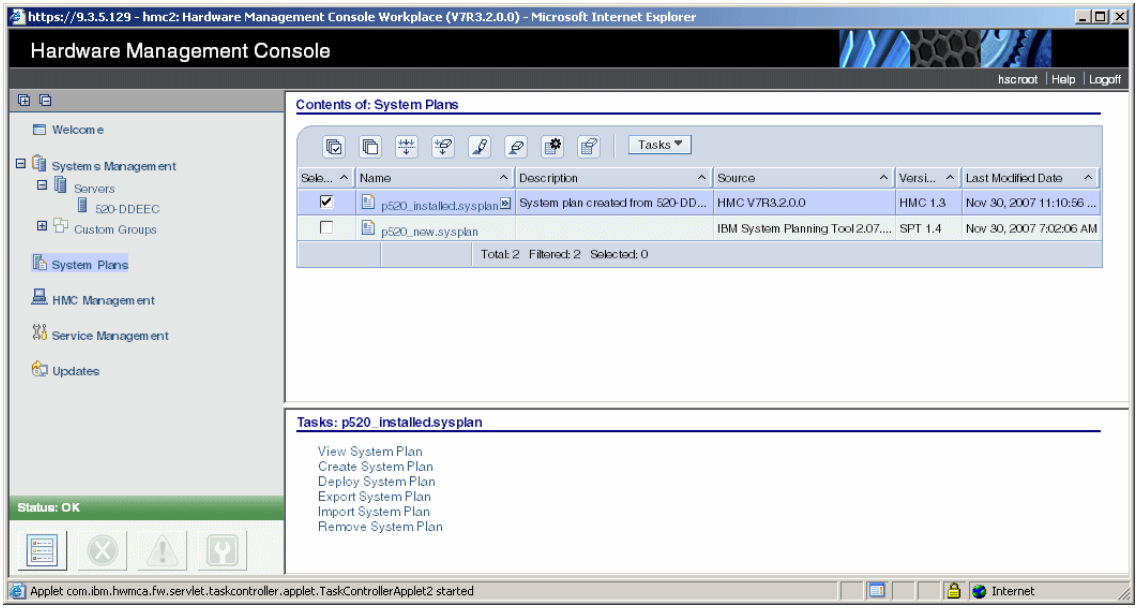


Figure 3-66 Options for the HMC system plan

A system plan can also be created with the HMC command **`mksysplan -m machine -f filename.sysplan`** as shown in Example 3-31.

Example 3-31 Creating a system plan using the command line

```
hscroot@hmc2:~> mksysplan -m 520-DDEEC -f p520_installed.sysplan
Gathering inventory ...
System plan p520_installed.sysplan created successfully for the system
520-DDEEC
```

Tip: If the create system plan fails, you can try the HMC command: **`mksysplan -m machine -f filename.sysplan -v -o noprobe`**

Specify noprobe to limit the inventory gathering to obtain only the PCI slot devices without any further inventory probes to active partitions, nor any refresh of inactive partition or unallocated hardware information.

Exporting an HMC system plan to SPT

An HMC system plan can be exported to another system and also back to SPT. When importing to SPT, the file must be converted to SPT format.

Notes:

1. This functionality was added in SPT 2.07.313.
2. The system plan must be converted when imported to SPT. The system plan usually requires some editing of storage devices to be valid in SPT.
3. The virtual SCSI server adapter for the DVD will not be available in SPT since the feature of having an adapter where any client can connect is currently not implemented.

Adding a partition in SPT to be deployed on the HMC

You can add a partition to the SPT system plan, import it to the HMC, and deploy it.

When deploying on the HMC the system plan is validated against the installed system and all existing partition profiles. Usually the HMC configuration will be further customized compared to the SPT system plan.

In this case the deployment can be done with the HMC command **deploysysplan** with the **-o d** option, as shown in Example 3-32 where validation against existing systems is omitted.

Example 3-32 System plan deployment with the deploysysplan command

```
hscroot@hmc2:~> deploysysplan -f p520_new_add_lpar.sysplan -m 520-DDEEC -o d
System plan 520-DDEEC was not completely deployed on managed system 520-DDEEC.
At least one planned operating environment that was part of the system plan was
not deployed. See previous messages to determine which planned operating
environments were not installed.
```

The new partition is added and the Virtual I/O Server, vios1, is updated with the virtual adapters required for the new profile.

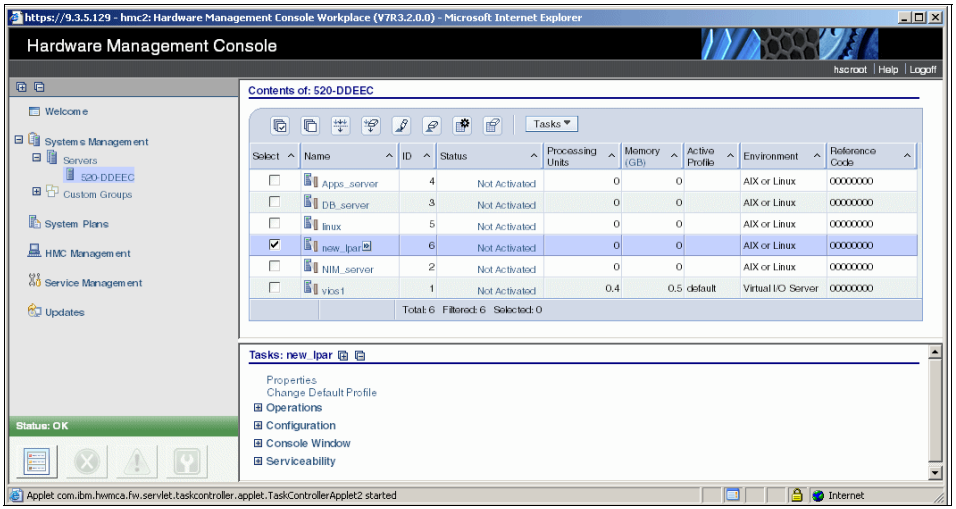


Figure 3-67 Added lpar using the system plan

4

Advanced virtualization configurations

This chapter starts with a discussion of advanced virtualization topics including:

- Providing continuous availability for virtual I/O

  This includes a discussion of how continuous availability for communication with external networks and virtualized storage can be achieved. The concepts are applied in four scenarios:

  - Scenario 1: Shared Ethernet Adapter failover
  - Scenario 2: Network Interface Backup in the client
  - Scenario 3: MPIO in the client with SAN
  - Scenario 4: LVM mirroring
- Installing and using Linux in a dual Virtual I/O Server configuration
- Configuring Multiple Shared-Processor Pools
- Extending VLANs into virtual networks
- Finally, supported configurations are listed, which concludes a section with some advice on the use of IBM TotalStorage® Solutions, HACMP, and GPFS™ in a virtualized environment.

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

4.1 Providing continuous availability for virtual I/O

This section provides a discussion of the requirements and concepts for providing continuous availability for virtual I/O. It shows how to use multiple Virtual I/O Servers for providing highly available virtual SCSI and shared Ethernet services to client partitions.

Several considerations are taken into account when deciding to use multiple Virtual I/O Servers. Client uptime, predicted I/O load averages for both network and storage, and system manageability are areas that need consideration when planning the Virtual I/O Server configuration.

For a small system with limited resources and limited I/O adapters, the system may not have enough resources for a second Virtual I/O Server. With limited I/O adapters, having a second Virtual I/O Server may adversely affect the overall performance provided to the clients if the additional adapters are used for increasing redundancy, not throughput.

For a larger system, there is a lower resource constraint and multiple Virtual I/O Server may be deployed without affecting overall client performance. More I/O adapters cater for both throughput and redundancy when used with additional Virtual I/O Servers.

The Virtual I/O Server is extremely robust. It mainly runs device drivers and does not run any application workloads, and regular users do not log in. Redundancy can be built into the Virtual I/O Server itself by using redundant physical adapters combined with MPIO and LVM mirroring (RAID) for storage devices and Link Aggregation for network devices.

The main reason to implement two Virtual I/O Severs is to provide continuous access when a Virtual I/O Server has to be taken offline for planned outages such as software updates (see 4.4, “System maintenance with dual Virtual I/O Servers” on page 260). In a two Virtual I/O Server configuration virtual SCSI and shared Ethernet can be configured in a redundant fashion allowing one Virtual I/O Server to be rebooted for maintenance or even reinstalled without affecting the virtual I/O clients.

Note: IVM supports a single Virtual I/O Server.

The following sections discuss these topics in detail:

- 4.2, “Virtual SCSI redundancy” on page 245
- 4.3, “Shared Ethernet redundancy” on page 251
- 4.4, “System maintenance with dual Virtual I/O Servers” on page 260

- 4.5, “Separating and balancing disk and network traffic” on page 262

Note: Basically, the concepts described apply both to AIX and Linux client partitions. The practical examples are mostly based on AIX. However, there is a whole section covering the technical implementation on Linux in 4.11, “Using Linux on POWER with two Virtual I/O Servers” on page 299.

4.2 Virtual SCSI redundancy

Virtual SCSI redundancy can be achieved using MPIO and LVM mirroring at client partition and Virtual I/O Server level.

Figure 4-1 on page 246 displays an advanced setup using both MPIO and LVM mirroring in the client partition at the same time. Two Virtual I/O Servers host disks for a client partition. The client is using MPIO to access a SAN disk and LVM mirroring to access two SCSI disks. From the client perspective, the following situations could be handled without causing downtime for the client:

- Either path to the SAN disk could fail, but the client would still be able to access the data on the SAN disk through the other path. No action has to be taken to reintegrate the failed path to the SAN disk after repair if MPIO is configured as described in 4.8.2, “Configuring MPIO in the client partitions” on page 282.
- The failure of a SCSI disk would cause stale partitions for the volume group with the assigned virtual disks. The client partition would still be able to access the data on the second copy of the mirrored disk. After the failed disk is available again the stale partitions have to be synchronized using the **varyonvg** command.
- Either Virtual I/O Server could be rebooted for maintenance. This would result in a temporary simultaneous failure of one path to the SAN disk and stale partitions for the volume group on the SCSI disks, as described before.

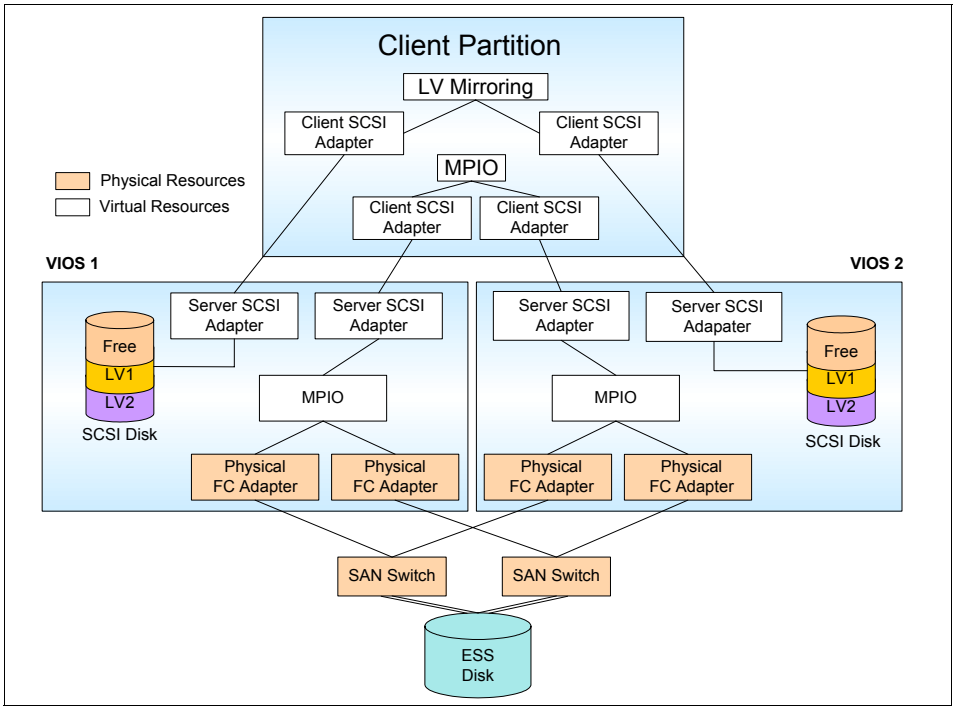


Figure 4-1 Virtual SCSI redundancy using MPIO and LVM mirroring

Notes:

- If mirroring and MPIO are both configurable in your setup, MPIO is the preferred method for adding disk redundancy to the client. LVM mirroring causes stale partitions that require synchronization, while MPIO does not. Depending on the RAID level used on the SAN disks the disk space requirements for LVM mirroring can be higher.
- It is recommend to use two Fibre Channel adapters in each Virtual I/O Server for adapter redundancy.

For further examples of virtual SCSI configurations, see 4.14.1, “Supported virtual SCSI configurations” on page 331.

4.2.1 MPIO in the client partition

With MPIO in the client partition each Virtual I/O Server can present a virtual SCSI device that is physically connected to the same physical disk. This

achieves redundancy for the Virtual I/O Server itself and for any adapter, switch, or device that is used between the Virtual I/O Server and the disk.

The MPIO for virtual SCSI devices 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 when both paths are enabled.

To balance the load of multiple client partitions across two 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 using the **chpath** command as shown in the example below (1 is the highest priority):

```
chpath -l hdisk0 -p vscsi0 -a priority=2
```

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 Virtual I/O Server configurations between primary and backup.

For MPIO support of virtual SCSI devices in the AIX 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 two Virtual I/O Servers. There should also be no volume group created on the SAN LUNs on the Virtual I/O Server. 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.

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.

The heartbeat check interval for each disk using MPIO should be configured so that the path status is updated automatically. This is done using the **chdev** command. as shown in the example below. Specifying hcheck_mode=nonactive

means that healthcheck commands are sent down paths that have no active I/O, including paths with a state of failed. The `hcheck_mode=nonactive -P` attribute defines how often the health check is performed.

```
chdev -l hdisk0 -a hcheck_interval=20 -a hcheck_mode=nonactive -P
```

Note: Because the attribute cannot be changed while the vsci device is in the active state, the -P flag is used so that the change is made in the ODM only. The changes are applied to the device when the system is rebooted.

On the virtual SCSI client adapter the virtual SCSI adapter path timeout feature should be configured. It allows the client adapter to check the health of Virtual I/O Server servicing a particular adapter and detect if a Virtual I/O Server is not responding to I/O requests. In such a case the client will failover to an alternate path if the Virtual SCSI adapter path timeout is configured. This feature is configured using the **chdev** command as shown in the following example.

```
chdev -l vscsi0 -a vscsi_path_to=30 -P
```

Figure 4-2 on page 249 shows a configuration where MPIO is used in the client partition. The Virtual I/O Server is using SAN storage used the SVC and the SDDPCM module. It shows which attributes should be set in this configuration. MPIO in the client partition can be used without configuring MPIO on the Virtual I/O Server. This would be the case if each Virtual I/O Server has just a single Fibre Channel adapter configured. The settings when using multiple Fibre Channel adapters and MPIO on the Virtual I/O Server level are described in detail in 4.8.2, “Configuring MPIO in the client partitions” on page 282.

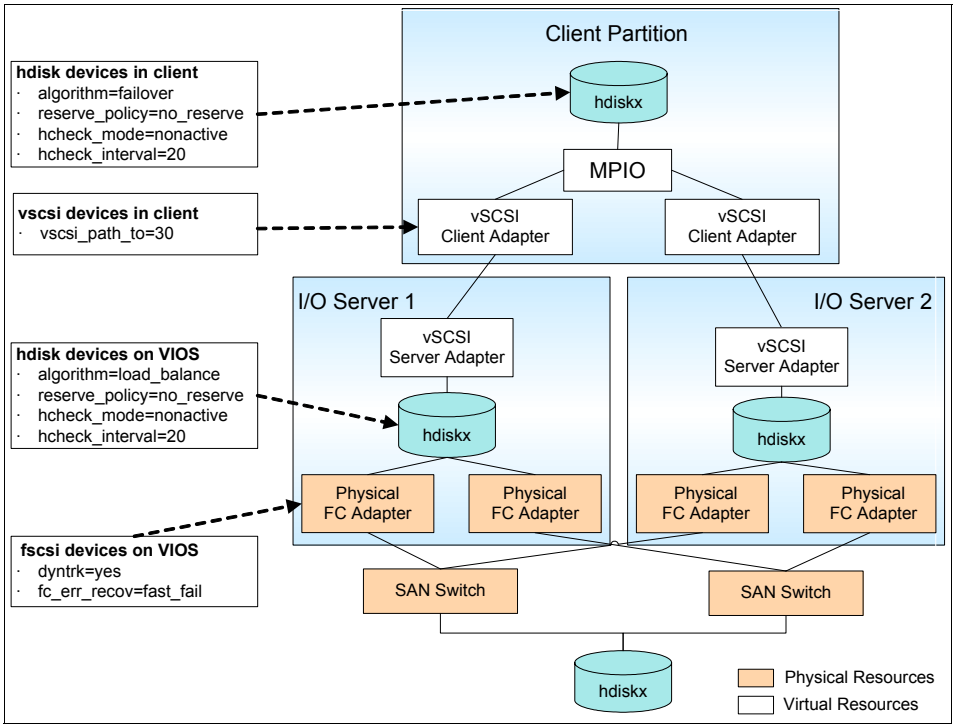


Figure 4-2 MPIO attributes

4.2.2 MPIO in the Virtual I/O Server

This section describes the configuration of the Virtual I/O Server when deploying MPIO in a dual Virtual I/O Servers environment.

Fibre channel 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 attribute. 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.

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 more information on supported configurations see 4.14.1, “Supported virtual SCSI configurations” on page 331 and the support matrix for all IBM and third-party storage solutions for the Virtual I/O Server which can be found at the following Web site:

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

Important: Third-party storage vendors provide their own recommendations relating to the installation of their storage devices on AIX and the Virtual I/O Server. Refer to the vendor-specific installation guides for configuration instructions.

hdisk device configuration on the Virtual I/O Server

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 Fibre Channel adapters within the Virtual I/O Servers when using the base AIX MPIO support, set the algorithm to `round_robin` for each physical disk as shown in Example 4-1. This is not required when you use SDD or SDDPCM as multipathing software:

Example 4-1 Changing disks to round_robin algorithm

```
$ chdev -dev hdisk1 -attr algorithm=round_robin
hdisk1 changed
$ 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
```

4.2.3 LVM mirroring in the client partition

With the use of LVM mirroring in the client partition, each Virtual I/O Server can present a virtual SCSI device that is physically connected to a different disk and then used in a normal AIX mirrored volume group on the client partition. This achieves a higher level of reliability by providing redundancy. Client volume group mirroring is also required when a Virtual I/O Server logical volume is used as a virtual SCSI device on the client. In this case, the virtual SCSI devices are associated with different SCSI disks, each controlled by one of the two Virtual I/O Server. You can find a sample LVM mirroring configuration in 4.9, “Scenario 4: LVM mirroring” on page 288

4.3 Shared Ethernet redundancy

In a single Virtual I/O Server configuration, communication to external networks ceases if the Virtual I/O Server looses connection to the external network. Client partitions will experience this disruption if they use the SEA as a means to access the external networks. Communication through the SEA is, for example, suspended when the physical network adapter in the Virtual I/O Server fails or loses connectivity to the external network because of a switch failure. Another reason might be a planned shutdown of the Virtual I/O Server for maintenance purposes. Communication resumes as soon as the Virtual I/O Server regains connectivity to the external network. Internal communication between partitions through virtual Ethernet connections continues unaffected while access to the external network is unavailable. VIO clients do not have to be rebooted or otherwise reconfigured to resume communication through the SEA. The clients are similarly affected as when unplugging and replugging an uplink of a physical Ethernet switch.

If the temporary failure of communication with external networks is unacceptable, more than a single forwarding instance and some function for failover has to be implemented in the Virtual I/O Server.

Consideration: The Integrated Virtualization Manager (IVM) supports a single Virtual I/O Server. This section only applies to systems managed by the Hardware Management Console (HMC).

There are several approaches to achieve continuous availability for shared Ethernet access to external networks. Most commonly used are Shared Ethernet Adapter Failover and Network Interface Backup which are described in detail in the following paragraphs.

There are other approaches to achieve continuous availability for shared Ethernet access by taking advantage of configurations that are also used in physical network environments. These include:

- IP Multipathing with Dead Gateway Detection (DGD) or Virtual IP Addresses (VIPA) and dynamic routing protocols, such as Open Shortest Path First (OSPF).
- IP Address Takeover (IPAT), with High Availability Cluster Management or Automation Software, such as HACMP for AIX or Tivoli System Automation (TSA).

4.3.1 Shared Ethernet Adapter failover

Shared Ethernet Adapter failover offers Ethernet redundancy to the client at the Virtual I/O Server level. In a SEA failover configuration two Virtual I/O Servers have the bridging functionality of the Shared Ethernet Adapter. They use a control channel to determine which of them is supplying the Ethernet service to the client. If one SEA loses access to the external network through its physical Ethernet adapter or one Virtual I/O Server is shut down for maintenance it will automatically fail over. You can also trigger a manual failover.

The client partition gets one virtual Ethernet adapter bridged by two Virtual I/O Servers. The client partition has no special protocol or software configured and uses the virtual Ethernet adapter as though it was bridged by only one Virtual I/O Server.

Shared Ethernet Adapter failover supports IEEE 802.1Q VLAN tagging just like the basic SEA feature.

As shown in Figure 4-3 on page 253, both Virtual I/O Servers attach to the same virtual and physical Ethernet networks and VLANs, and both virtual Ethernet adapters of both SEAs will have the *access external network* flag enabled and a *trunk priority* set. An additional virtual Ethernet connection has to be set up as a separate VLAN between the two Virtual I/O Servers and must be attached to the Shared Ethernet Adapter (SEA) as a *control channel*, not as regular member of

the SEA. This VLAN serves as a channel for the exchange of keep-alive or heartbeat messages between the two Virtual I/O Servers and therefore controls the failover of the bridging functionality. No network interfaces have to be attached to the control channel Ethernet adapters. The control channel adapter should be placed on a dedicated VLAN that is not used for anything else.

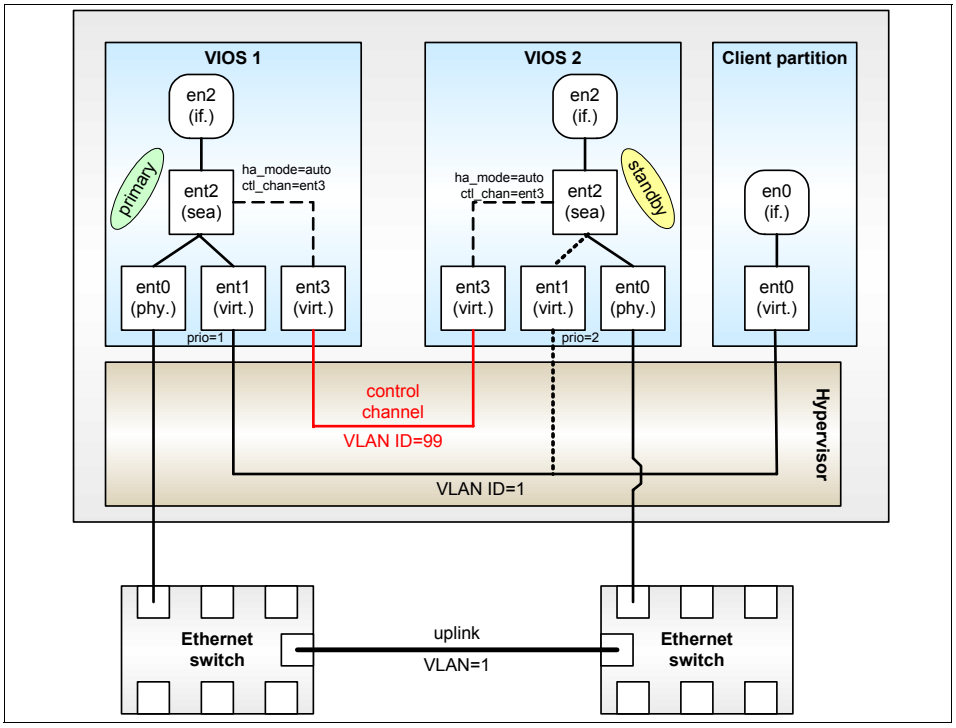


Figure 4-3 Basic SEA failover configuration

You must select different priorities for the two SEAs by setting all virtual Ethernet adapters of each SEA to that priority value. The priority value defines which of the two SEAs will be the primary (active) and which will be the backup (standby). The lower the *priority value*, the higher the priority, thus priority=1 means highest priority.

There may also be some types of network failures that would not trigger a failover of the SEA, because keep-alive messages are only sent over the control channel. No keep-alive messages are sent over other SEA networks, especially not over the external network. The SEA failover feature can be configured to periodically check the reachability of a given IP address. The SEA will periodically ping this IP address, so it can detect some other network failures. This is similar to the IP address to ping that may be configured with Network Interface Backup.

Important: The Shared Ethernet Adapters must have network interfaces, with IP addresses associated, to be able to use this periodic reachability test.

These IP addresses have to be unique and you have to use different IP addresses on the two SEAs.

There are basically four different cases that will initiate a SEA failover:

1. The standby SEA detects that keep-alive messages from the active SEA are no longer received over the control channel.
2. The active SEA detects that a loss of the physical link is reported by the physical Ethernet adapter's device driver.
3. On the Virtual I/O Server with the active SEA, a manual failover can be initiated by setting the active SEA to standby mode.
4. The active SEA detects that it cannot ping a given IP address anymore.

An end of the keep-alive messages would occur when the Virtual I/O Server with the primary SEA is shut down or halted, has stopped responding, or has been deactivated from the HMC.

Important: You may experience up to a 30-second delay in failover when using SEA failover. The behavior depends on the network switch and the spanning tree settings.

Figure 4-4 shows an alternative setup where the IP address of the Virtual I/O Servers is configured on a separate physical Ethernet adapter. In this configuration it is not possible for the SEAs to periodically ping a given IP address. It is recommended to associate the interface and IP address to the SEA, as shown in Figure 4-3 on page 253.

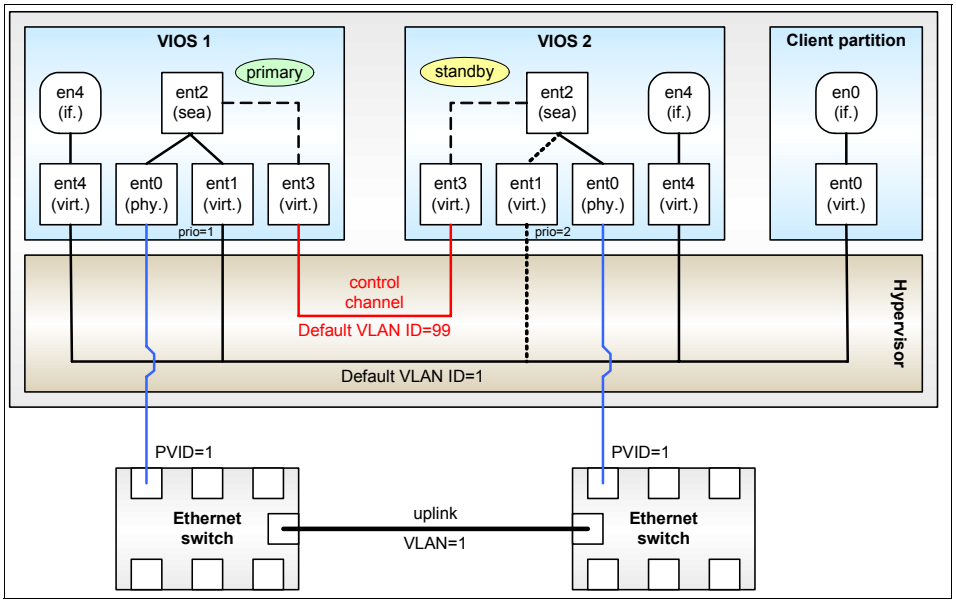


Figure 4-4 Alternative configuration for SEA failover

4.3.2 Network Interface Backup in the client partition

Network Interface Backup (NIB) in the client partition can be used to achieve network redundancy when using two Virtual I/O Severs. An EtherChannel with only one primary adapter and one backup adapter is said to be operating in Network Interface Backup mode.

The client partition uses two virtual Ethernet adapters to create an EtherChannel that consists of one primary adapter and one backup adapter. The interface is defined on the EtherChannel. If the primary adapter becomes unavailable, the Network Interface Backup switches to the backup adapter. See 4.7, “Scenario 2: Network Interface Backup in the client” on page 270 for setup details. Figure 4-5 on page 256 shows the concept.

A Link Aggregation of more than one active virtual Ethernet adapter is not supported. Only one primary virtual Ethernet adapter plus one backup virtual Ethernet adapter are supported. To increase the bandwidth of a virtual Ethernet

adapter Link Aggregation has to be done on the Virtual I/O Server as described in 4.3.4, “Using Link Aggregation on the Virtual I/O Server” on page 258.

When configuring NIB in a client partition, each virtual Ethernet adapter has to be configured on a different VLAN. It is not possible to configure additional VLANs. The two different internal VLANs are then bridged to the same external VLAN, as shown in Figure 4-5.

,

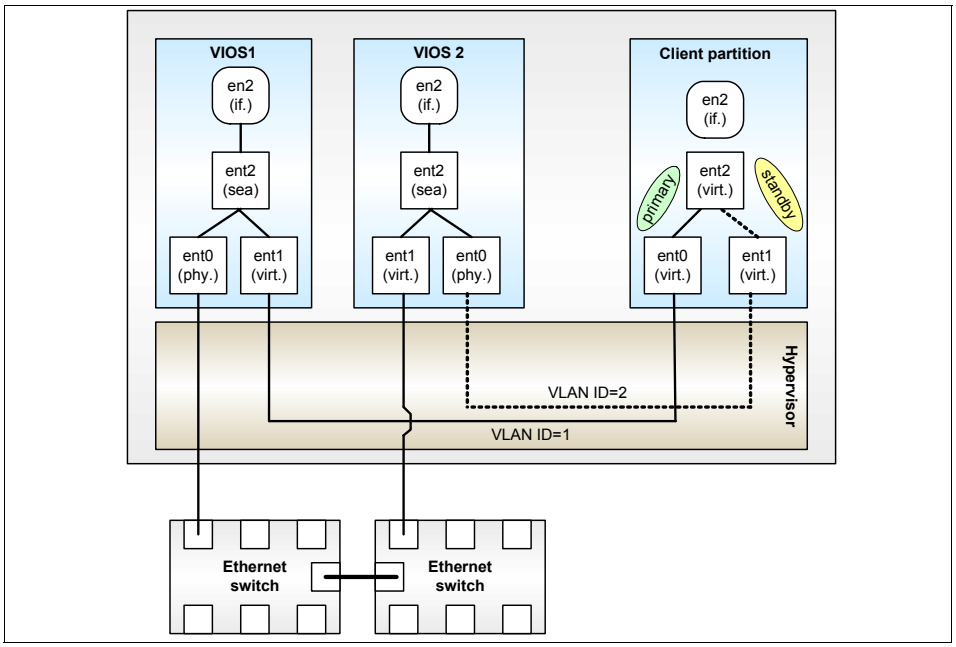


Figure 4-5 Network redundancy using two Virtual I/O Servers and NIB

Important: When using NIB with virtual Ethernet adapters, it is mandatory to use the ping-to-address feature to be able to detect network failures, because there is no hardware link failure for virtual Ethernet adapters to trigger a failover to the other adapter.

4.3.3 When to use SEA failover or Network Interface Backup

In the previous sections, we described two different approaches on how to provide high availability for bridged (layer-2) access to external networks with redundant Shared Ethernet Adapters for two Virtual I/O Servers. The following sections discuss which approach to use when.

Advantages of Network Interface Backup

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

- 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.
- 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.

When to use Network Interface Backup

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

- You want to load balance client partitions to use both Virtual I/O Servers.
- You are not using VLAN tagging.
- You have an existing Network Interface Backup configuration and do not want to change the Ethernet setup.

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.

- Only the SEAs send out periodic ping requests for checking the network availability. With NIB every client partition will send out ping requests resulting in more network traffic.

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.

Note: SEA failover and NIB have a common behavior: They do not check the reachability of the specified IP address through the backup-path as long as the primary path is active. That is because the virtual Ethernet adapter is always connected and there is no *link up* event as is the case with physical adapters. You do not know if you really have an operational backup until your primary path fails.

4.3.4 Using 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 can be used on the Virtual I/O Server to increase the bandwidth compared to when using a single network adapter and avoid bottlenecks when sharing one network adapter among many client partitions.

The main benefit of a Link Aggregation is that it has the network bandwidth of all of its adapters in a single network presence. If an adapter fails, the packets are automatically sent on the next available adapter without disruption to existing user connections. The adapter is automatically returned to service on the Link Aggregation when it recovers. Thus, Link Aggregation also provides some degree of increased availability. A link or adapter failure will lead to a performance degradation, but not a disruption.

Depending on the manufacturer, Link Aggregation is not a complete high-availability networking solution because all the aggregated links must connect to the same switch. By using a backup adapter you can add a single

additional link to the Link Aggregation, which is connected to a different Ethernet switch with the same VLAN. This single link will only be used as a backup.

As an example for Link Aggregation, ent0 and ent1 can be aggregated to ent2. The system considers these aggregated adapters as one adapter. Interface en2 would then be configured with an IP address. Therefore, IP is configured as on any other Ethernet adapter. In addition, all adapters in the Link Aggregation are given the same hardware (MAC) address, so they are treated by remote systems as though they were one adapter.

Two variants of Link Aggregation are supported:

- Cisco EtherChannel (EC)
- IEEE 802.3ad Link Aggregation (LA)

While EC is a Cisco-specific implementation of adapter aggregation, LA follows the IEEE 802.3ad standard. Table 4-1 shows the main differences between EC and LA.

Table 4-1 Main differences between EC and LA aggregation

Cisco EtherChannel	IEEE 802.3ad Link Aggregation
Cisco-specific	Open Standard
Requires switch configuration.	Little, if any, configuration of switch required to form aggregation. Some initial setup of the switch may be required.
Supports different packet distribution modes.	Supports only standard distribution mode.

The main benefit of using LA is, that if the switch supports the *Link Aggregation Control Protocol* (LACP), no special configuration of the switch ports is required. The benefit of EC is the support of different packet distribution modes. This means it is possible to influence the load balancing of the aggregated adapters. In the remainder of this book, we use Link Aggregation where possible since that is considered a more universally understood term.

Figure 4-6 on page 260 shows the aggregation of two plus one adapters to a single pseudo-Ethernet device, including a backup feature. The Ethernet adapters ent0 and ent1 are aggregated for bandwidth and must be connected to the same Ethernet switch, while ent2 connects to a different switch, but is only used for backup, for example, if the main Ethernet switch fails. The adapters ent0 and ent1 are now exclusively accessible through the pseudo Ethernet adapter ent5 and its interface en5. You could not, for example, attach a network interface en0 to ent0, as long as ent0 is a member of an EtherChannel or Link Aggregation.

Note: A Link Aggregation or EtherChannel of virtual Ethernet adapters is not supported. But you may use the Network Interface Backup feature of Link Aggregation with virtual Ethernet adapters.

A Link Aggregation with only one primary Ethernet adapter and one backup adapter is said to be operating in Network Interface Backup (NIB).

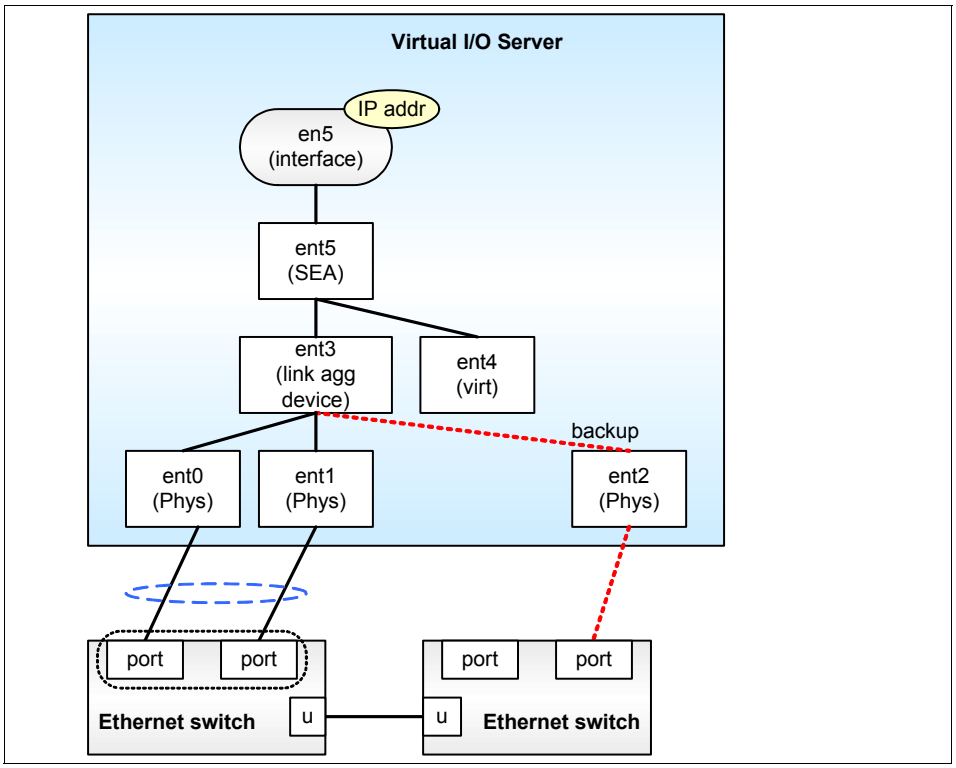


Figure 4-6 Link Aggregation (EtherChannel) on the Virtual I/O Server

4.4 System maintenance with dual Virtual I/O Servers

Redundancy allows for easier system management, such as software upgrades. With two Virtual I/O Servers, the layer of physical resource dependency has been removed.

System maintenance can now be performed on a Virtual I/O Server and any external device it connects to, such as a network or SAN switch. With virtual SCSI and shared Ethernet being hosted by a second Virtual I/O Server,

rebooting or disconnecting the Virtual I/O Server from its external devices is now possible without causing client outages. With the client partition running MPIO and using SEA failover, no actions will need to be performed on the client partition while the system maintenance is being performed or after it has completed. This results in improved uptime and reduced system administration efforts for the client partitions.

Upgrading and rebooting a Virtual I/O Server, network switch, or SAN switch is simpler and more compartmentalized, since the client is no longer dependant on the availability of all of the environment.

In Figure 4-7, a client partition has virtual SCSI devices and a virtual Ethernet adapter hosted from two Virtual I/O Servers. The client has MPIO implemented across the virtual SCSI devices and Shared Ethernet Adapter Failover for the virtual Ethernet.

Note: A combination of MPIO for disk redundancy and Shared Ethernet Adapter Failover for network redundancy is recommended.

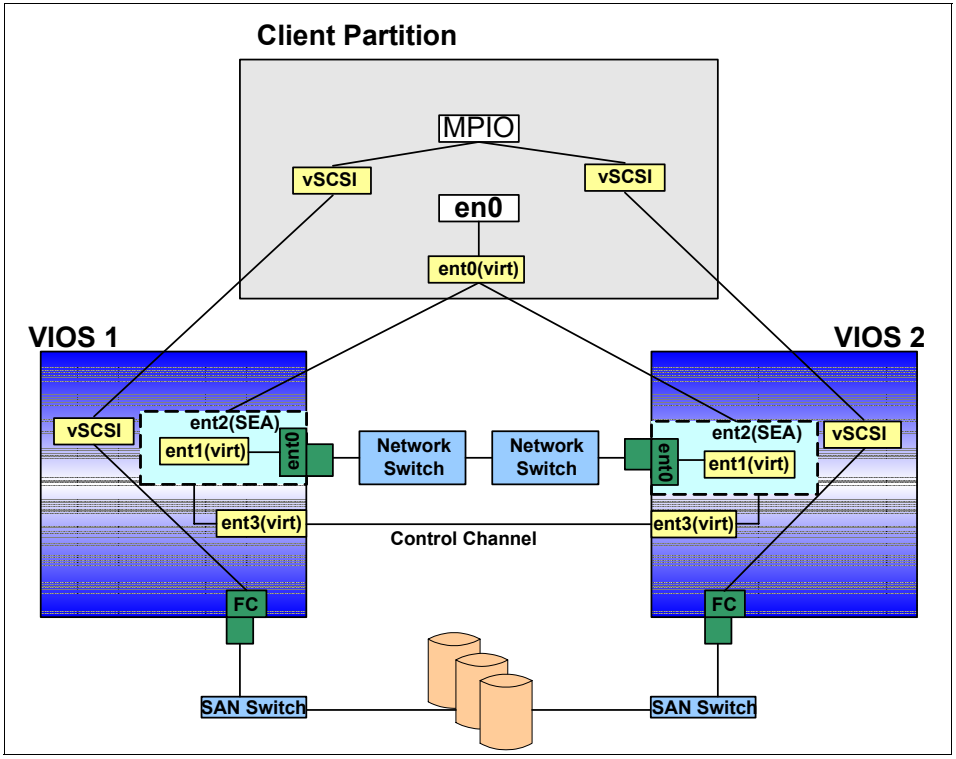


Figure 4-7 Redundant Virtual I/O Servers before maintenance

When Virtual I/O Server 2 is shut down for maintenance, as shown in Figure 4-8, the client partition continues to access the network and SAN storage through Virtual I/O Server 1.

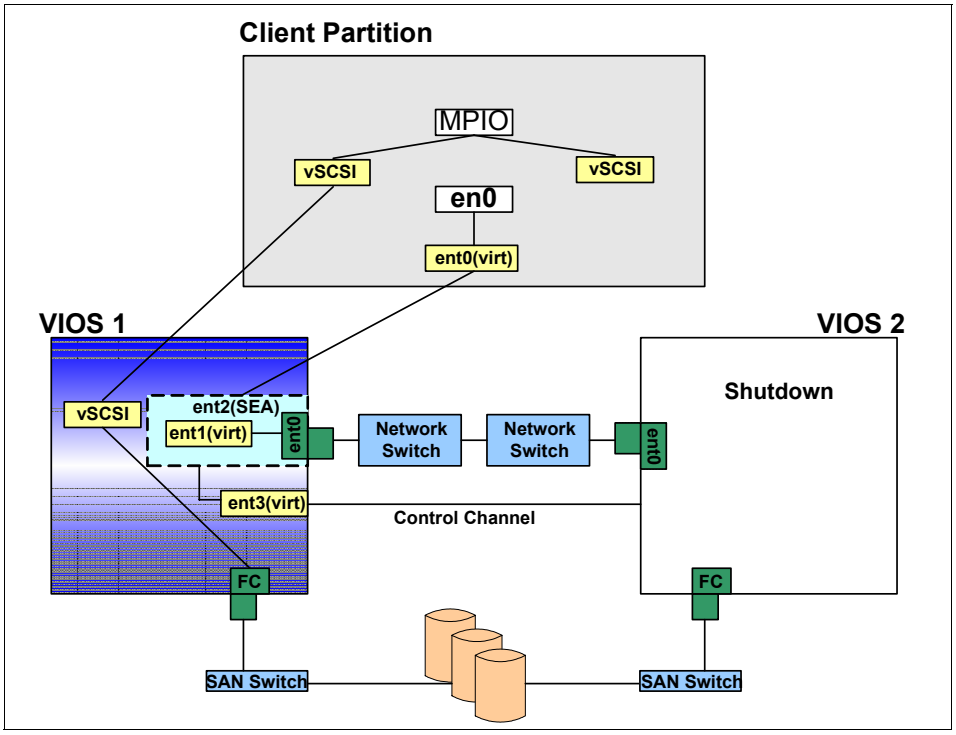


Figure 4-8 Redundant Virtual I/O Servers during maintenance

When Virtual I/O Server 2 returns to a full running state, the client will continue using the MPIO path through Virtual I/O Server 1, while the virtual Ethernet will be reset to the primary Virtual I/O Server Shared Ethernet Adapter.

4.5 Separating and balancing disk and network traffic

In addition to continuous availability, a dual Virtual I/O Server setup also provides the ability to separate or balance the virtual I/O load and the resulting resource consumption across the Virtual I/O Servers.

Virtual Ethernet traffic is generally heavier on the Virtual I/O Server than virtual SCSI traffic. Virtual Ethernet connections generally take up more CPU cycles than connections through physical Ethernet adapters. The reason is that modern physical Ethernet adapters contain many functions to off-load some work from

the system's CPUs, for example, checksum computation and verification, interrupt modulation, and packet reassembly.

In a configuration running MPIO and a single SEA per Virtual I/O Server you typically separate the traffic in such a way that the virtual Ethernet traffic is going through one Virtual I/O Server and the virtual SCSI traffic is going through the other. This is done by defining the SEA trunk priority and the MPIO path priority. See 4.6, "Scenario 1: Shared Ethernet Adapter failover" on page 264 for details on SEA failover configuration and 4.8, "Scenario 3: MPIO in the client with SAN" on page 275 for details on MPIO configuration.

Note: Do not switch off SEA threading on the Virtual I/O Server that is used as primary for the network traffic. It will do both disk and network serving in case the other Virtual I/O Server is unavailable.

Tip: By default MPIO traffic is normally routed through the first Virtual I/O Server. The easiest way to separate disk and network traffic is therefore to define the Shared Ethernet Adapter to use the second Virtual I/O Server as primary channel. Like the path priorities for the virtual hdisk can be left unchanged. Otherwise you have to update the path priorities for each hdisk in each client partition.

In an MPIO configuration with several Shared Ethernet Adapters per Virtual I/O Server you typically balance the network and virtual SCSI traffic between the Virtual I/O Servers.

Note: Use the storage configuration software to check that the preferred paths on the storage subsystem are in accordance with the path priorities set in the virtual I/O clients.

When using MPIO together with NIB the virtual Ethernet traffic can be spread more flexibly. For each network interface running in NIB mode you can configure through which Virtual I/O server the traffic should go. If your network traffic is driving one Virtual I/O Server to its limits, you can balance the workloads by adapting the NIB configuration.

If the Virtual I/O Servers are running in uncapped mode, processor resources are automatically balanced.

See 4.7, "Scenario 2: Network Interface Backup in the client" on page 270 for details on configuring NIB and 4.8, "Scenario 3: MPIO in the client with SAN" on page 275 for details configuring MPIO.

Figure 4-9 shows an example configuration where network and disk traffic are separated.

- Virtual I/O Server 1 has priority 1 for network and priority 2 for disk.
- Virtual I/O Server 2 has priority 2 for network and priority 1 for disk.

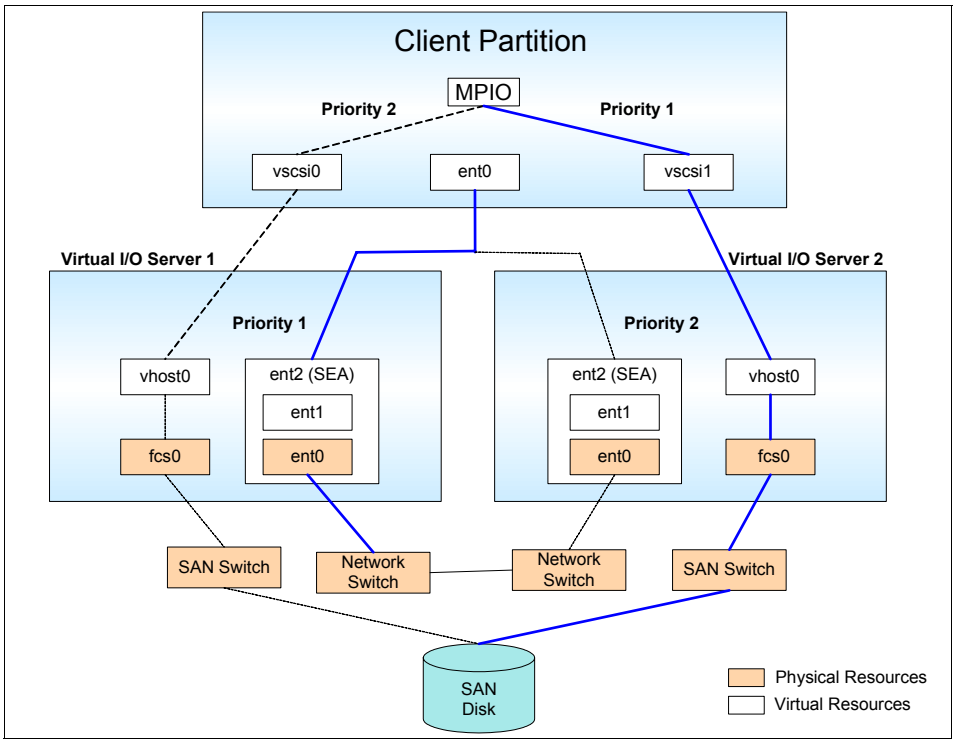


Figure 4-9 Separating disk and network traffic

4.6 Scenario 1: Shared Ethernet Adapter failover

This scenario shows how to set up a highly available Shared Ethernet Adapter configuration using Shared Ethernet Adapter failover. Figure 4-10 on page 265 shows the configuration. To keep the picture simple, only the client partition DB_Server is shown in the picture. In a Shared Ethernet Adapter failover configuration there are two Virtual I/O servers, each running a Shared Ethernet Adapter. The Shared Ethernet Adapters communicate with each other on a control channel using two virtual Ethernet adapters configured on a separate VLAN. The control channel is used to carry heartbeat packets between the two Shared Ethernet Adapters. When the primary Shared Ethernet Adapter loses

connectivity or the Virtual I/O Server is shut down for maintenance, the network traffic is automatically switched to the backup Shared Ethernet Adapter.

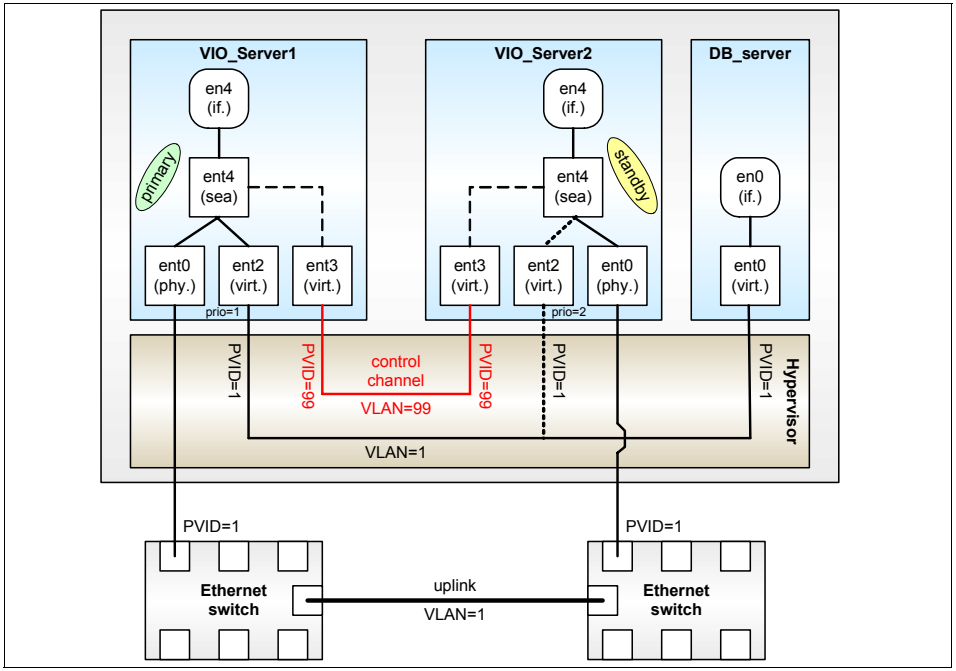


Figure 4-10 Highly available Shared Ethernet Adapter setup

4.6.1 Configuring Shared Ethernet Adapter failover

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 3.3, “Creating a Virtual I/O Server partition” on page 153. In step 10, select one physical Ethernet adapter and one physical storage adapter.
2. Install both Virtual I/O Servers by following the instructions in 3.4, “Virtual I/O Server software installation” on page 173.
3. Configure each Virtual I/O Server with two virtual Ethernet adapters. One virtual Ethernet adapter is configured on VLAN 1 and is used for network traffic to the external network. The second virtual Ethernet adapter is on VLAN 99 and is used for heartbeat traffic between to the SEAs. Table 4-2 on page 266 shows an overview of the required virtual Ethernet adapters. The trunk priority for the Virtual Ethernet adapter on VIO_Server1 which has the Access external network flag set is set to 1. This means that normally the network traffic will go through VIO_Server1. VIO_Server2 with trunk priority 2

is used as backup in case VIO_Server1 has no connectivity to the external network.

Note: The two adapters used for SEA heartbeat must be on the same VLAN. The VLAN must be the same for both adapters and must not be used by any other adapter on the machine.

Table 4-2 Virtual Ethernet adapter overview for Virtual I/O Servers

Virtual I/O Server	Virtual I/O Server slot	VLAN ID	Trunk priority	Access external network
VIO_Server1	10	1	1	yes
VIO_Server1	11	99	0	no
VIO_Server2	10	1	2	yes
VIO_Server2	11	99	0	no

4. Also make the virtual SCSI configuration highly available by either configuring LVM mirroring as described in 4.8, “Scenario 3: MPIO in the client with SAN” on page 275 MPIO as described in 4.9, “Scenario 4: LVM mirroring” on page 288.
5. Define the SEA adapter device on VIO_Server1 and VIO_Server2 using the **mkdev** command, as shown in Example 4-2. The SEA is defined with an IP address to ping using the -netaddr flag. The SEA will periodically ping this IP address, so it can detect some other network failures. This is similar to the IP address to ping that may be configured with Network Interface Backup. In this example the SEA is also configured with the largesend option to enable TCP segmentation offload.

Example 4-2 Shared Ethernet Adapter with failover creation

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1 -attr
ha_mode=auto ctl_chan=ent3 netaddr=9.3.4.1 largesend=1
ent4 Available
en4
et4
```

Important: Mismatching SEA and SEA failover could cause broadcast storms to occur and affect the stability of your network. When upgrading from an SEA to an SEA failover environment, it is imperative that the Virtual I/O Server with the regular SEA be modified to SEA failover *prior* to creating the second SEA with SEA failover enablement.

Note: If you have an existing SEA you configure it to failover mode you using the **chdev** command as shown below

```
$ chdev -dev ent4 -attr ha_mode=auto ctl_chan=ent3
```

6. Verify the SEA adapter attributes on both Virtual I/O Servers' SEA adapter, as shown in Example 4-3.

Example 4-3 Verify and change attributes for SEA adapter

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

ctl_chan      ent3    Control Channel adapter for SEA failover                          True
gvrp          no      Enable GARP VLAN Registration Protocol (GVRP)                     True
ha_mode       auto    High Availability Mode                                            True
jumbo_frames  no      Enable Gigabit Ethernet Jumbo Frames                              True
large_receive no      Enable receive TCP segment aggregation                            True
largesend     1       Enable Hardware Transmit TCP Resegmentation                       True
netaddr       9.3.4.1 Address to ping                                                   True
pvid          1       PVID to use for the SEA device                                    True
pvid_adapter  ent2    Default virtual adapter to use for non-VLAN-tagged packets        True
real_adapter  ent0    Physical adapter associated with the SEA                          True
thread        1       Thread mode enabled (1) or disabled (0)                           True
virt_adapters ent2    List of virtual adapters associated with the SEA (comma separated) True
```

7. Define the network configuration for each Virtual I/O Server on the newly created SEA interface. You can use the **mktcpip** command or the **cfgassist** command to launch the SMIT menus, as shown in Figure 4-11 for VIO_Server2.

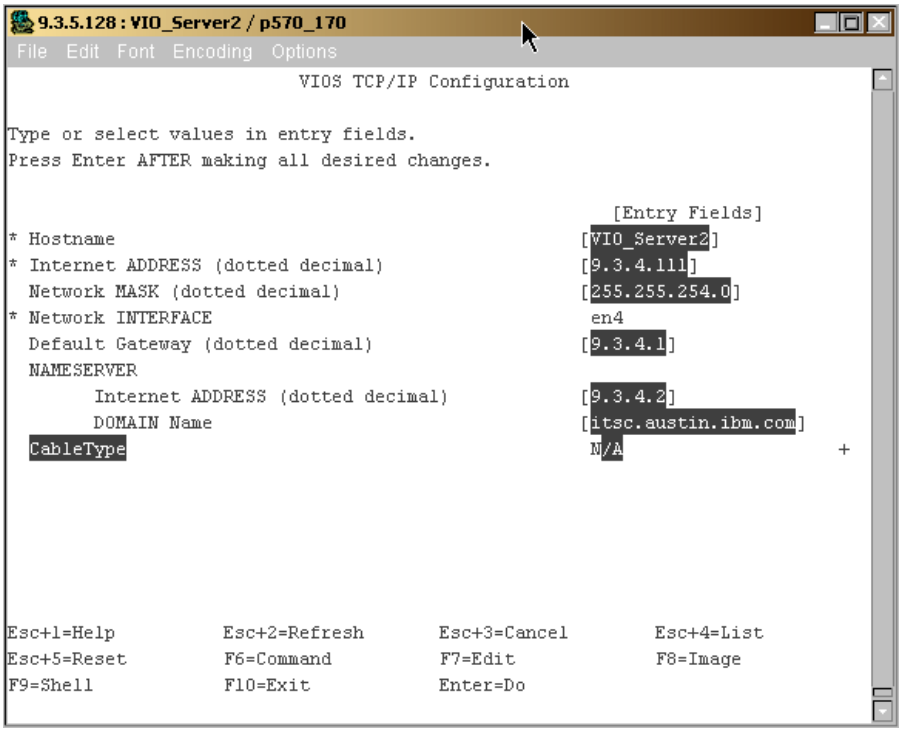


Figure 4-11 Create an IP address on the Shared Ethernet Adapter using cfgassist

8. Create the client partitions following the instructions in 3.5.5, “Creating client partitions” on page 197. Each client partition needs to be configured with one virtual Ethernet adapter in VLAN 1.

Testing Shared Ethernet Adapter failover

Perform the following steps to test whether the SEA failover configuration works as expected.

1. Open a remote shell session from a system on the external network to any of the client partitions. If your session gets disconnected during any of the tests your configuration is not highly available. You might want to run a **ping** command to verify that you have continuous connectivity.

2. On the VIO_Server1 check whether the primary adapter is active using the **entstat** command:

```
$ entstat -all ent4 | grep Active
  Priority: 1  Active: True
```

Attention: If your Virtual I/O Servers also provide virtual SCSI disks in addition to SEA, stale partitions will occur on the client partitions when a Virtual I/O Server shuts down. Thus, after each test, you must make sure that you resynchronize all stale partitions before proceeding with the next test case.

3. Perform a manual failover using the following **chdev** command to switch to the standby adapter:

```
chdev -dev ent4 -attr ha_mode=standby
```

4. Check whether the SEA failover was successful using the **entstat** command. On VIO_Server1 the entstat output should look like this:

```
$ entstat -all ent4 | grep Active
  Priority: 1  Active: False
```

You should also see the following entry in the errorlog when you issue the **errlog** command:

```
40D97644   1205135007 I H ent4           BECOME BACKUP
```

On VIO_Server2 the entstat output should look like this:

```
$ entstat -all ent4 | grep Active
  Priority: 2  Active: True
```

You should see the following entry in the errorlog when you issue the **errlog** command.

```
E136EAFA   1205135007 I H ent4           BECOME PRIMARY
```

Important: You may experience up to 30 seconds delay in failover when using SEA failover. The behavior depends on the network switch and the spanning tree settings.

5. On VIO_Server1 switch back to the primary adapter and verify that the primary adapter is active using these commands:

```
$ chdev -dev ent4 -attr ha_mode=auto
ent4 changed
$ entstat -all ent4 | grep Active
  Priority: 1  Active: True
```

6. Unplug the link of the physical adapter on VIO_Server1. Use the entstat command to check whether the SEA has failed over to the standby adapter.
7. Re-plug the link of the physical adapter on VIO_Server1 and verify that the SEA has switched back to the primary.

4.7 Scenario 2: Network Interface Backup in the client

Network Interface Backup (NIB) can be used to provide redundant access to external networks when two Virtual I/O Servers are used. Figure 4-12 on page 271 shows the sample configuration. To keep the picture simple, only the client partition DB_Server is shown.

In a NIB configuration the Shared Ethernet Adapters (SEA) on the two Virtual I/O Servers are each configured on a different VLAN. In our example VIO_Server1 is configured on VLAN 1 and VIO_Server2 is configured on VLAN 2. Each Virtual I/O Server therefore needs a virtual Ethernet Adapter configured with the correct VLAN ID.

The client partitions has two virtual Ethernet Adapters. One of them is configured on VLAN 1 providing the connection to the SEA in VIO_Server1. The second one is configured on VLAN 2 connecting to VIO_Server2. Using these two adapters an EtherChannel in Network Interface Backup mode is configured.

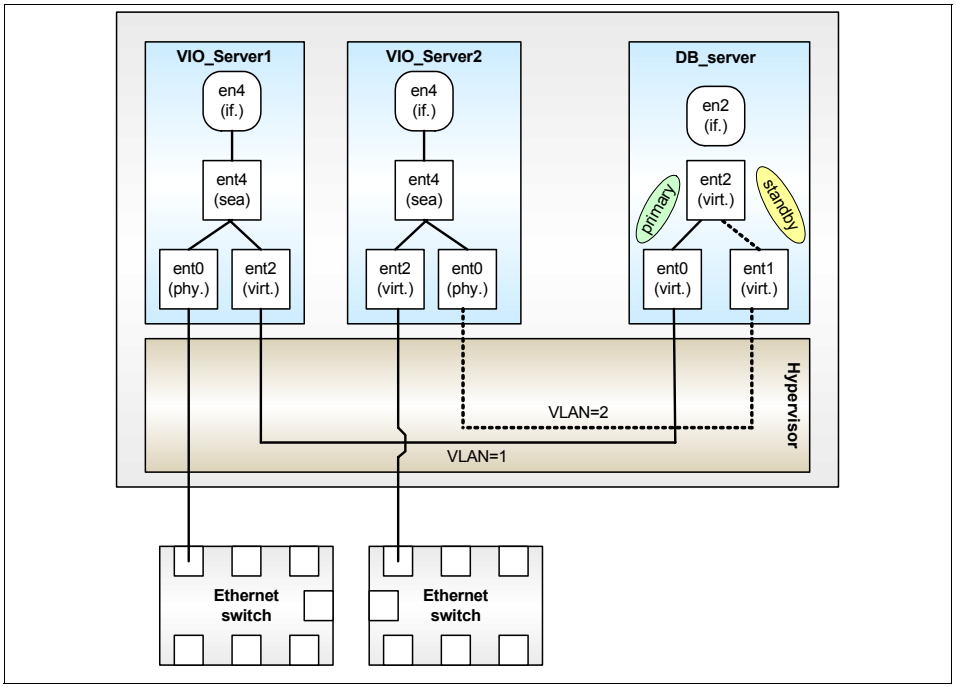


Figure 4-12 NIB configuration

4.7.1 Configuring Network Interface Backup

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 3.3, “Creating a Virtual I/O Server partition” on page 153. In step 10, select one Ethernet Adapter and one storage adapter.
2. Install both Virtual I/O Servers by following the instructions in 3.4, “Virtual I/O Server software installation” on page 173.
3. Configure each Virtual I/O Server with one virtual Ethernet adapter. Each Virtual I/O Server needs to be on a different VLAN.

Table 4-3 Network interface backup configuration examples

Virtual I/O Server	Virtual I/O Server slot	VLAN
VIO_Server1	10	1
VIO_Server2	10	2

4. On each Virtual I/O Server define a Shared Ethernet Adapter with the correct VLAN ID. Example 4-4 shows the configuration of the SEA on VIO_Server1 with VLAN ID 1 through the *defaultid* flag. Example 4-5 shows the configuration of the SEA on VIO_Server2 with VLAN ID 2.

Example 4-4 SEA adapter configuration on VIO_Server1

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1
ent4 Available
en4
et4
```

Example 4-5 SEA adapter configuration on VIO_Server2

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 2
ent4 Available
en4
et4
```

5. Also add virtual SCSI adapters to have a highly available virtual SCSI configuration as described in 4.8, “Scenario 3: MPIO in the client with SAN” on page 275 or 4.9, “Scenario 4: LVM mirroring” on page 288.
6. Create the client partitions following the instructions in 3.5.5, “Creating client partitions” on page 197. Each client partition needs to be configured with one virtual Ethernet adapter in VLAN1 and one in VLAN2.
7. In each client partition define the EtherChannel entering **smit** → **Devices** → **Communication** → **EtherChannel / IEEE 802.3ad Link Aggregation** → **Add An EtherChannel / Link Aggregation** or using the **smitty etherchannel** command. Select the primary adapter, ent0, and press Enter. Example 4-6 shows the SMIT menu for adding an EtherChannel/Link Aggregation.

Example 4-6 Add An EtherChannel / Link Aggregation smit menu

```
Add An EtherChannel / Link Aggregation

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

                                                  [Entry Fields]
  EtherChannel / Link Aggregation Adapters        ent0                  +
  Enable Alternate Address                        no                    +
  Alternate Address                               []                    +
  Enable Gigabit Ethernet Jumbo Frames            no                    +
  Mode                                            standard              +
  Hash Mode                                       default               +
  Backup Adapter                                  ent1                  +
```

```
     Automatically Recover to Main Channel          yes                    +
     Perform Lossless Failover After Ping Failure   yes                    +
  Internet Address to Ping                         [9.3.4.1]
  Number of Retries                                []                     +#
  Retry Timeout (sec)                              []                     +#


Esc+1=Help          Esc+2=Refresh       Esc+3=Cancel        Esc+4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Note: The following message, which appears in the errorlog when the EtherChannel is created, can be ignored:

```
5561971C   1123194707 P S ent2 UNSUPPORTED IOCTL IN DEVICE DRIVER
```

4.7.2 Testing Network Interface Backup

Perform the following steps to verify that the NIB configuration works as expected. The steps below show how failover works when the network connection of a Virtual I/O Server is disconnected. The failover works in the same fashion when a Virtual I/O Server is rebooted. You can test this by rebooting VIO_Server1 instead of unplugging the network cable in step 2.

1. Do a remote login to the client partition using Telnet or SSH. Once you are logged in check whether the primary channel that is connected to VIO_Server1 is active using the **entstat** command, as shown in Example 4-7.

   Example 4-7 Verifying the active channel in an EtherChannel

   ```
   # entstat -d ent2 | grep Active
   Active channel: primary channel
   ```

2. Unplug the network cable from the physical network adapter that is connected to VIO_Server1.
3. As soon as the EtherChannel notices that it has lost connection it should perform a switchover to the backup adapter. You will see a message as shown in Example 4-8 in the errorlog.

   Important: Your telnet or SSH connection should not be disconnected. If it is disconnected your configuration is not highly available.

   Example 4-8 Errorlog message when primary channel fails

   ```
   LABEL:          ECH_PING_FAIL_PRMRY
   ```

```
IDENTIFIER:     9F7B0FA6

Date/Time:       Fri Nov 23 19:53:35 CST 2007
Sequence Number: 141
Machine Id:      00C1F1704C00
Node Id:         NIM_server
Class:           H
Type:            INFO
WPAR:            Global
Resource Name:   ent2
Resource Class:  adapter
Resource Type:   ibm_ech
Location:

Description
PING TO REMOTE HOST FAILED

Probable Causes
CABLE
SWITCH
ADAPTER

Failure Causes
CABLES AND CONNECTIONS

        Recommended Actions
        CHECK CABLE AND ITS CONNECTIONS
        IF ERROR PERSISTS, REPLACE ADAPTER CARD.

Detail Data
FAILING ADAPTER
PRIMARY
SWITCHING TO ADAPTER
ent1
Unable to reach remote host through primary adapter: switching over
to backup adapter
```

As shown in Example 4-9, the **entstat** command will also show that the backup adapter is now active.

Example 4-9 Verifying the active channel in an EtherChannel

```
# entstat -d ent2 | grep Active
Active channel: backup adapter
```

4. Reconnect the physical adapter in VIO_Server1.
5. The EtherChannel will not automatically switch back to the primary channel. In the SMIT menu, there is an option to **"**`Automatically Recover to Main Channel`**"**. It is set to Yes by default and this is the behavior when using physical adapters. However, virtual adapters do not adhere to this. Instead, the backup channel is used until it fails and then switches to the primary channel. You have to manually switch back using the **`/usr/lib/methods/ethchan_config -f`** command as shown in Example 4-10. A message will appear in the errorlog when the EtherChannel recovers to the primary channel.

Example 4-10 Manual switch to primary channel using entstat

```
# /usr/lib/methods/ethchan_config -f ent2
#  entstat -d ent2 | grep Active
Active channel: primary channel
# errpt
8650BE3F 1123195807 I H ent2 ETHERCHANNEL RECOVERY
```

Note: You can use the **`dsh`** command to automate the check or switch back if you have a lot of client partitions.

4.8 Scenario 3: MPIO in the client with SAN

This scenario shows how to set up a highly available virtual SCSI configuration using MPIO in the client partition. Figure 4-13 on page 276 shows the configuration. To keep the picture simple, only two client partitions are shown.

The disks are located on an external storage subsystem on the SAN. The disks on the storage subsystem are assigned to both Virtual I/O Servers. The client partitions see the disks through two paths using MPIO. Each of the paths is going through a different Virtual I/O server.

Note: The MPIO for virtual SCSI devices only supports failover mode.

In this scenario the disk subsystem is a DS4200. Each Virtual I/O Server is attached to the SAN using one Fiber Channel adapter. On the Virtual I/O Servers the RDAC driver is used to access the SAN disks. The RDAC driver is already provided in the base configuration regardless of the existence of multiple Fibre Channel adapters.

Note: Typically you will use two Fibre Channel adapters connected to two different SAN switches in a productive setup.

When using only one Fibre Channel per Virtual I/O Server, you need an additional switch to have a highly available configuration. In this case, it is important that the SAN zoning be configured such that the single Fibre Channel in each Virtual I/O Server LPAR is zoned to see both storage controllers in the FAStT. If a second Fibre Channel is used for additional redundancy, the storage administrator must ensure that each Fibre Channel is zoned to only one of the DS4200 controllers. Rules for attachment of FAStT storage units to AIX can be found in the Storage Manager documentation of the Storage Manager products.

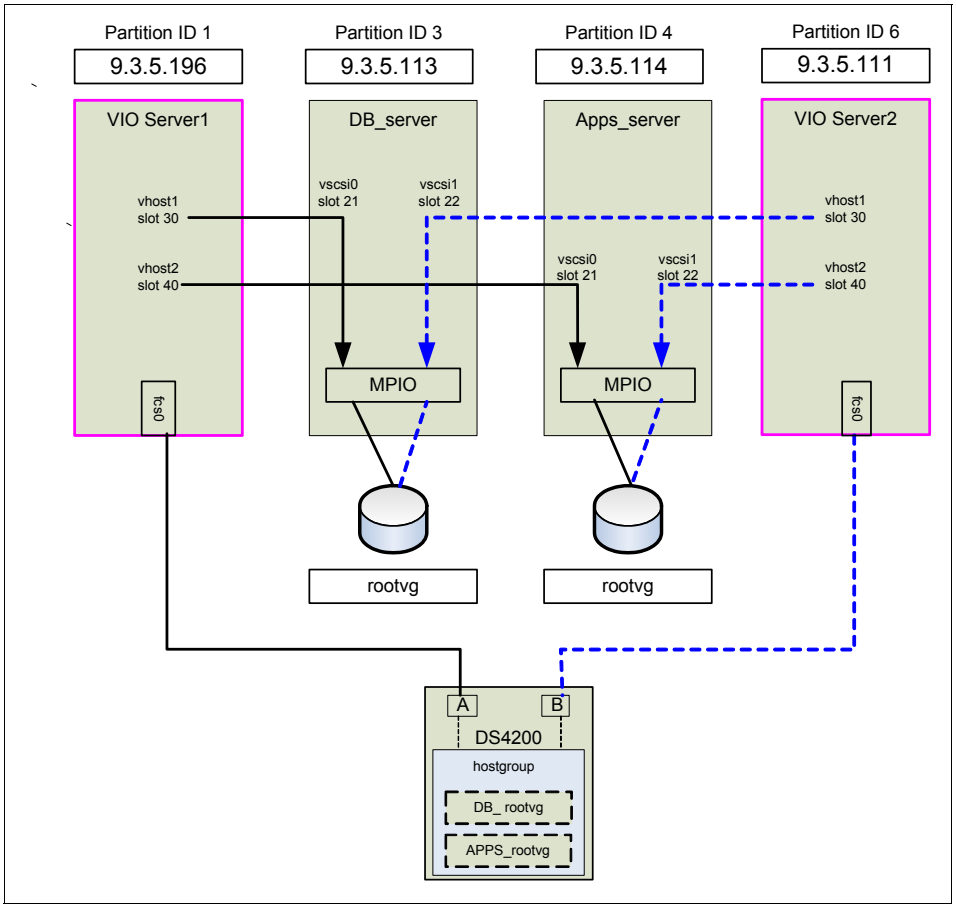


Figure 4-13 SAN attachment with MPIO with multiple Virtual I/O Server

4.8.1 Configuring MPIO

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 3.3, “Creating a Virtual I/O Server partition” on page 153. In step 10, select one Fibre Channel adapter in addition to the shown physical adapter.
2. Install both Virtual I/O Servers by following the instructions in 3.4, “Virtual I/O Server software installation” on page 173.
3. Change the fc_err_recov to fast_fail and dyntrk to yes attributes on the Fibre Channel adapters. You can use the **lsdev -type adapter** command to find the number of the Fibre Channel adapter. Use the **chdev** command, as shown in Example 4-11, to change the attributes.

Example 4-11 Changing the attributes of the Fibre Channel adapter

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
$ lsdev -dev fscsi0 -attr
attribute    value     description
user_settable

attach       switch    How this adapter is CONNECTED         False
dyntrk       yes       Dynamic Tracking of FC Devices        True
fc_err_recov fast_fail FC Fabric Event Error RECOVERY Policy True
scsi_id      0x660c00  Adapter SCSI ID                       False
sw_fc_class  3         FC Class for Fabric                   True
```

The reason for changing the fc_err_recov attribute to fast_fail is that if the Fibre Channel adapter driver detects a link event, such as a lost link between a storage device and a switch, then any new I/O or future retries of the failed I/O operations will be failed immediately by the adapter until the adapter driver detects that the device has rejoined the fabric. The default setting for this attribute is delayed_fail.

Setting the dyntrk attribute to yes makes AIX tolerate cabling changes in the SAN. Be aware that this function is not supported on all storage systems. Check with your storage vendor for support.

Important: If you have two or more Fibre Channel adapters per Virtual I/O Server you have to change the attributes for each of them.

4. Reboot the Virtual I/O Servers for the changes to the Fibre Channel devices to take effect.
5. Create the client partitions following the instructions in 3.5.5, “Creating client partitions” on page 197. Table 4-4 shows the required virtual SCSI adapters based on the configuration shown in Figure 4-13 on page 276.

Table 4-4 Virtual SCSI adapter configuration for MPIO

Virtual I/O Serve	Virtual I/O Server slot	Client partition	Client partition slot
VIO_Server1	30	DB_server	21
VIO_Server1	40	Apps_Server	21
VIO_Server2	30	DB_server	22
VIO_Server2	40	Apps_Server	22

6. Also add one or two virtual Ethernet adapters to each client to provide the highly available network access: One adapter if you plan on using SEA failover for network redundancy, as described in 4.6, “Scenario 1: Shared Ethernet Adapter failover” on page 264, or two adapters if you plan on using Network Interface Backup for network redundancy, as described in 4.7, “Scenario 2: Network Interface Backup in the client” on page 270.
7. On VIO_Server1 and VIO_Server2 use the **fget_config** command to get the LUN to hdisk mappings, as shown in Example 4-12. In this example hdisk2 and hdisk3 are used for the client partitions. It is important that you identify the same disks on both Virtual I/O Servers.

Note: Depending on the MPIO software you use the commands to identify the SAN disks will be different. If you have SDDPCM installed use the **pcmpath query devices** command. If you have SDD installed use the **datapath query devices** command.

Example 4-12 Listing the LUN to hdisk mappings

```
# fget_config -vA

---dar0---

User array name = 'FAST200'
dac0 ACTIVE dac1 ACTIVE

Disk     DAC   LUN Logical Drive
utm              1
```

```
hdisk0   dac1    0 1
hdisk1   dac0    2 2
hdisk2   dac0    3 4
hdisk3   dac1    4 3
hdisk4   dac1    5 5
hdisk5   dac0    6 6
```

You can also use the **lsdev -dev hdisk*n* -vpd** command, where *n* is the hdisk number, to retrieve this information, as shown in Example 4-13.

Example 4-13 Listing the LUN to hdisk mapping using the lsdev command

```
$ lsdev -dev hdisk2 -vpd
  hdisk2
U789D.001.DQDYKYW-P1-C2-T1-W200500A0B8110D0F-L3000000000000  3542
(200) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:  block
$ lsdev -dev hdisk3 -vpd
  hdisk3
U789D.001.DQDYKYW-P1-C2-T1-W200500A0B8110D0F-L4000000000000  3542
(200) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:
block
```

Note: If possible keep the hdisk numbering on the two Virtual I/O Servers identical. Having identical numbering makes the management of the mappings easier.

8. The disks are to be accessed through both Virtual I/O Servers. The reserve_policy for each disk must be set to no_reserve on VIO_Server1 and VIO_Server2. Change the reserve_policy attribute to no_reserve using the **chdev** command, as shown in Example 4-14 on page 280.

Example 4-14 Set the attribute to no_reserve

```
$ chdev -dev hdisk2 -attr reserve_policy=no_reserve
hdisk2 changed
$ chdev -dev hdisk3 -attr reserve_policy=no_reserve
hdisk3 changed
```

9. Check, using the **lsdev** command, to make sure reserve_policy attribute is now set to no_reserve, as shown in Example 4-15.

Example 4-15 Output of the hdisk attribute with changed reserve_policy attribute

```
$ lsdev -dev hdisk2 -attr
attribute      value                           description
user_settable

PR_key_value   none                            Persistant Reserve Key Value           True
cache_method   fast_write                      Write Caching method                   False
ieee_volname   600A0B8000110D0E0000000E47436859 IEEE Unique volume name               False
lun_id         0x0003000000000000              Logical Unit Number                    False
max_transfer   0x100000                        Maximum TRANSFER Size                  True
prefetch_mult  1                               Multiple of blocks to prefetch on read False
pvid           none                            Physical volume identifier             False
q_type         simple                          Queuing Type                           False
queue_depth    10                              Queue Depth                            True
raid_level     5                               RAID Level                             False
reassign_to    120                             Reassign Timeout value                 True
reserve_policy no_reserve                      Reserve Policy                         True
rw_timeout     30                              Read/Write Timeout value               True
scsi_id        0x660a00                        SCSI ID                                False
size           20480                           Size in Mbytes                         False
write_cache    yes                             Write Caching enabled                  False
```

Important: The no_reserve policy has to be set for each disk which will use MPIO.

10. Double-check both Virtual I/O Servers to make sure that the vhost adapters have the correct slot numbers by running the **lsmap -all** command.

11. Map the hdisks to the vhost adapters using the **mkvdev** command, as shown in Example 4-16.

Example 4-16 Mapping the disks to the vhost adapters

```
$ mkvdev -vdev hdisk2 -vadapter vhost0 -dev app_server
app_server Available
$ mkvdev -vdev hdisk3 -vadapter vhost1 -dev db_server
db_server Available
```

Important: When multiple Virtual I/O Servers attach to the same disk, only hdisk is supported as a backing device. You cannot create a volume group on these disks and use a logical volume as a backing device.

Check the mappings, as shown in Example 4-17.

Example 4-17 lsmap -all output after mapping the disks on VIO_Server1

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.MMA.101F170-V1-C30                    0x00000000

VTD                   app_server
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk2
Physloc
U789D.001.DQDYKYW-P1-C2-T1-W200500A0B8110D0F-L3000000000000

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost1          U9117.MMA.101F170-V1-C40                    0x00000000

VTD                   db_server
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk3
Physloc
U789D.001.DQDYKYW-P1-C2-T1-W200500A0B8110D0F-L4000000000000
```

Notice the LUN numbers from the listing (Example 4-18).

Example 4-18 lsmap -all output after mapping the disks on VIO_Server2

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.MMA.101F170-V6-C30                    0x00000000

VTD                   app_server
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk2
Physloc               U789D.001.DQDYKYW-P1-C3-T2-W200500A0B8110D0F-L3000000000000

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost1          U9117.MMA.101F170-V6-C40                    0x00000000

VTD                   db_server
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk3
Physloc               U789D.001.DQDYKYW-P1-C3-T2-W200500A0B8110D0F-L4000000000000
```

12. Install the client partitions. For instructions on installing the AIX client partitions, refer to 3.5.6, “AIX client partition installation” on page 207.

4.8.2 Configuring MPIO in the client partitions

The following steps configure MPIO in the client partitions. In our example the DB_server partition uses VIO_Server1 as the primary path for virtual SCSI traffic. The APP_server partition will use VIO_Server2 as the primary path.

Note: MPIO for virtual SCSI devices currently only support failover mode in AIX.

Alternatively, if you use SEA failover for network redundancy, you could direct all virtual SCSI traffic to the backup Virtual I/O Server for the network. See 4.5, “Separating and balancing disk and network traffic” on page 262 for details on separating traffic.

The following steps show how to configure the client partitions:

1. Check the MPIO configuration by running the commands shown in Example 4-19. Only one configured hdisk shows up in this scenario.

Example 4-19 Verifying the disk configuration on the client partitions

```
# lspv
hdisk0          00c1f1707355c8e5                    rootvg
active
# lsdev -Cc disk
hdisk0 Available  Virtual SCSI Disk Drive
```

2. Run the **lspath** command to verify that the disk is attached using two different paths. Example 4-20 shows that hdisk0 is attached using the VSCSI0 and VSCSI1 adapter that point to different Virtual I/O Servers. Both Virtual I/O Servers are up and running. Both paths are enabled.

Example 4-20 Verifying the paths of hdisk0

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

3. Enable the health check mode for the disk so that the status of the disks is automatically updated. Health check mode is disabled by default (hckeck_interval=0). As long as it is disabled the client partition does not update the path status in case of a failure of the active path. To activate the health check function, use the **chdev** command, as shown in Example 4-21. In this example, we use a health check interval of 20 seconds. To check for the attribute setting, use the **lsattr** command.

Note: The path switching also works if the hcheck_interval attribute is disabled, but it is recommended to have the status updated automatically.

Example 4-21 Changing the health check interval

```
# chdev -l hdisk0 -a hcheck_interval=20 -P
hdisk0 changed
# lsattr -El hdisk0
PCM             PCM/friend/vscsi                 Path Control Module        False
algorithm       fail_over                        Algorithm                  True
hcheck_cmd      test_unit_rdy                    Health Check Command       True
hcheck_interval 20                               Health Check Interval      True
hcheck_mode     nonactive                        Health Check Mode          True
max_transfer    0x40000                          Maximum TRANSFER Size      True
pvid            00c1f1707355c8e50000000000000000 Physical volume identifier False
queue_depth     3                                Queue DEPTH                True
reserve_policy  no_reserve                       Reserve Policy             True
```

Note: MPIO on the client partition runs a fail_over algorithm. That means only one path is active at a time. If you shut down a Virtual I/O Server that serves the inactive path, then the path mode does not change to failed because no I/O is using this path.

Failover may require a small period of time in which the client re-establishes a path to the SAN. Testing of production workloads should be made in order to verify that this delay is acceptable.

4. Enable the virtual SCSI client adapter path timeout feature for each virtual SCSI client adapter. By default the feature is disabled (vscsi_path_to=0). To activate it use the **chdev** command as shown in the following example. In this example the timeout is set to the minimum value of 30 seconds.

Example 4-22 Configuring the virtual SCSI client adapter path timeout feature

```
# chdev -l vscsi0 -a vscsi_path_to=30 -P
vscsi0 changed
# lsattr -El vscsi0
vscsi_path_to 30 Virtual SCSI Path Timeout True
```

Note: If you try to set a value smaller than 30 seconds for the vscsi_path_to attribute it will be automatically changed to 30 seconds.

5. In our example we want the DB_server virtual SCSI traffic to go through VIO_Server1 and the Apps_server traffic to through VIO_Server2. Determine which path is connected to which Virtual I/O Servers using the **lscfg** command and verify the slot numbers for the VSCSI devices, as shown in Example 4-23.

Example 4-23 Find out which parent belongs to which path

```
# lscfg -vl vscsi0
  vscsi0           U9117.MMA.101F170-V3-C21-T1  Virtual SCSI Client Adapter

        Hardware Location Code......U9117.MMA.101F170-V3-C21-T1

# lscfg -vl vscsi1
  vscsi1           U9117.MMA.101F170-V3-C22-T1  Virtual SCSI Client Adapter

        Hardware Location Code......U9117.MMA.101F170-V3-C22-T1
```

In our example slot number 21 points to VIO_Server1 and slot number 22 to VIO_Server2.

6. By default all the paths are defined with priority 1 meaning that traffic will go through the first path (Priority 1 is the highest priority, and you can define a priority from 1 to 255). For the DB_server partition no updates to the path priority are necessary because the first path is using vscsi0 which is connected to VIO_Server1. For the Apps_server partition the path priority has to be updated so that the primary path is going through VIO_Server2.

   To set the primary path to use the VSCSI1 device, the priority of the VSCI0 path is changed to 2 so that it becomes a lower priority, as shown in

Example 4-24. The priority for the VSCSI1 devices remains at 0 so that it is the primary path.

Example 4-24 Changing the priority of a path

```
# chpath -l hdisk0 -p vscsi0 -a priority=2
path Changed
# lspath -AHE -l hdisk0 -p vscsi0
attribute value description user_settable

priority  2     Priority    True

# lspath -AHE -l hdisk0 -p vscsi1
attribute value description user_settable

priority  1     Priority    True
```

Note: You may have to update the Preferred Path in the SAN to reflect the changed priority settings.

7. Reboot the client partition for the changes to take effect. Both changes require a reboot to be activated since this disk belongs to the rootvg and is in use.

4.8.3 Testing MPIO

Perform the following steps to verify that the MPIO configuration works as expected. The steps below show how failover works when one Virtual I/O Server is shutdown. This can happen if you are performing maintenance on the Virtual I/O Server. This example shows how path failover works in the DB_server partition.

1. Log in to the client partition and verify that all the paths are enabled using the **lspath** command as shown in Example 4-25.

Example 4-25 Displaying MPIO path status in the client partition

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

2. Shut down VIO_Server2.
3. As soon as the failover to the alternate path has been done you will see a message in the errorlog, as shown in Example 4-26 on page 287. The **lspath**

command will show the path as failed, as shown in Example 4-27 on page 288.

Example 4-26 Errorlog message when path fails

```
LABEL:          SC_DISK_ERR7
IDENTIFIER:     DE3B8540

Date/Time:       Sat Nov 24 15:21:39 CST 2007
Sequence Number: 10
Machine Id:      00C1F1704C00
Node Id:         db_server1
Class:           H
Type:            PERM
WPAR:            Global
Resource Name:   hdisk0
Resource Class:  disk
Resource Type:   vdisk
Location:        U9117.MMA.101F170-V3-C21-T1-L810000000000

Description
PATH HAS FAILED

Probable Causes
ADAPTER HARDWARE OR CABLE
DASD DEVICE

Failure Causes
UNDETERMINED

        Recommended Actions
        PERFORM PROBLEM DETERMINATION PROCEDURES
        CHECK PATH

Detail Data
PATH ID
           1
SENSE DATA
0A00 2800 0017 0E38 0000 0804 0000 0000 0000 0000 0000 0000 0200
0B00 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
```

```
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000
```

Example 4-27 lspath output with failed path

```
# lspath
Enabled hdisk0 vscsi0
Failed  hdisk0 vscsi1
```

4. Start the Virtual I/O Server.
5. When the Virtual I/O Server comes up the status of the failed path will go back to enabled if the hcheck_interval has been set. Use the **lspath** command as show in figure Example 4-28 to verify that all paths are enabled again.

Example 4-28 lspath output with enabled paths

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

Note: The path will only automatically go back to enabled status if the hcheck_interval has been defined. Else you have to manually set it back to enabled status using the chpath command as shown in the following example:

```
chpath -s enabled -l hdisk -p vscsi0
```

4.9 Scenario 4: LVM mirroring

This scenario shows how to set up a highly available virtual SCSI configuration using LVM mirroring in the client partition. Figure 4-14 on page 289 shows the configuration. The client partitions are configured with two virtual SCSI adapters. Each of these virtual SCSI adapters is connected to a different Virtual I/O server and provides one disk to the client partition. On the Virtual I/O Servers the virtual disks are backed by a logical volume. The client partition uses LMV mirroring so that the rootvg disk access is redundant.

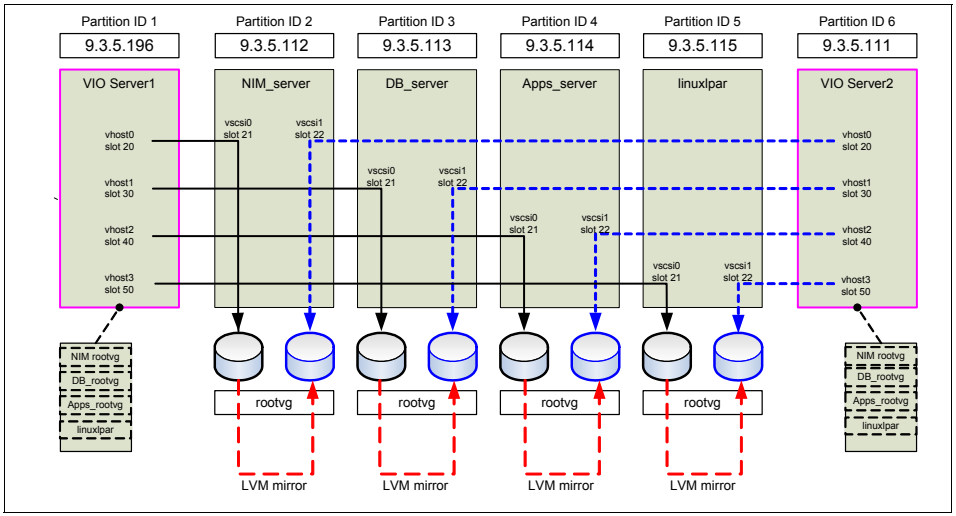


Figure 4-14 LVM mirroring scenario

4.9.1 Configuring LVM mirroring

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 3.3, “Creating a Virtual I/O Server partition” on page 153. In step 10, select one Ethernet Adapter and one storage adapter. Depending on the hardware configuration you have this might be a Fibre Channel adapter or a SCSI adapter. Figure 4-15 on page 290 shows the adapters that were selected in the example configuration for VIO_Server2. Slot C5 on bus 514 contains an Ethernet adapter and slot c3 on bus 518 contains a Fibre Channel adapter.

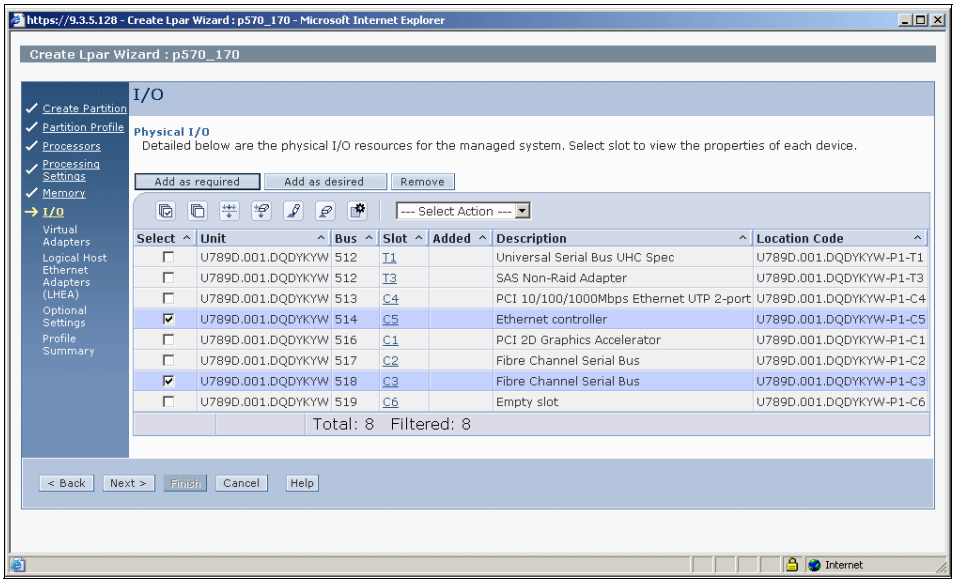


Figure 4-15 VIO_Server2 physical adapter selection

2. Install both Virtual I/O Servers by following the instructions in 3.4, “Virtual I/O Server software installation” on page 173.
3. Configure the virtual SCSI adapters on VIO_Server1 and VIO_Server2 as shown in table Table 4-5.

Table 4-5 Virtual SCSI adapter configuration for LVM mirroring

Virtual I/O Server	Virtual I/O Server slot	Client partition	Client partition slot
VIO_Server1	20	NIM_server	21
VIO_Server1	30	DB_server	21
VIO_Server1	40	Apps_Server	21
VIO_Server1	50	linuxlpar	21
VIO_Server2	20	NIM_server	22
VIO_Server2	30	DB_server	22
VIO_Server2	40	Apps_Server	22
VIO_Server2	50	linuxlpar	22

4. Create the client partitions as shown in 4.9.1, “Configuring LVM mirroring” on page 289 following the instructions in 3.5.5, “Creating client partitions” on page 197. Each client partition needs to be configured with two virtual SCSI adapters as shown in Table 4-5 on page 290. Figure 4-16 shows the virtual SCSI adapter configuration for VIO_Server2 on the HMC.

Note: In Figure 4-16 there is a filter applied to the *Type* column so that only the virtual SCSI adapters are shown.

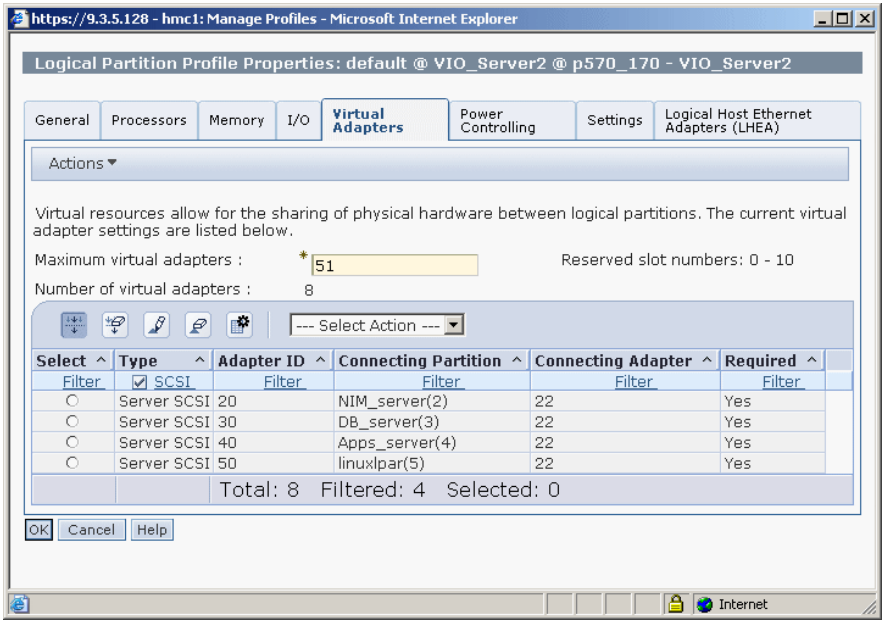


Figure 4-16 Virtual SCSI adapters for VIO_Server2

5. Also add one or two virtual Ethernet adapters to each client to provide the highly available network access. One adapter if you plan on using SEA failover for network redundancy, as described in 4.6, “Scenario 1: Shared Ethernet Adapter failover” on page 264, or two adapters if you plan on using Network Interface Backup for network redundancy, as described in 4.7, “Scenario 2: Network Interface Backup in the client” on page 270.
6. Create the volume group and logical volumes on VIO_Server1 and VIO_Server2. Refer to 3.5.2, “Defining virtual disks” on page 182 for more information.

7. Check the devices list on VIO_Server1 and VIO_Server2 using the **lsdev** command, as shown in Example 4-29. There should be four vhost devices available.

Example 4-29 VIO_Server2 partition

```
$ lsdev -virtual
name           status
description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vasi0           Available  Virtual Asynchronous Services Interface (VASI)
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vhost2          Available  Virtual SCSI Server Adapter
vhost3          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
vappssrv_rvg    Available  Virtual Target Device - Logical Volume
vdbsrv_rvg      Available  Virtual Target Device - Logical Volume
vlinux          Available  Virtual Target Device - Logical Volume
vnimsrv_rvg     Available  Virtual Target Device - Logical Volume
```

8. A logical volume from the rootvg_clients volume group should be mapped to each of the four vhost devices as shown in Example 4-30.

Example 4-30 Virtual SCSI mappings

```
$ mkvdev -vdev nimsrv_rvg -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev dbsrv_rvg -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev appssrv_rvg -vadapter vhost2 -dev vappssrv_rvg
vappssrv_rvg Available
$ mkvdev -vdev linux -vadapter vhost3 -dev vlinux
vlinux Available
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.MMA.101F170-V6-C20                    0x00000000

VTD                   vnimsrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        nimsrv_rvg
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost1          U9117.MMA.101F170-V6-C30                    0x00000000
```

```
VTD                   vdbsrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        dbsrv_rvg
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost2          U9117.MMA.101F170-V6-C40                    0x00000000

VTD                   vappssrv_rvg
Status                Available
LUN                   0x8100000000000000
Backing device        appssrv_rvg
Physloc

SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost3          U9117.MMA.101F170-V6-C50                    0x00000000

VTD                   vlinux
Status                Available
LUN                   0x8100000000000000
Backing device        linux
Physloc
```

9. When you bring up the client partitions, you should have hdisk0 and hdisk1 available, as shown in Example 4-31. Mirror the rootvg as you would normally do on AIX.

Example 4-31 NIM_server partition with hdisk1 served off of VIO_Server2

```
# hostname
NIM_server
# lspv
hdisk0          00c1f1706e8787f9                    rootvg          active
hdisk1          none                                None
# lsdev -Cc disk
hdisk0 Available  Virtual SCSI Disk Drive
hdisk1 Available  Virtual SCSI Disk Drive
# lsdev -p vscsi0
hdisk0 Available  Virtual SCSI Disk Drive
# lsdev -p vscsi1
hdisk1 Available  Virtual SCSI Disk Drive
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
# mirrorvg -m rootvg hdisk1
0516-1804 chvg: The quorum change takes effect immediately.
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.
# bosboot -a -d /dev/hdisk1

bosboot: Boot image is 26927 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -o
hdisk0 blv=hd5
hdisk1 blv=hd5
```

4.9.2 Testing LVM mirroring

Perform the following steps to test if the LVM mirroring configuration works as expected.

1. Make sure that all logical volumes in the rootvg volume group are correctly mirrored. One mirror copy should be on the virtual disk provided by VIO_Server1 while the other should be on the virtual disk provided by VIO_Server2.
2. Verify that all the logical volumes are synchronized and that you have no stale partitions, as shown in Example 4-32

Example 4-32 Verifying synchronization status using the lsvg command

```
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
hd5                 boot       3       6       2    closed/syncd  N/A
hd6                 paging     64      128     2    open/syncd    N/A
hd8                 jfs2log    1       2       2    open/syncd    N/A
hd4                 jfs2       3       6       2    open/syncd    /
hd2                 jfs2       82      164     2    open/syncd    /usr
hd9var              jfs2       2       4       2    open/syncd    /var
hd3                 jfs2       8       16      2    open/syncd    /tmp
hd1                 jfs2       2       4       2    open/syncd    /home
hd10opt             jfs2       10      20      2    open/syncd    /opt
hd11admin           jfs2       16      32      2    open/syncd    /admin
```

Important: If your Virtual I/O Servers also provide Shared Ethernet Adapters, then Shared Ethernet Adapter of Network Interface Backup failovers will occur on the VIO clients when a Virtual I/O Server shuts down or fails.

3. Shut down the VIO_Server2 partition using the **shutdown** command.
4. Once VIO_Server2 is down, the hdisks which are served by this Virtual I/O Server should go to missing state in the client partitions. You should also start

to see stale partitions as shown in Example 4-33 in the errorlog. The client partitions should continue to be available and not experience a crash.

Example 4-33 Errors when virtual disk becomes unavailable

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
EAA3D429   1123163907 U S LVDD           PHYSICAL PARTITION MARKED STALE
F7DDA124   1123163907 U H LVDD           PHYSICAL VOLUME DECLARED MISSING
52715FA5   1123163907 U H LVDD           FAILED TO WRITE VOLUME GROUP STATUS
AREA
E86653C3   1123163907 P H LVDD           I/O ERROR DETECTED BY LVM
EAA3D429   1123163907 U S LVDD           PHYSICAL PARTITION MARKED STALE
E86653C3   1123163907 P H LVDD           I/O ERROR DETECTED BY LVM
857033C6   1123163907 T S vscsi1         Underlying transport error
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                   VOLUME GROUP:     rootvg
PV IDENTIFIER:      00c1f1706e9e4ca9 VG IDENTIFIER
00c1f17000004c00000001166e8788c9
PV STATE:           missing
STALE PARTITIONS:   16                       ALLOCATABLE:      yes
PP SIZE:            8 megabyte(s)            LOGICAL VOLUMES:  10
TOTAL PPs:          639 (5112 megabytes)     VG DESCRIPTORS:   1
FREE PPs:           448 (3584 megabytes)     HOT SPARE:        no
USED PPs:           191 (1528 megabytes)     MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  125..64..03..128..128
USED DISTRIBUTION:  03..64..124..00..00
#
```

5. Reactivate the VIO_Server2 partition from the HMC. Once it is up issue a **varyonvg rootvg** command in the client partition. This will put the missing hdisk back into active state and synchronize the stale partitions as shown in Example 4-34 on page 296. Depending on the size of the volume group and the number of stale partitions, it might take a few minutes until all logical volumes are synchronized. You can also issue the **varyonvg -n rootvg** command and then start the synchronization manually with the **syncvg -v rootvg** command.

Important: If you have several volume groups in a client partition that are served by Virtual I/O Server make sure you issue a varyonvg for each of them and verify each one if there are no stale partitions.

Note: Depending on your dump device configuration you might receive the following error message when issuing the varyonvg rootvg command:

```
0516-1774 varyonvg: Cannot varyon volume group with an active
dump device on a missing physical volume.
Use sysdumpdev to temporarily replace the dump device with
/dev/sysdumpnull and try again.
```

In this case set the dump device temporarily to /dev/sysdumpnull and restore the original dump device configuration after the varyonvg.

Example 4-34 Adding the missing disk back

```
# varyonvg rootvg
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                  VOLUME GROUP:     rootvg
PV IDENTIFIER:      00c1f1706e9e4ca9 VG IDENTIFIER
00c1f17000004c00000001166e8788c9
PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            8 megabyte(s)           LOGICAL VOLUMES:  10
TOTAL PPs:          639 (5112 megabytes)    VG DESCRIPTORS:   1
FREE PPs:           448 (3584 megabytes)    HOT SPARE:        no
USED PPs:           191 (1528 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  125..64..03..128..128
USED DISTRIBUTION:  03..64..124..00..00
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
hd5                 boot       3       6       2    closed/syncd  N/A
hd6                 paging     64      128     2    open/syncd    N/A
hd8                 jfs2log    1       2       2    open/syncd    N/A
hd4                 jfs2       3       6       2    open/syncd    /
hd2                 jfs2       82      164     2    open/syncd    /usr
hd9var              jfs2       2       4       2    open/syncd    /var
hd3                 jfs2       8       16      2    open/syncd    /tmp
hd1                 jfs2       2       4       2    open/syncd    /home
hd10opt             jfs2       10      20      2    open/syncd    /opt
hd11admin           jfs2       16      32      2    open/syncd
/admin
```

4.10 Linux on POWER network installation

Although Linux on POWER can be installed from CD or DVD in a virtualized environment, you normally use network installation. A Linux installation can be

fully automated and run unattended using the distribution-specific tools such as autoyast or kickstart.

After the initial installation additional service and productivity tools provided by IBM should be installed. These tools are required for the dynamic LPAR functionality. You can find the tools and the installation instructions at:

http://www.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

Linux network installation on POWER requires the following components:

- BOOTP or DHCP server to answer BOOTP requests
- TFTP server to provide the boot file
- NFS, FTP or HTTP server to provide the installation files repository
- Optional kickstart or autoyast file for unattended automated installation

These components are usually located on the same server. As installation server you can use an AIX server (for example a NIM server) or a Linux server used for other Linux installations.

4.10.1 Using an AIX server for Linux installation

If you plan to use an AIX server for installing Linux it must have BOOTP, TFTP and NFS enabled. This is normally already the case if the AIX server is used as a NIM server. If not, you have to configure these services in /etc/inetd.conf and refresh the inetd daemon.

The following steps are required to enable a Linux network installation from an AIX server:

1. Add the following line to the /etc/bootptab file. Adapt the IP addresses and subnet mask according to your environment:

   ```
   linuxlpar0:bf=/tftpboot/ppc64.img:ip=9.3.5.200:ht=ethernet:sa=9.3.5.
   197:sm=255.255.254.0:
   ```

   If you are going to initiate a broadcast BOOTP request add the MAC address to the line using the ha=XXXXXXXXXXXX statement. The ha statement must be placed after ht=ethernet statement.

2. Copy the content of the installation DVDs to the installation server, then export the directory by adding it to the /etc/exports file as shown here:

   ```
   /export/linux/rhel5
   /export/linux/sles10
   ```

Tip: If you are getting “permission denied” errors when trying to NFS mount, add the IP address of the Linux partition you are trying to install to the /etc/hosts file on the installation server.

3. Copy the network boot kernel to /etc/tftpboot. For SLES10 the file is called inst64. For Red Hat 5 the file is called ppc64.img.
4. Set up the remote IPL settings in SMS or use the Open Firmware prompt to perform a network boot. The Open Firmware prompt is only needed to provide boot arguments for the Red Hat installation for example to enable multipath or to have VNC during the installation. The following example shows how to perform a network boot from the Open Firmware prompt:

```
0 > devalias net /vdevice/l-lan@30000002  ok
0 > boot net:9.3.5.197,,9.3.5.115, install mpath vnc
vncpassword=abc123
```

More details on the installation can be found at:

http://www.ibm.com/collaboration/wiki/display/LinuxP/Installation

The Linux on POWER installation is also described in Chapter 2 of *Deploying Linux on IBM eserver pSeries Clusters*, SG24-7014, which can be found at:

http://www.redbooks.ibm.com/abstracts/sg247014.html

4.10.2 Using a Linux installation server for installation

If you have a Linux installation server with DHCP enabled it can be used for the installation of Linux on POWER partitions.

Add the following lines to /etc/dhcpd.conf:

```
ignore unknown-clients;
not authoritative;
allow bootp;
allow booting;
```

Note: The DHCP server only answers to BOOTP broadcast requests. Therefore, the IPL settings defined in SMS must not contain any IP address settings. If you set the IP address in order to test connectivity to the installation server using the ping functionality in SMS, make sure that you reset the values back to zero before initiating the installation.

4.11 Using Linux on POWER with two Virtual I/O Servers

When using two Virtual I/O Servers a Linux on POWER client partition can be configured for continuous availability. Similar to the scenarios described in 4.1, “Providing continuous availability for virtual I/O” on page 244, the following techniques can be used the achieve continuous availability for Linux on POWER partition:

- Device mapper multipath functionality for MPIO
- Software RAID mirroring or mdadm
- Ethernet connection bonding
- Shared Ethernet Adapter failover

Shared Ethernet Adapter Failover works in the same fashion for a Linux client partition as it does for AIX. No special configuration is required in the Linux client partition. Therefore it is not covered in further detail here. See 4.3.1, “Shared Ethernet Adapter failover” on page 252 for more details.

In most installation scenarios Shared Ethernet Adapter Failover is the recommended solution for providing continuous access to external networks because it is easy to set up and maintain. Ethernet connection bonding offers more flexibility for separating network traffic between the Virtual I/O Servers but requires more configuration efforts. See 4.3.3, “When to use SEA failover or Network Interface Backup” on page 256 for a more detailed comparison of the two solutions.

4.11.1 Ethernet connection bonding

Ethernet connection bonding in a Linux on POWER client partition can be used to achieve network redundancy when using two Virtual I/O Servers.

In such a configuration each Virtual I/O Server provides an active Ethernet connection which is combined in a bonding device. For more details on the bonding configuration, see:

http://www-941.ibm.com/collaboration/wiki/display/LinuxP/Bonding+configuration

Because there is no hardware link failure for virtual Ethernet devices to trigger a failover to the other adapter network interface, backup mode with arpping is used. The kernel should support the arp_validation feature. This is the case starting with kernel version 2.6.19.

The bonding options are set during the module load. They are added to the `/etc/modprobe.conf` as shown in Example 4-35 on page 300.

Example 4-35 /etc/modprobe.conf example

```
alias bond0 bonding
options bond0 mode=active-backup arp_interval=1000
arp_ip_target=9.156.175.1,9.156.175.8 primary=eth0 arp_validate=all
alias eth0 ibmveth
alias eth1 ibmveth
alias scsi_hostadapter ibmvscsic
```

Once the module is loaded, the interfaces have to be configured. Example 4-36 shows an example configuration in the /etc/sysconfig/network-scripts/ifcfg-bond0 file.

Example 4-36 /etc/sysconfig/network-scripts/ifcfg-bond0

```
DEVICE=bond0
BOOTPROTO=static
BROADCAST=9.3.5.255
IPADDR=9.3.5.115
NETMASK=255.255.255.0
NETWORK=9.3.5.0
ONBOOT=yes
GATEWAY=9.3.5.1
```

Example 4-37 and Example 4-38 show the configuration files for the slave devices. Make sure that the slave devices have no IP addresses defined and that there are no definitions left from previous configurations. The **ifconfig -a** command only shows the IP address assigned to all the devices.

Example 4-37 etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

Example 4-38 etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

```
USERCTL=no
```

Note: In most installation scenarios SEA failover is easier to use and to maintain and therefore the recommended solution. Also see 4.3.3, “When to use SEA failover or Network Interface Backup” on page 256.

Testing Ethernet connection bonding

To verify that the device mapper multipath configuration works as expected, perform the following steps:

1. Verify that both links are up as shown below. The status of both slave interfaces should be up and the link failure count should be 0. In the example below the active slave adapter is eth0.

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v2.6.3-rh (June 8, 2005)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:02

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:03
```

2. Unplug the connection of the SEA in one of the Virtual I/O Servers to the external network. In our example we are unplugging the link in VIO_Server1 to which the slave interface eth0 is connected. As you can see in the example below the slave interface is going to status down. The currently active slave interface has changed to eth1.

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v2.6.3-rh (June 8, 2005)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
```

```
Currently Active Slave: eth1
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: down
Link Failure Count: 1
Permanent HW addr: ba:d3:f0:00:40:02

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:03
```

3. You can also see the status of the Ethernet bonding connection using the **dmesg** command as shown here:

```
bonding: bond0: link status down for active interface eth0,
disabling it
bonding: bond0: making interface eth1 the new active one.
```

Note: Once the primary slave interface becomes active again there is no automatic switch back.

4.11.2 Software RAID mirroring

In a software RAID mirroring configuration the client partitions are configured with two virtual SCSI adapters. Each of these virtual SCSI adapters is connected to a different Virtual I/O server and provides one disk to the client partition. On the Virtual I/O Servers the virtual disks are backed by a logical volume or physical disk. The client partition uses software RAID mirroring so that disk access is redundant. In Linux mdadm provides the software RAID functionality. Figure 4-17 shows the configuration.

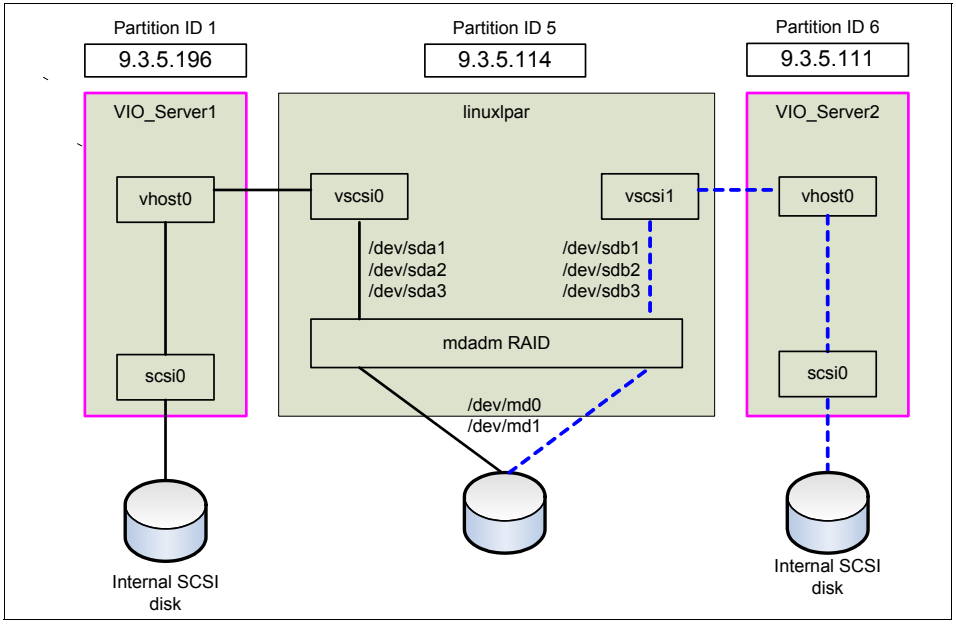


Figure 4-17 Linux client partition using mirroring with mdadm

The first virtual disk (on the virtual adapter with lowest slot number on the client) will become /dev/sda in the Linux client partition. The second virtual disk will become /dev/sdb. If you are mirroring the root disks you have to partition the disks first. Then create the software RAID devices containing the disk partitions.

Note: In order to be bootable the disk has to contain a PrepBoot partition at the beginning (equivalent to the AIX boot device /dev/hd5). This device cannot be on a software RAID and it has to be created first.

Figure 4-18 shows the disk partitioning configuration. The setup differs depending on the Linux distribution and the version you are using. Detailed information can be found in the IBM Linux on POWER Wiki at:

http://www-941.ibm.com/collaboration/wiki/display/LinuxP/Raid

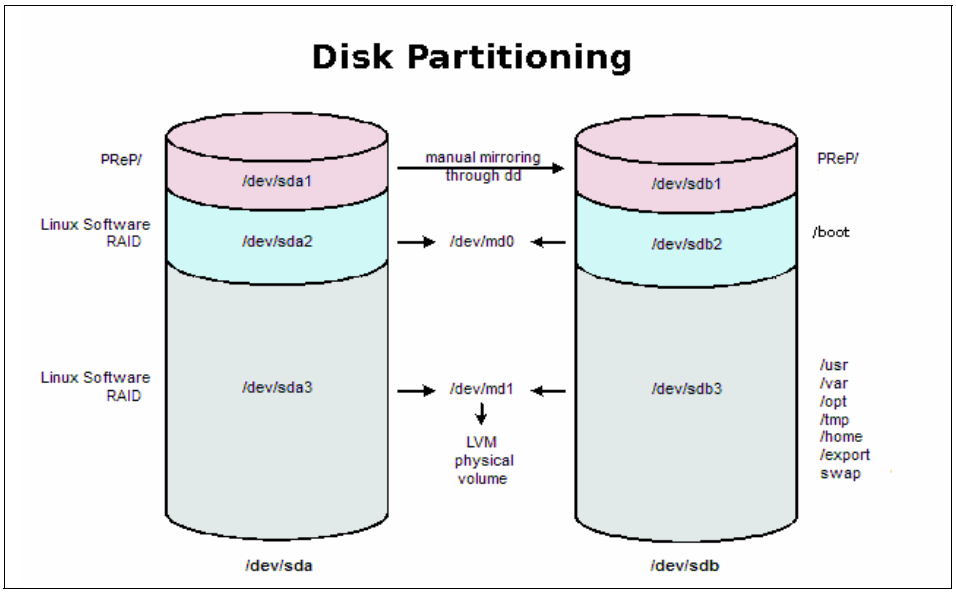


Figure 4-18 Linux partitioning layout for mdadm mirroring (Red Hat)

Recovering from disk failure

If a virtualized disk is not accessible, you will see the disk marked as failed as shown in Example 4-39.

Example 4-39 cat /proc/mdstat showing lost disk

```
[root@op720-1-client4 ~]# cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 sdb3[1] sda3[0]
      1953728 blocks [2/2] [UU]

md2 : active raid1 sdb4[1] sda4[2](F)
      21794752 blocks [2/1] [_U]

md0 : active raid1 sdb2[1] sda2[2](F)
      98240 blocks [2/1] [_U]
unused devices: <none>
```

To recover the disk, perform the following steps:

1. Set the disk to faulty status

```
# mdadm --manage --set-faulty /dev/md0 /dev/sda2
# mdadm --manage --set-faulty /dev/md1 /dev/sda3
# mdadm --manage --set-faulty /dev/md2 /dev/sda4
```

2. Remove the device

```
# mdadm --manage --remove /dev/md0 /dev/sda2
# mdadm --manage --remove /dev/md1 /dev/sda3
# mdadm --manage --remove /dev/md2 /dev/sda4
```

3. Rescan the device (choose the corresponding path)

```
echo 1 > /sys/class/scsi_device/0\:0\:1\:0/device/rescan
```

4. Hot add the device to mdadm

```
# mdadm --manage --add /dev/md0 /dev/sda2
# mdadm --manage --add /dev/md1 /dev/sda3
# mdadm --manage --add /dev/md2 /dev/sda4
```

5. Verify that resynchronization is running

```
# cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 sda3[0] sdb3[1]
      1953728 blocks [2/2] [UU]

md2 : active raid1 sda4[2] sdb4[1]
      21794752 blocks [2/1] [_U]
      [=>...................]  recovery =  5.8% (1285600/21794752)
finish=8.2min speed=41470K/sec
md0 : active raid1 sda2[0] sdb2[1]
      98240 blocks [2/2] [UU]
```

Important: Never reboot the other Virtual I/O Server as long as the recovery is not finished.

4.11.3 Device mapper multipath

The example in Figure 4-19 shows a configuration where the disks are located on an external storage subsystem on the SAN. The disks are assigned to both Virtual I/O Servers. The client partitions see the disks through two paths using device mapper multipath. Each of the paths is going through a different Virtual I/O server.

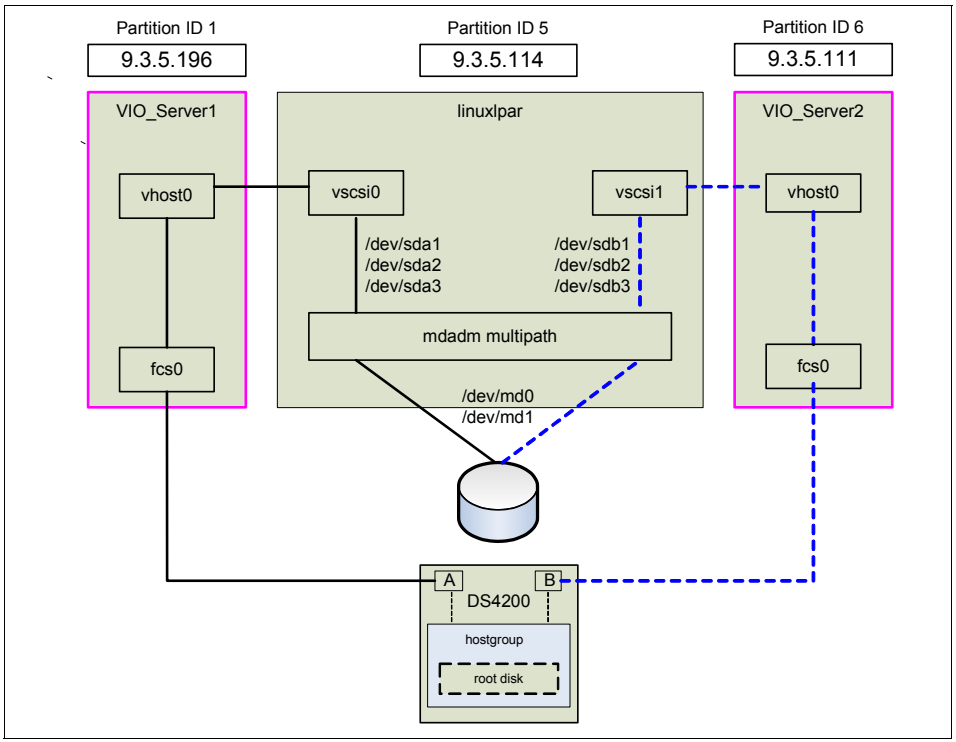


Figure 4-19 Linux client partition using MPIO to access SAN storage

To set up a Red Hat 5 Linux client partition with multipathing enabled, the **mpath** argument has to be specified when performing the initial Linux installation (see 4.10, “Linux on POWER network installation” on page 296).

The main advantage of the MPIO configuration compared to RAID mirroring is that after the failure the lost paths will be automatically recovered once they are available again. There is no need to log in to a client partition and take corrective actions.

The device mapper multipath devices can be listed using the **multipath -ll** command.

Testing device mapper multipath

To verify that your device mapper multipath configuration works as expected, perform the following steps:

1. Verify that both paths are active using the **multipath** command:

```
# multipath -ll
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDASD
[size=20G][features=0][hwhandler=0]
\_ round-robin 0 [prio=1][active]
 \_ 0:0:1:0 sda 8:0   [active][ready]
\_ round-robin 0 [prio=1][enabled]
 \_ 1:0:1:0 sdb 8:16  [active][ready]
```

> **Note:** For path management you can also use the interactive shell, which can be started using the **multipathd** command.

2. Shut down one of the Virtual I/O Servers. In this example VIO_Server1 is shut down. In var/log/messages you will see the following messages:

```
ibmvscsi: Partner adapter not ready
ibmvscsi: error after reset
Dec  7 15:46:57 localhost kernel: ibmvscsi: Virtual adapter failed rc 2!
Dec  7 15:46:57 localhost kernel: rpa_vscsi: SPR_VERSION: 16.a
Dec  7 15:46:57 localhost kernel: ibmvscsi: Partner adapter not ready
Dec  7 15:46:57 localhost kernel: ibmvscsi: error after reset
device-mapper: multipath: Failing path 8:0.
Dec  7 15:47:01 localhost kernel: device-mapper: multipath: Failing path 8:0.
Dec  7 15:47:01 localhost multipathd: 8:0: reinstated
Dec  7 15:47:01 localhost multipathd: mpath0: remaining active paths: 2
Dec  7 15:47:01 localhost multipathd: sda: readsector0 checker reports path is down
Dec  7 15:47:01 localhost multipathd: checker failed path 8:0 in map mpath0
Dec  7 15:47:01 localhost multipathd: mpath0: remaining active paths: 1
Dec  7 15:47:06 localhost multipathd: sda: readsector0 checker reports path is down
```

3. When using the **multipath** command to display the path status, one path should be marked as failed as shown here:

```
# multipath -ll
sda: checker msg is "readsector0 checker reports path is down"
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDASD
[size=20G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][enabled]
 \_ 0:0:1:0 sda 8:0   [failed][faulty]
\_ round-robin 0 [prio=1][active]
 \_ 1:0:1:0 sdb 8:16  [active][ready]
[root@localhost ~]#
```

4. Restart the Virtual I/O Server. Once the Virtual I/O Server is rebooted you will see the following messages in /var/adm/messages:

```
ibmvscsic: sent SRP login
ibmvscsi: host srp version: 16.a, host partition VIO_Server1 (1), OS 3, max io
1048576
Dec  7 15:50:36 localhost last message repeated 9 times
Dec  7 15:50:39 localhost kernel: ibmvscsi: partner initialized
Dec  7 15:50:39 localhost kernel: ibmvscsic: sent SRP login
Dec  7 15:50:39 localhost kernel: ibmvscsi: SRP_LOGIN succeeded
Dec  7 15:50:39 localhost kernel: ibmvscsi: host srp version: 16.a, host partition
VIO_Server1 (1), OS 3, max io 1048576
Dec  7 15:50:41 localhost multipathd: sda: readsector0 checker reports path is down
Dec  7 15:50:46 localhost multipathd: sda: readsector0 checker reports path is up
Dec  7 15:50:46 localhost multipathd: 8:0: reinstated
Dec  7 15:50:46 localhost multipathd: mpath0: remaining active paths: 2
```

5. When using the **multipath** command to display the path status, both paths should be active again:

```
# multipath -ll
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDASD
[size=20G][features=0][hwhandler=0]
\_ round-robin 0 [prio=1][enabled]
 \_ 0:0:1:0 sda 8:0   [active][ready]
\_ round-robin 0 [prio=1][active]
 \_ 1:0:1:0 sdb 8:16  [active][ready]
```

4.12 Configuring Multiple Shared-Processor Pools

Before reading this section you should be familiar with the concepts and terms for Multiple Shared-Processor Pools as described in 2.3.3, “POWER6 Multiple Shared-Processor Pools” on page 44.

Note: The terms used in 2.3.3, “POWER6 Multiple Shared-Processor Pools” on page 44 are not the same as you will see on the HMC. The Maximum Pool Capacity (MPC) is called maximum processing units in the HMC dialogs. Reserved Pool Capacity (RPC) is referred to as reserved processing units. In the text of the following section the terms as they are defined in 2.3.3, “POWER6 Multiple Shared-Processor Pools” on page 44 are used. On the screenshots you will see the HMC terminology.

Based on the partition configuration we used in the previous sections (see 4.9, “Scenario 4: LVM mirroring” on page 288) we will create Multiple Shared-Processor Pools to limit the amount of processing capacity that can be consumed by the micro-partitions. Table 4-6 on page 309 provides an overview

of the micro-partition configuration. All the micro-partitions are running in uncapped mode.

Table 4-6 Micro-partition configuration and Shared-Processor Pool assignments

Partition name	Entitlement	Number of virtual processors	Pool name	Pool ID
VIO_Server1	0.3	2	DefaultPool	0
VIO_Server2	0.3	2	DefaultPool	0
DB_server	1	3	Production	1
Apps_server	1	3	Production	1
NIM_server	0.2	1	Development	2
linuxlpar	0.2	1	Development	2

The two Virtual I/O Servers are created in the Shared-Processor Pool with ID 0. On the HMC this pool is referred to as DefaultPool. The DefaultPool cannot be modified. Each of the Virtual I/O Servers has an entitlement of 0.3 processing units and 2 virtual processors configured. Each one is therefore guaranteed to get 0.3 processing units and can consume up to 2 processing units if required and if spare processing capacity is available in the DefaultPool.

The DB_server and the Apps_server micro-partitions are assigned to the Shared-Processor Pool with ID 1 called Production. On the HMC the pool IDs in the range between 1 and 63 represent Shared-Processor Pools which can be modified. This is in contrast to the DefaultPool, which cannot be modified.

As shown in Table 4-7 on page 310 the Maximum Pool Capacity of the Production pool is limited to 3. A reason for limiting a production workload might be to reduce licensing cost for applications running in these partitions.

The Entitled Pool Capacity, which is the amount of processing units that the pool is guaranteed to get, is 2.5 for the Production pool. This is the sum of the entitlements of the partitions in the pool (in this case 1.0 for the DB_server and 1.0 for the Apps_server) plus the Reserved Pool Capacity of 0.5 processing units. The Reserved Pool Capacity is distributed within the bounds of the pool according to the weighting factors defined for each partition. A reason for using Reserved Pool Capacity might be to guarantee unallocated processor resource to balance workload demand in production workloads.

Note: On the HMC you can only see the Maximum Pool Capacity and the Reserved Pool Capacity. The Entitled Pool Capacity cannot be displayed on the HMC. To calculate the Entitled Pool Capacity for a Shared-Processor Pool, you have to add the entitlements of all micro-partitions assigned to a Shared-Processor Pool plus the Reserved Pool Capacity.

Table 4-7 Shared-Processor Pool attributes

Pool name	Maximum pool capacity	Entitled pool capacity	Reserved pool capacity
Production	3	2.5	0.5
Development	1	0.4	0

Note: The pool name can be up to 14 characters long and can contain blanks.

Although the DB_server and the Apps_server micro-partitions each have 3 virtual processors configured, the total processing capacity which can be used by the two micro-partitions at a time is limited to 3 physical processors. This is due to the Maximum Pool Capacity in the Shared-Processor Pool. If they were assigned to the DefaultPool they could use up to 6 physical processors if capacity is available.

The NIM_server and the linuxlpar partitions are assigned to the Development pool. The Development pool has a maximum pool capacity of 1 processing units. The micro-partitions in the Development pool can use at most the capacity of 1 physical processor. The Entitled Pool Capacity of the Development pool is 0.4 as shown in Table 4-7.

Note: The pool attributes and partition assignments can be changed dynamically while the micro-partitions are running.

4.12.1 Shared-Processor Pool management using the HMC GUI

To configure the pools as described above, perform the following steps:

1. Select the server for which you want to configure the pools. Then select **Configuration** → **Shared Processor Pool Management** as shown in Figure 4-20 to open the Shared-Processor Pool configuration window.

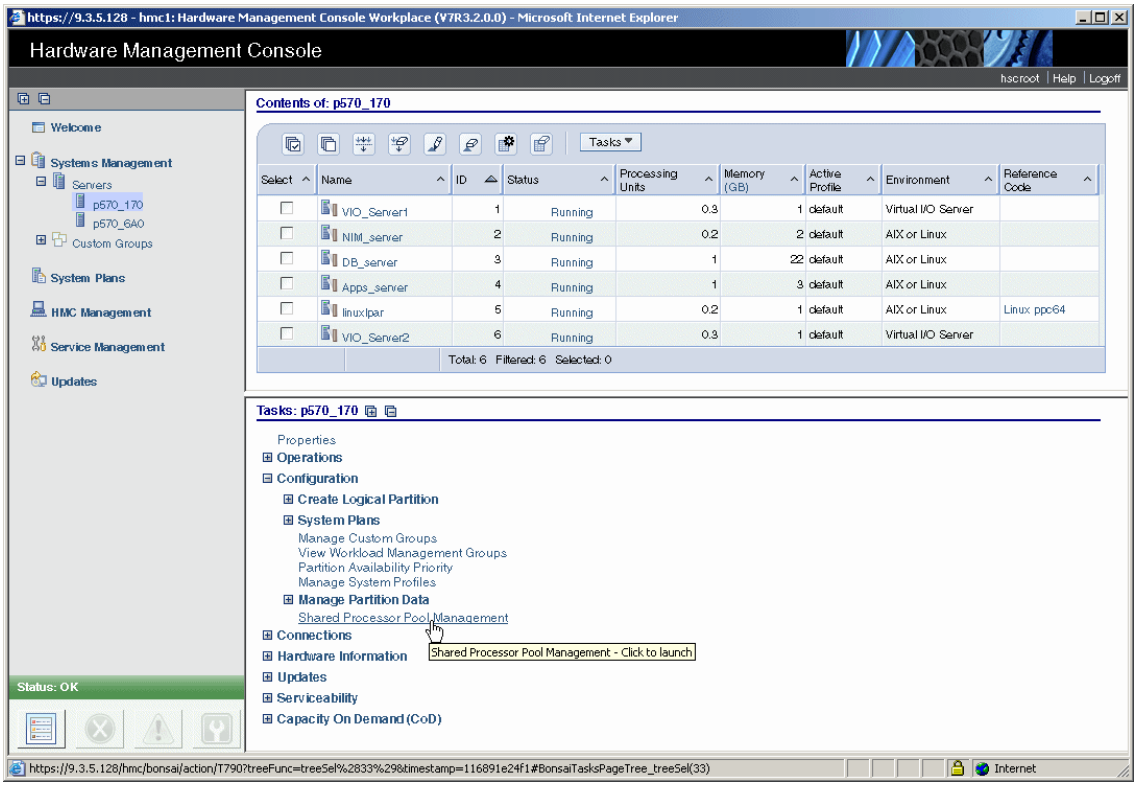


Figure 4-20 Starting Shared-Processor Pool configuration

2. Click **SharedPool01** to open the window for setting the pool attributes, as shown in Figure 4-21.

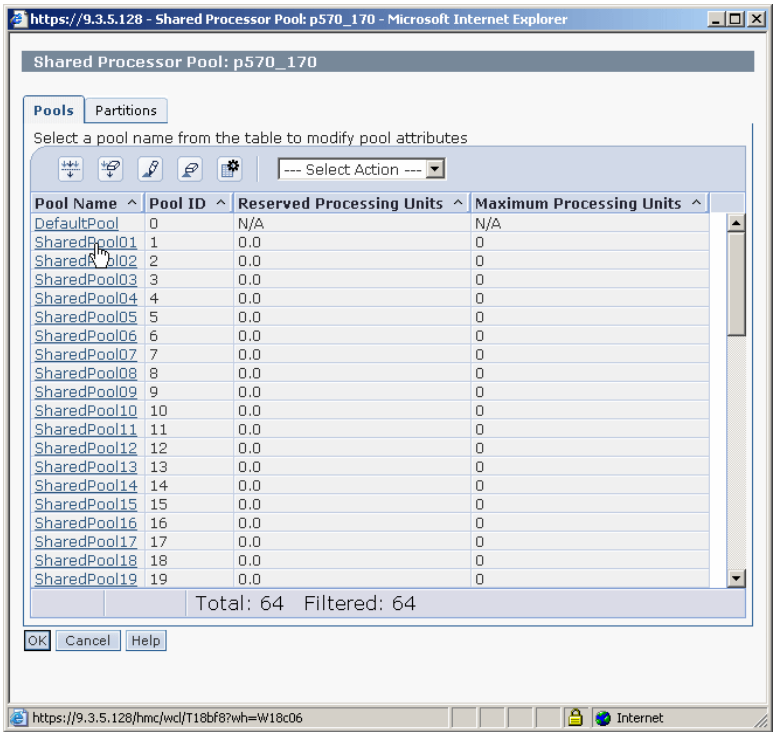


Figure 4-21 Virtual Shared-Processor Pool selection

3. Change the Pool name to Production, Maximum processing units to 3 and Reserved processing units to 0.5, as shown in Figure 4-22. Perform the same for SharedPool02 so that it becomes the Development pool with Maximum processing units of 1.

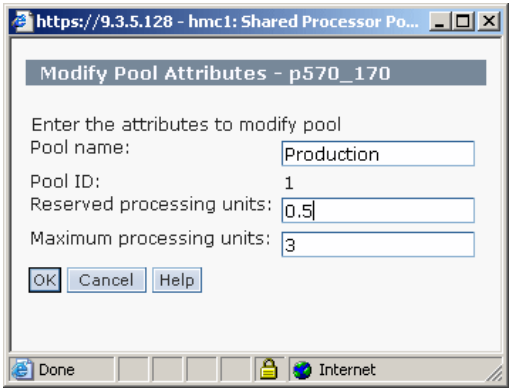


Figure 4-22 Shared-Processor Pool configuration

Notes:

- The Reserved processing units can be specified in fractions of 1/100 while the Maximum processing units can only be specified in whole numbers.
- Because the Reserved processing units are guaranteed to the Shared-Processor Pool, the number of Reserved processing units you specify has to be available on the system.

4. Change to the **Partitions** tab as shown in Figure 4-23 to perform partition assignments. Click each partition you want to reassign.

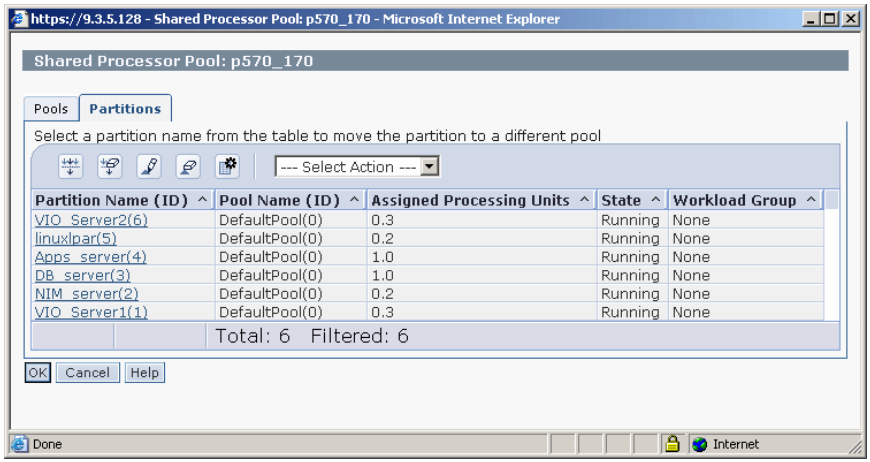


Figure 4-23 Virtual Shared-Processor Pool partition tab

5. Select the desired pool for each partition. Figure 4-24 shows the assignment of the DB_server partition to the production pool. Assign the other partitions according to Table 4-6 on page 309.

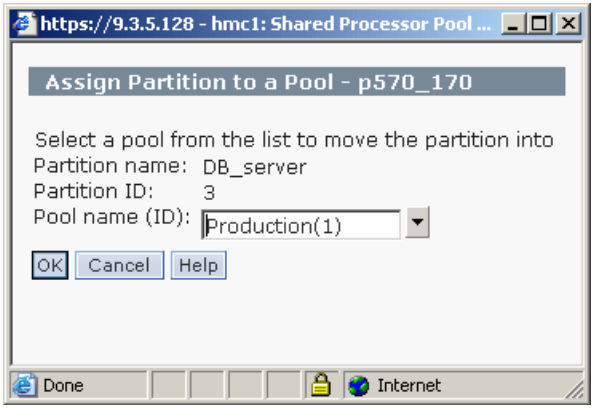


Figure 4-24 Shared-Processor Pool partition assignment

Note: You cannot assign a micro-partition to a Shared-Processor Pool if it causes the Entitled Pool Capacity to exceed the Maximum Pool Capacity.

Important: Partitions which have not been activated since they were created do not show up in the window with available partitions like the one shown in Figure 4-23. To change the assignment of a partition which has never been activated to another Shared-Processor Pool, you have to update the Shared-Processor Pool attribute which can be found in the Processors tab of the partition profile.

Figure 4-25 shows the configuration after all partitions have been assigned

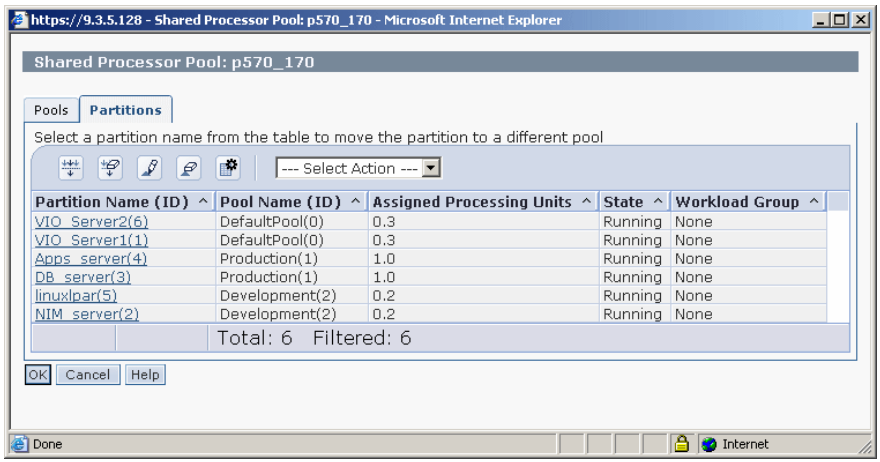


Figure 4-25 Overview of Shared-Processor Pool assignments

Important: If you move a partition to another Shared-Processor Pool, the partition profile will not be updated. Unless you change the profile the partition will be assigned to the original pool the next time it is activated. This is similar to when you are performing dynamic LPAR updates to a partition. When you save the current configuration of a running partition into a new profile, the currently assigned Shared-Processor Pool will be saved.

4.12.2 Shared-Processor Pool management using the command line

You can also manage Shared-Processor Pools through the HMC command line using the **chhwres** command. Example 4-40 shows how to configure the two Shared-Processor Pools as shown in Table 4-7 on page 310.

Example 4-40 Defining Shared-Processor Pools using the command line

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 1 -a
"new_name=Production,max_pool_proc_units=3,reserved_pool_proc_units=0.5"
```

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 2 -a
"new_name=Development,max_pool_proc_units=1,reserved_pool_proc_units=0"
```

As shown in Example 4-41, the **lshwres** command can be used to display the Shared-Processor Pool configuration.

Example 4-41 Displaying Shared-Processor Pools using chhwres

```
hscroot@hmc1:~> lshwres -r procpool -m p570_170
name=DefaultPool,shared_proc_pool_id=0,"lpar_names=VIO_Server1,VIO_Server2,DB_server,Apps_serve
r,NIM_server,linuxlpar","lpar_ids=1,6,3,4,2,5"
name=Production,shared_proc_pool_id=1,max_pool_proc_units=3.0,curr_reserved_pool_proc_units=0.0
,pend_reserved_pool_proc_units=0.0,lpar_ids=none
name=Development,shared_proc_pool_id=2,max_pool_proc_units=1.0,curr_reserved_pool_proc_units=0.
0,pend_reserved_pool_proc_units=0.0,lpar_ids=none
```

The assignment of a partition to a Shared-Processor Pool is also done using the chhwres command. Example 4-42 shows the assignment of the example partitions according to Table 4-6 on page 309.

Example 4-42 Assigning partitions to Shared-Processor Pools using chhwres

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p DB_server -a
"shared_proc_pool_name=Production"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p Apps_server -a
"shared_proc_pool_name=Production"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p NIM_server -a
"shared_proc_pool_name=Development"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p linuxlpar -a
"shared_proc_pool_name=Development"
```

Note: If you try to assign a partition that has never been activated you will get the following error:

```
This operation is not allowed because the partition is using
dedicated processors
```

To change the assignment of a partition that has never been activated to another Shared-Processor Pool, you have to update the shared_proc_pool_name or shared_proc_pool_id attributes in the partition profile using the **chsyscfg** command.

Partitions can be dynamically moved between the Shared-Processor Pools. Example 4-43 on page 317 shows how to move the linuxlpar partition from the Development pool to the Production pool.

Example 4-43 Moving a micro-partition to another Shared-Processor Pool

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p linuxlpar -a
"shared_proc_pool_name=Production"
```

Notes:

- When you move a partition to another Shared-Processor Pool, the Entitled Pool Capacities change. In our case the Entitled Pool capacity of the Development pool is reduced by 0.2 while the Entitled Pool Capacity of the Production Pool is increased by 0.2.
- If you move a partition to another Shared-Processor Pool the partition profile will not be updated. Unless you change the profile the partition will be assigned to the original pool the next time it is activated. This is similar to when you are performing dynamic LPAR updates to a partition. When you save the current configuration of running partition in to a new profile the currently assigned Shared-Processor Pool will be saved.

To deactivate a Shared-Processor Pool all the micro-partitions have to be removed from it and the Maximum Pool Capacity as well as the Reserved Pool Capacity have to be set to 0. Example 4-44 shows the commands required to deactivate the Development pool. The NIM_server micro-partition is moved to the DefaultPool and then the Development pool is set to 0 and renamed to the default name it had initially.

Example 4-44 Deactivating a Shared-Processor Pool using chhwres

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p NIM_server -a
"shared_proc_pool_name=DefaultPool"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 2 -a
"new_name=SharedPool02,max_pool_proc_units=0,reserved_pool_proc_units=0"
```

4.13 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, and cables. 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 in a PowerVM environment.

4.13.1 POWER Hypervisor switch implementation

The POWER Hypervisor switch is consistent with IEEE 802.1 Q. It works on OSI-Layer 2 and supports up to 4096 networks (4096 VLAN IDs).

When a message arrives at a Logical LAN switch port from a Logical LAN adapter, the POWER Hypervisor caches the message's source MAC address to use as a filter for future messages to the adapter. The POWER Hypervisor then processes the message depending on whether the port is configured for IEEE VLAN headers. If the port is configured for VLAN headers, the VLAN header is checked against the port's allowable VLAN list. If the message specified VLAN is not in the port's configuration, the message is dropped. After the message passes the VLAN header check, it passes onto destination MAC address processing.

If the port is *not* configured for VLAN headers, the POWER Hypervisor inserts a two-byte VLAN header (based on the port's configured VLAN number) into the message. Next, the destination MAC address is processed by searching the table of cached MAC addresses.

If a match for the MAC address is not found and if no *trunk adapter* is defined for the specified VLAN number, the message is dropped; otherwise, if a match for the MAC address is not found and if a trunk adapter is defined for the specified VLAN number, the message is passed on to the trunk adapter. If a MAC address match is found, then the associated switch port's configured, allowable VLAN number table is scanned for a match to the VLAN number contained in the message's VLAN header. If a match is not found, the message is dropped.

Next, the VLAN header configuration of the destination switch port is checked. If the port is configured for VLAN headers, the message is delivered to the destination Logical LAN adapters, including any inserted VLAN header. If the port is configured for no VLAN headers, the VLAN header is removed before being delivered to the destination Logical LAN adapter.

Figure 4-26 on page 319 shows a graphical representation of the behavior of the virtual Ethernet when processing packets.

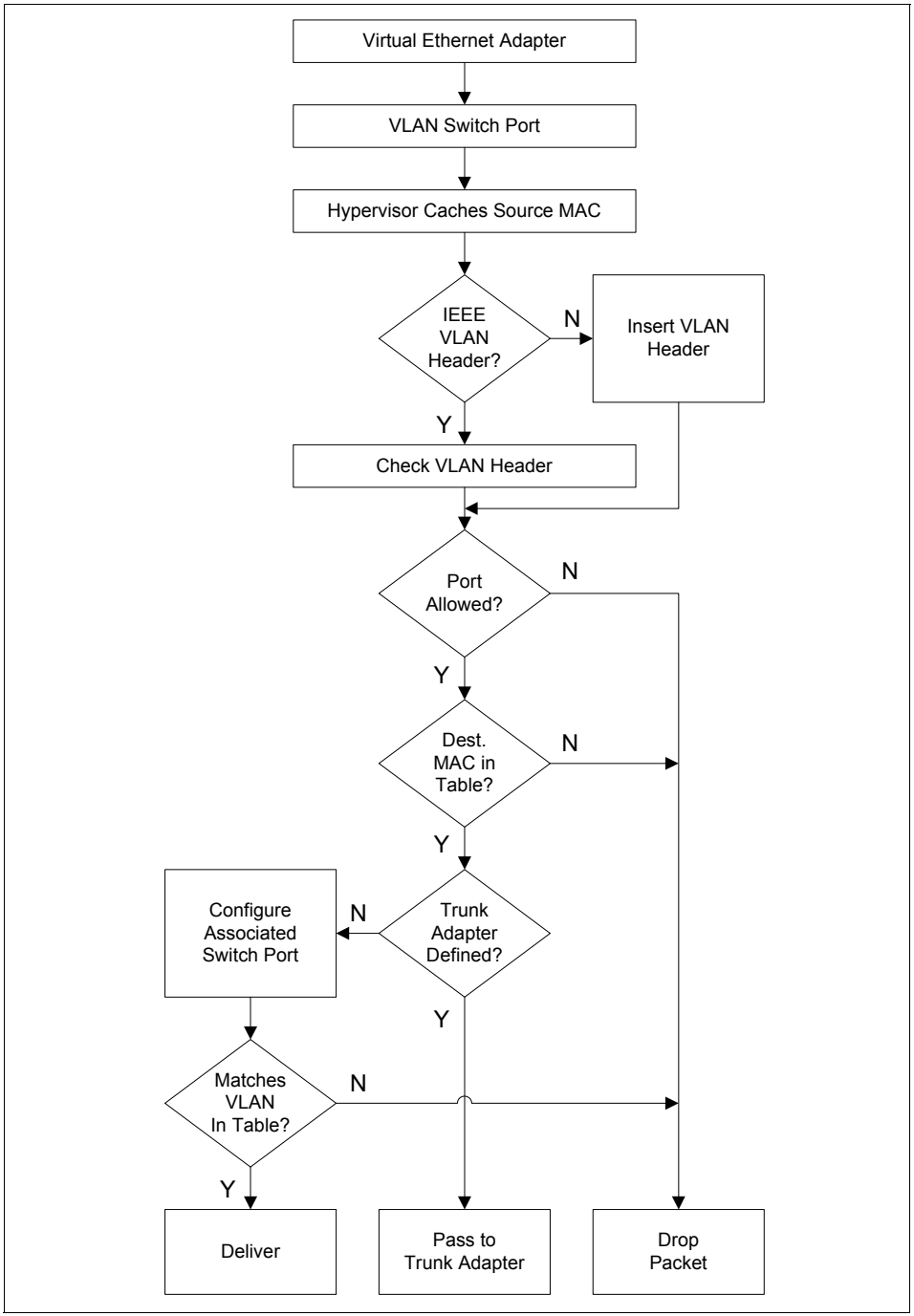


Figure 4-26 Flow chart of virtual Ethernet

4.13.2 Scenario for setting up a VLAN

In an PowerVM 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 need 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 4-27 on page 321, 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.

This implementation requires that the enterprise security policy encompasses the following:

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

  The IEEE 802.1Q VLAN tagging is implemented in the PowerVM hypervisor firmware. The Virtual I/O Server is able to have up to 21 VLANs per Shared Ethernet Adapter, but in order to use these, 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.
- The enterprise security policy allows a network switch to have multiple VLANs.

  The enterprise security policy allows multiple VLANs to share a network switch (non-physical security). If it is a security requirement that a network switch only have one VLAN, every VLAN will require a separate Shared Ethernet Adapter or Virtual I/O Server. 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.

Figure 4-27 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.

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 Virtual I/O Server 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.

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 4.13.5, “Ensuring VLAN tags are not stripped on the Virtual I/O Server” on page 325.

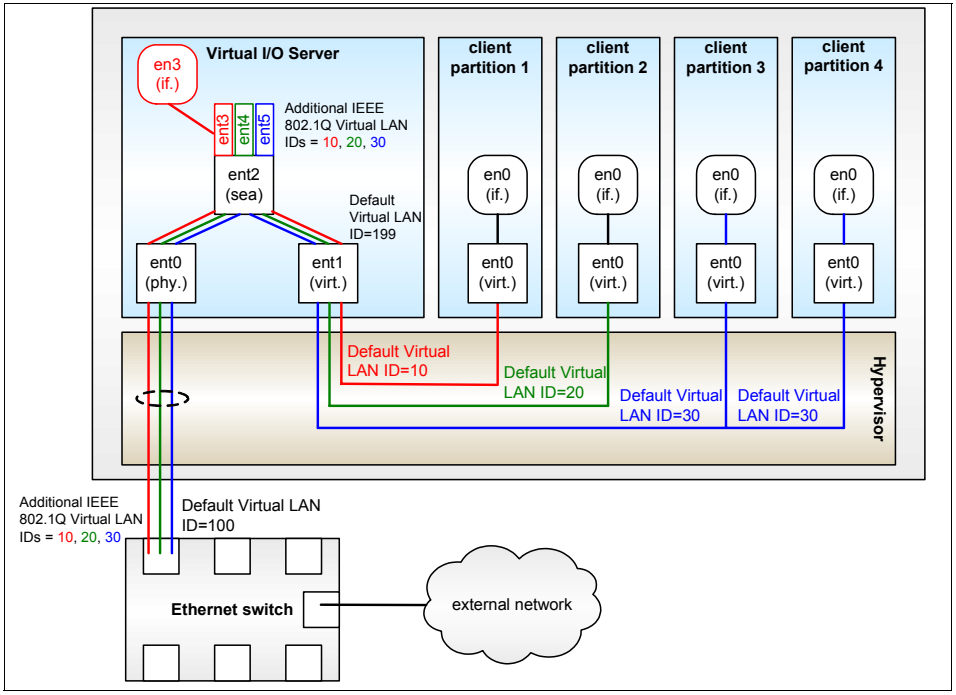


Figure 4-27 VLAN configuration scenario

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

4.13.3 Configuring the client partitions

As shown in Figure 4-27 on page 321, the configuration and setup of the internal VLAN is a straightforward process. Using the scenario shown in Figure 4-27 on page 321, 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 4-27 on page 321. Figure 4-28 on page 323 shows an example of creating a virtual Ethernet adapter for client partition 1.
4. Make sure that both the Access external network and IEEE 802.1Q compatible adapter flags are cleared.
5. Finalize the partition creation.

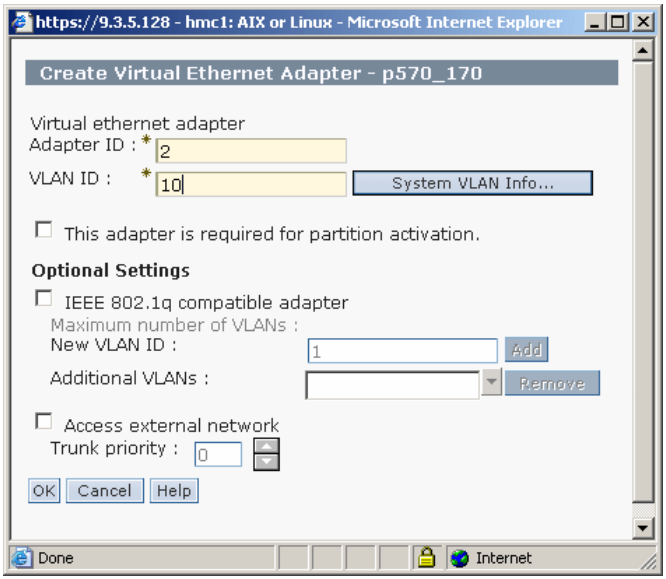


Figure 4-28 Virtual Ethernet configuration for the client partition using the HMC

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. The Hypervisor will strip them off before delivery. For example, as shown in Figure 4-27 on page 321, 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.

4.13.4 Configuring the Virtual I/O Server

The Virtual I/O Server is the 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 4-27 on page 321, 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 4-29 on page 324. 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. To display a list of the VLANs that have been added, click the arrow next to the **additional VLANs** field.
5. Make sure that the **Access external network** flag is selected. Leave the Trunk priority as the default unless using a SEA failover configuration.

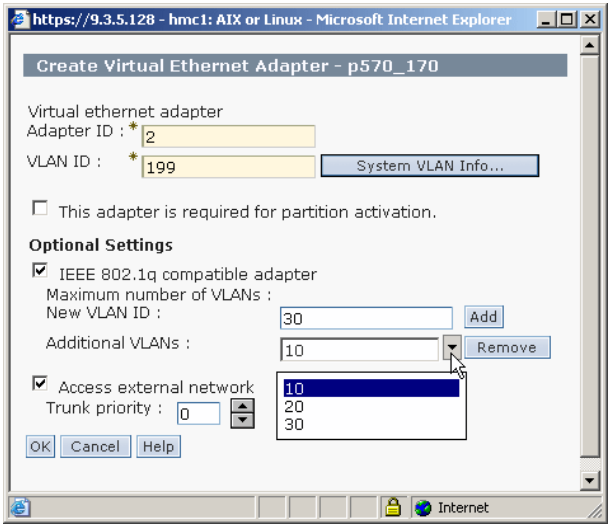


Figure 4-29 Virtual Ethernet configuration for Virtual I/O Server using the HMC

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.

4.13.5 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 4-29 on page 324, 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).

4.13.6 Configuring the Shared Ethernet Adapter for VLAN use

In this scenario, use the **mkvdev -sea** command on the Virtual I/O Server. This 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 Ethernet adapter 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: This creates a new Ethernet adapter for each of the previous commands.

The addition of VLAN interfaces to the SEA adapter is only necessary if the VIO Server itself needs to communicate on these VLANs.

3. Using the **mktcpip** command or the **cfgassist** 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 4-27 on page 321, 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.

4.13.7 Extending multiple VLANs into client partitions

There are two different ways how VLANs can be configured in a client partition. Figure 4-30 on page 328 shows a configuration where both Partition 1 and Partition 2 can communicate on two different VLANs.

Partition 1 has only one Virtual Ethernet adapter that has an unused PVID configured and VLANs 10 and 80 configured as 802.1Q VLANs. For VLAN 10 and 80 there is a separate network interface configured on the Virtual Ethernet adapter. A network packet sent through en0 is tagged with VLAN ID 80 in the client partition. No VLAN tag is added by the Hypervisor switch. The VLAN tag is not stripped off by the SEA because the VLAN ID does not match the PVID of the virtual Ethernet adpater in the SEA.

Partition 2 has two virtual Ethernet adapters where the VLAN is configured using the PVID. One is configured for VLAN 20 and the other for VLAN 90. No VLAN interfaces need to be configured in the client partition. A network packet sent through en0 is tagged with VLAN ID 90 by the Hypervisor switch. The VLAN tag is not stripped off by the SEA because the VLAN ID does not match the PVID of the virtual Ethernet adpater in the SEA. This method is normally the preferred one because it is easier to configure in the client partition.

For dynamic LPAR operations the HMC needs connectivity to all the partitions. Therefore, it is important to consider in which network the HMC will be placed. Since the HMC does not offer VLAN tagging a solution can be to put the HMC behind a firewall as shown in Figure 4-30 on page 328 and Figure 4-31 on page 329.

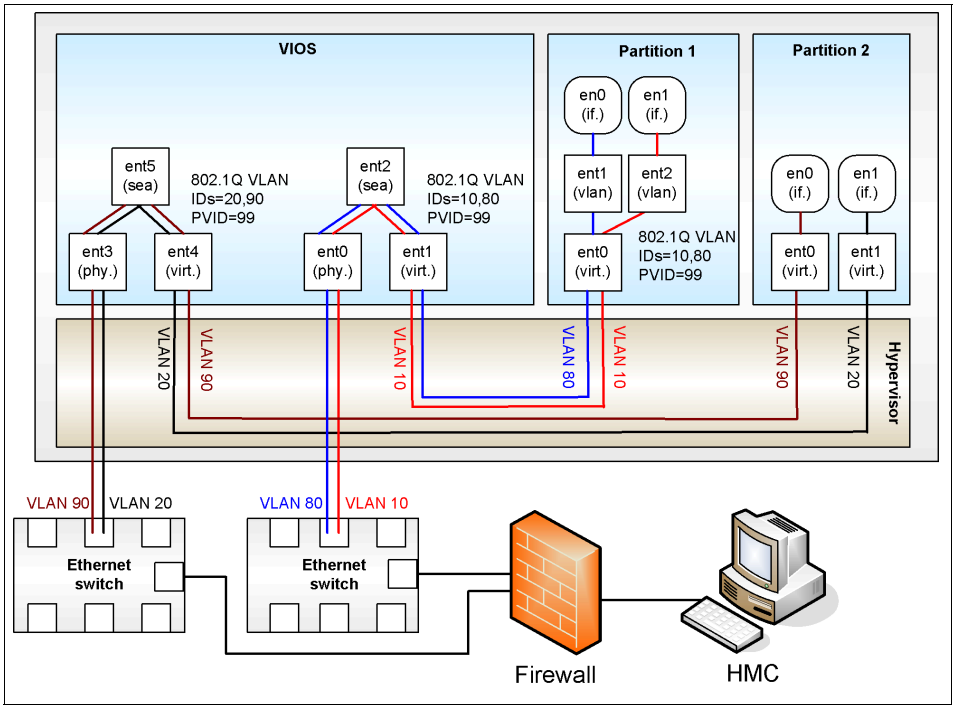


Figure 4-30 HMC in a VLAN tagged environment

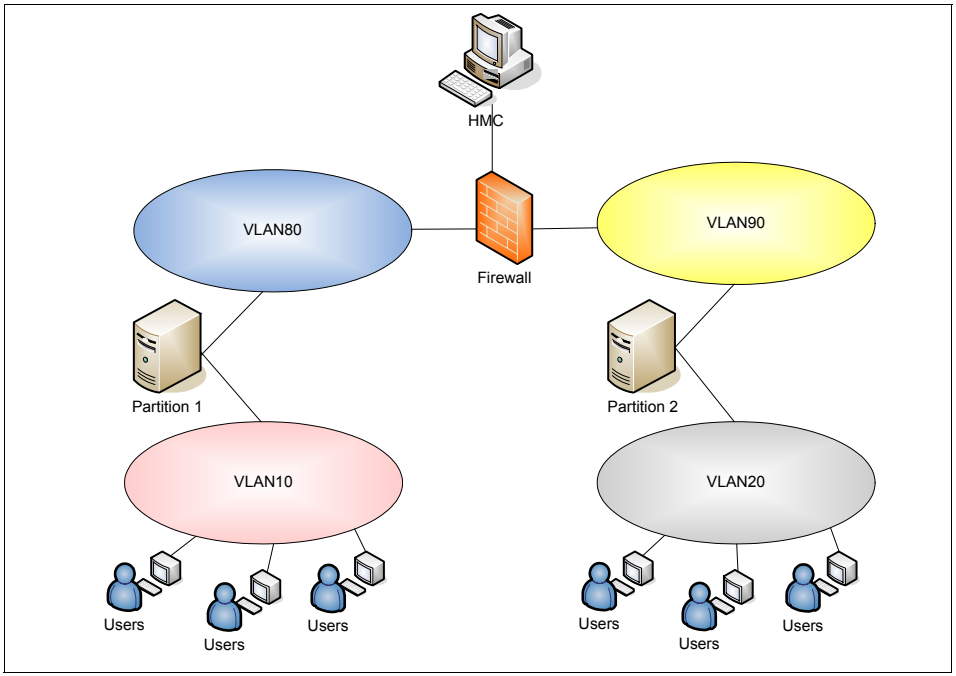


Figure 4-31 Cross-network VLAN tagging with a single HMC

4.13.8 Virtual Ethernet and SEA considerations

The following are for your consideration when implementing virtual Ethernet and Shared Ethernet Adapters in the Virtual I/O Server:

- Virtual Ethernet requires a POWER5 or POWER6 processor-based system running IVM or an HMC to define the virtual Ethernet adapters.
- Virtual Ethernet is available on all POWER5 or POWER6 processor-based systems, while Shared Ethernet Adapter and the Virtual I/O Server require the configuration of the PowerVM Standard Edition or PowerVM Enterprise Edition feature on some models.
- Virtual Ethernet can be used in both shared and dedicated processor partitions.
- Virtual Ethernet does not require a Virtual I/O Server for communication between partitions in the same system.
- A maximum of up to 256 virtual Ethernet adapters are permitted per partition.
- Each virtual Ethernet adapter is capable of being associated with up to 21 VLANs (20 VIDs and 1 PVID).

- A system can support up to 4096 different VLANs, as defined in the IEEE802.1Q standard.
- The partition must be running a minimum level of AIX Version 5.3 or Linux with the 2.6 kernel or a kernel that supports virtual Ethernet.
- A mixture of virtual Ethernet connections, real network adapters, or both are permitted within a partition.
- Virtual Ethernet can only connect partitions within a single system.
- Virtual Ethernet connections between AIX and Linux partitions are supported.
- Virtual Ethernet uses the system processors for all communication functions instead of off-loading that load to processors on network adapter cards. As a result, there is an increase in system processor load generated by the use of virtual Ethernet.
- Up to 16 virtual Ethernet adapters with 21 VLANs (20 VID and 1 PVID) on each can be associated to a Shared Ethernet Adapter, sharing a single physical network adapter.
- There is no explicit limit on the number of partitions that can attach to a VLAN. In practice, the amount of network traffic will limit the number of clients that can be served through a single adapter.
- To provide highly available virtual Ethernet connections to external networks, two Virtual I/O Servers with Shared Ethernet Adapter Failover or another network HA mechanism has to be implemented.
- You cannot use SEA failover with Integrated Virtualization Manager (IVM), because IVM only supports a single Virtual I/O Server.

4.14 Supported configurations

This section discusses various supported configurations in a Virtual I/O Server environment. The following configurations are described:

- Supported virtual SCSI configurations with:
  - Mirrored virtual SCSI devices on the client and server
  - Multi-path configurations in a SAN environment
- Virtual devices and HACMP support
- Virtual devices and GPFS support

The configurations described in this section are not a complete list of all available supported configurations. We show a collection of the most frequently adopted configurations that meet the requirements of most production environments.

For the latest information about Virtual I/O Server supported environments, visit the following Web site:

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

4.14.1 Supported virtual SCSI configurations

This discussion on virtual SCSI configurations is a detailed one about supported and recommended configurations when using the Virtual I/O Server. It is an extension to the discussion in 4.1, “Providing continuous availability for virtual I/O” on page 244. An understanding of the principles of redundancy and availability is assumed.

Refer to 2.6, “Virtual SCSI introduction” on page 75 for considerations when using logical volumes on the Virtual I/O Server as virtual disks for client partitions.

Supported configurations with mirrored VSCSI devices

Figure 4-32 on page 332 shows a supported way for mirroring disks with only one Virtual I/O Server.

On the Virtual I/O Server, you either configure two logical volumes and map them to the vhost adapter assigned to the client partition or you directly map the hdisks to the vhost adapter. On the client partition, the mapped devices will appear as two disks. The client mirrors the two virtual disks using standard AIX LVM mirroring.

In Figure 4-32, two separate physical disks on the Virtual I/O Server are attached to two separate physical adapters for continuous availability. This scenario also works when the disks are attached to only one physical adapter, but in this case it does not protect against adapter failure.

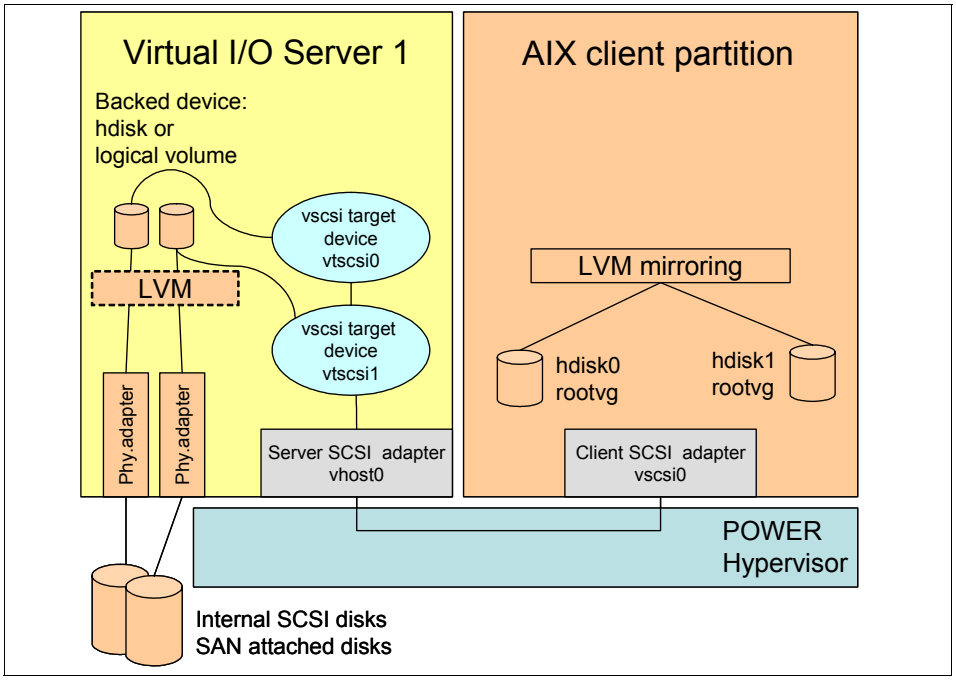


Figure 4-32 Supported and recommended ways to mirror virtual disks

Important: A logical volume of the Virtual I/O Server used as a virtual disk should not span multiple disks.

You can verify that a logical volume is confined to a single disk with the **`lslv -pv lvname`** command. The output of this command should only display a single disk.

Using mirroring, striping, or concatenation of physical disks using the LVM in the VIO client, or to use such features of special RAID-capable host bus adapters or storage subsystems on the Virtual I/O Server is recommended. Thus, to provide redundancy for the backed disk, a hardware RAID 5 array on the Virtual I/O Server can be used. Figure 4-33 shows the Virtual I/O Server configured with a SCSI RAID adapter.

A list of supported adapters can be found at:

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

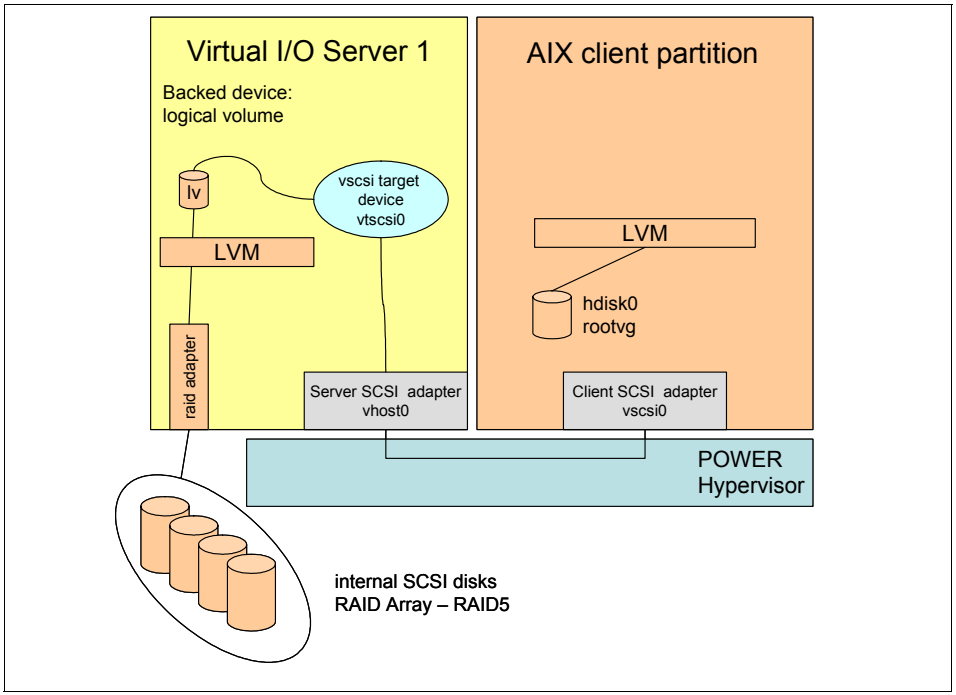


Figure 4-33 RAID5 configuration using a RAID adapter on the Virtual I/O Server

Attention: Only RAID hardware is supported for this configuration.

After creating a RAID 5 array, it will appear as one hdisk on the Virtual I/O Server. You can then divide the large disk into logical volumes and map them to your client partitions. Note that in this configuration the RAID adapter is a single point of failure.

When using this configuration plan for two additional disks for the installation of the Virtual I/O Server, that should be mirrored over two physical disks.

Important: Do not use the Virtual I/O Server rootvg for logical volumes that will be exported as virtual disks for the clients.

When planning for multiple Virtual I/O Servers, Figure 4-34 shows the recommended way for mirroring the virtual disks on the client partitions.

Note: One Virtual I/O Server is supported on the Integrated Virtualization Manager (IVM). Use the HMC to manage more than one Virtual I/O Server.

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

Either configure a logical volume on each Virtual I/O Server and map it to the vhost adapter that is assigned to the same client partition or directly map an hdisk to the appropriate vhost adapter. The mapped logical volume or hdisk will be configured as an hdisk on the client side, each belonging to a different Virtual I/O Server. Then use LVM mirroring on the client site.

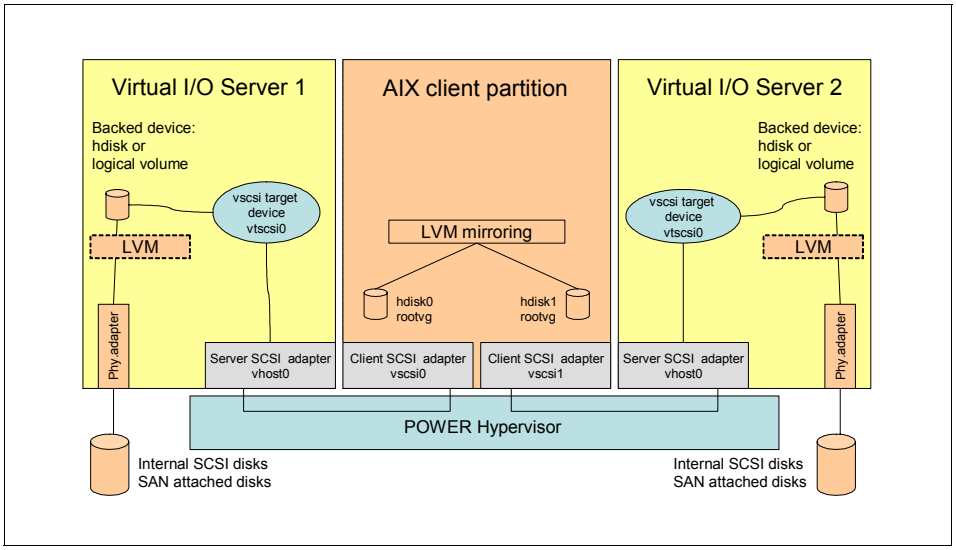


Figure 4-34 Recommended way to mirror virtual disks with two Virtual I/O Server

Supported multi-path configurations in a SAN environment

When discussing a supported configuration with MPIO, you need to distinguish two different scenarios:

- One Virtual I/O Server attaching a LUN in the SAN over more than one path. In this case, you only need to implement multi-path software on the Virtual I/O Server.
- Having multiple Virtual I/O Servers connect to the same LUN and backed up to the same client. In this case, the client partition uses MPIO to access the virtual disk as a single device. You may also consider using multi-path software on the Virtual I/O Server to access the LUN over more than one path for path failover and load balancing.

Note: This book only covers IBM storage solutions. For support statements of other storage vendors, contact your IBM representative or your storage vendor directly and ask for specifications on supported configurations.

Supported IBM TotalStorage solutions

At the time of writing, the supported TotalStorage Enterprise Server includes the following models:

- 2105 Enterprise Storage Server (models 800, 750, and Fxx)
- 2107 Model 921 IBM TotalStorage DS8100
- 2107 Model 922 IBM TotalStorage DS8300
- 2107 Model 9A2 IBM TotalStorage DS8300
- 2107 Model 92E IBM TotalStorage DS8000 Expansion Unit
- 2107 Model 9AE IBM TotalStorage DS8000 Expansion Unit
- 1750 Model 511 IBM TotalStorage DS6800
- 1750 Model EX1 IBM TotalStorage DS6000 Expansion Unit

When attaching the Virtual I/O Server to an IBM TotalStorage DS Family, the RDAC driver is used on the Virtual I/O Server. At the time of writing, the IBM TotalStorage DS Family includes the following models:

- DS4100 (FAStT100)
- DS4200 (FAStT200)
- DS4300 (FAStT600)
- DS4400 (FAStT700)
- DS4500 (FAStT900)
- DS4700

- DS4800
- FAStT500 Storage Server

At the time of writing the supported models from the IBM System Storage N series includes the following models:

- N3700
- N5000
- N7000

For other supported IBM Storage solutions, such as the TotalStorage SAN Volume Controller, check the official IBM Web site:

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

Supported scenarios using one Virtual I/O Server

Figure 4-35 on page 337 represents a configuration with MPIO with only one Virtual I/O Server attached to IBM TotalStorage Enterprise Storage Systems. The LUN is connected over two Fibre Channel adapters to the Virtual I/O Server to increase redundancy or throughput.

Because the disk is only attached to one Virtual I/O Server, it is possible to create logical volumes and map them to the vhost adapter that is assigned to the appropriate client partition. You can also choose to map the disk directly to the vhost adapter.

For attaching the IBM TotalStorage Enterprise Storage Server to only one Virtual I/O Server, MPIO with the SDDPCM or SDD is supported.

Important: The recommended method to attach IBM TotalStorage Enterprise Storage Systems is MPIO. Virtual disk devices created with SDD may require a migration effort in the future.

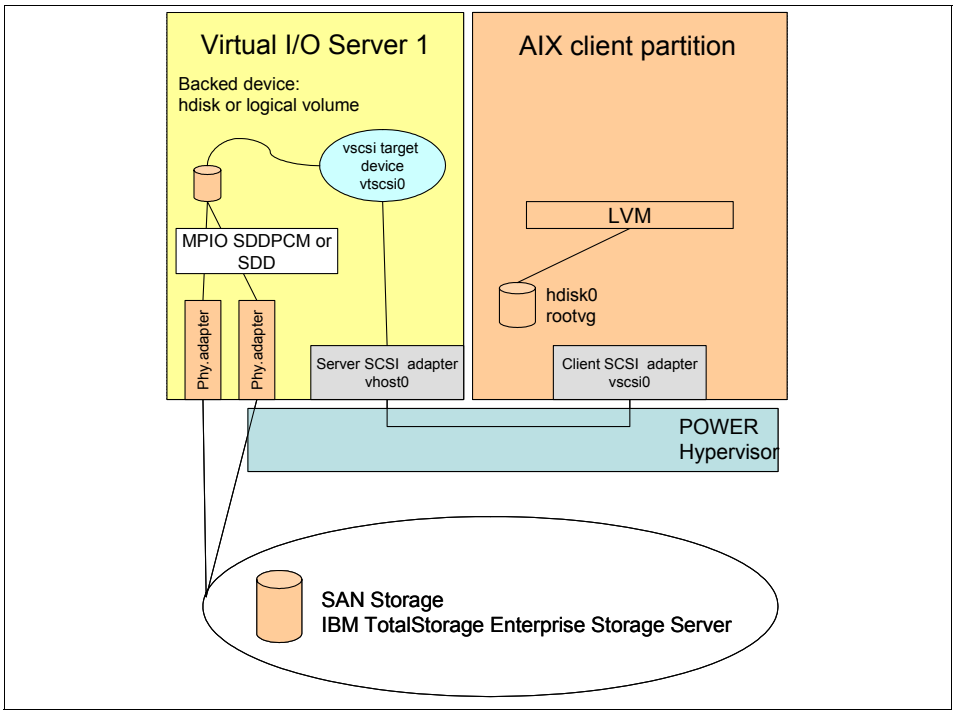


Figure 4-35 Using MPIO on the Virtual I/O Server with IBM TotalStorage

Figure 4-36 shows the configuration with only one Virtual I/O Server for IBM TotalStorage DS Family using RDAC.

Only the RDAC driver is supported for attaching the IBM TotalStorage DS Family. Because the RDAC driver is using the IEEE volume identifier, there are no migration issues.

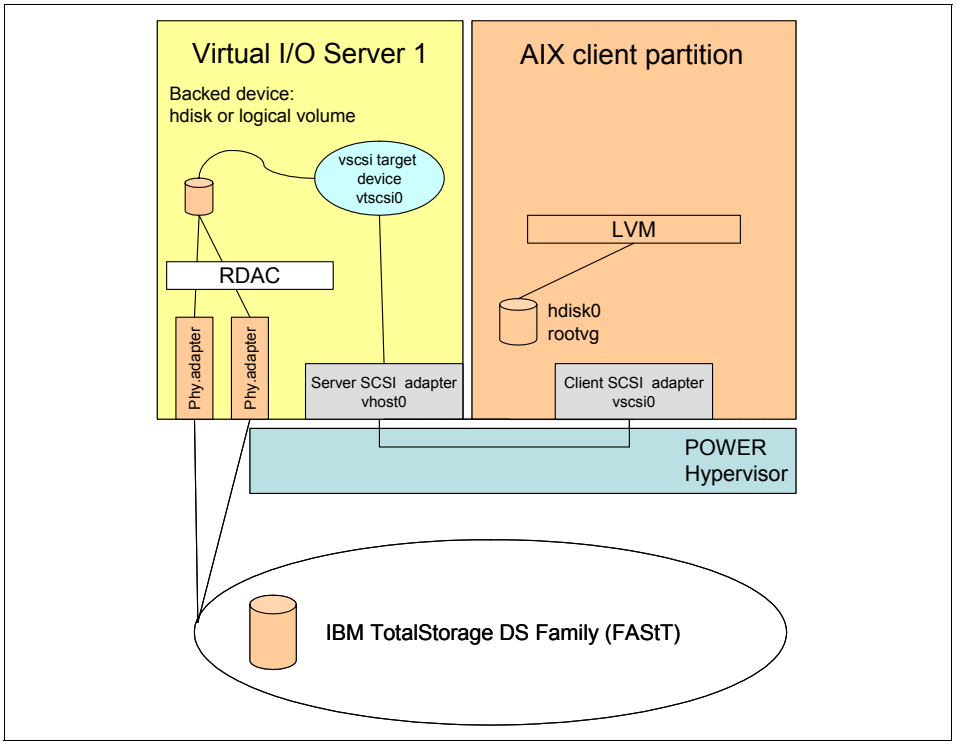


Figure 4-36 Using RDAC on the Virtual I/O Server with IBM TotalStorage

Supported scenarios using multiple Virtual I/O Servers

When configuring multiple Virtual I/O Servers, Figure 4-37 shows a supported configuration when attaching IBM TotalStorage Enterprise Storage Systems using multi-path software on the Virtual I/O Server for additional redundancy and throughput.

Attention: When using multiple Virtual I/O Servers, and exporting the same LUN to the client partitions, only mapping of hdisks is supported (logical volumes are not supported).

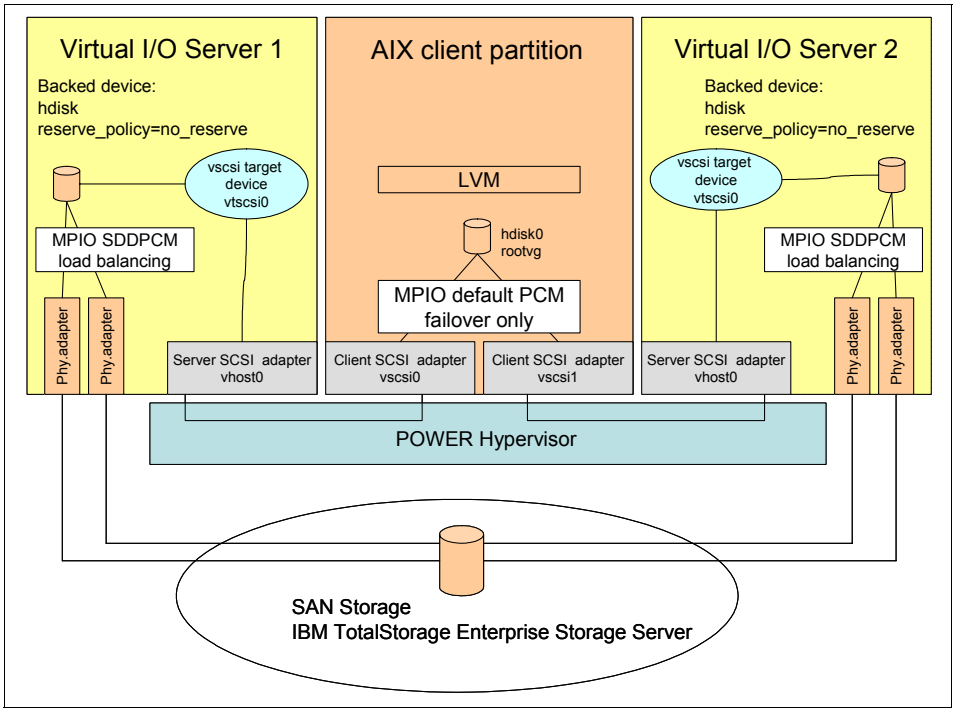


Figure 4-37 Configuration for multiple Virtual I/O Server and IBM ESS

Attaching IBM TotalStorage Enterprise Storage Servers to the Virtual I/O Servers, only MPIO using the SDDPCM is supported when you are doing MPIO in the client partition.

You can also attach IBM TotalStorage DS Family using the RDAC driver, as shown in Figure 4-38.

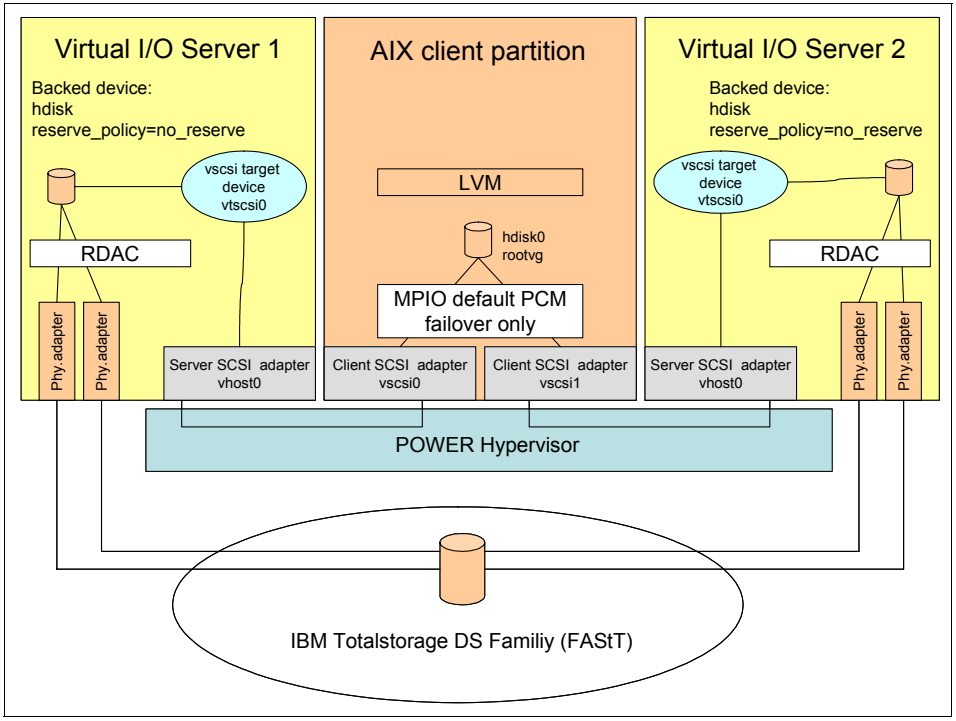


Figure 4-38 Configuration for multiple Virtual I/O Servers and IBM FAStT

On both Virtual I/O Servers, you have to set the hdisk attribute *reserve_policy* to no. This attribute prevents the Virtual I/O Server setting a reservation flag on the disk at the time of mapping. The MPIO component on the client partition will take the responsibility of managing the disk.

On the client partition MPIO, using the default PCM is supported, which only allows a failover policy and no load balancing. Only one path to the disks is active; the other path is used in the event that the active path fails, for example, when the Virtual I/O Server that serves the disk over the active path is rebooted.

It is possible to choose the active path on the client side. Users may manually configure the active paths for clients, enabling you to spread the workload evenly across the Virtual I/O Server. For detailed configuration steps, refer to 4.8, “Scenario 3: MPIO in the client with SAN” on page 275.

4.14.2 HACMP for virtual I/O clients

The IBM HACMP software provides a computing environment that ensures that mission-critical applications running in AIX stand-alone servers or partitions can recover quickly from hardware and software failures. The HACMP software is a high availability system that ensures that critical resources for an application are available for processing in a cluster of servers or partitions. High availability combines custom software with hardware to minimize downtime by quickly restoring services if a system, component, or application fails.

In order to help clients improve the availability for their virtualized servers, IBM tested and certified HACMP with the PowerVM features of virtual SCSI and virtual Ethernet. Clients can design highly available clusters of dedicated-processor partitions or micro-partitions that use services from Virtual I/O Servers for their critical applications. Information about the announcement of support for HACMP for the PowerVM feature can be found at:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10390

The tasks related to maintaining active cluster storage resources in each of the Virtual I/O Servers that serve AIX client partitions (cluster nodes) are handled by the combination of certified software levels and the configuration of disk drives and volume groups. The most important considerations include:

- The external storage content (volume groups and logical volumes) is handled at the AIX client partition level. Virtual I/O Servers only connect external storage with HACMP/AIX nodes (client partitions).
- No disk reservations apply at the hardware level. Storage access, depending on the configuration, is a combination of HACMP/AIX LVM.
- All the volume groups must be Enhanced Concurrent volume groups regardless of whether they are used in concurrent access mode or not.
- If any HACMP/AIX nodes access the volume groups through a Virtual I/O Server, then all nodes must access through the Virtual I/O Server.
- All the HACMP/AIX nodes must use the volume groups with the same basis, concurrent or not concurrent.

Figure 4-39 shows the basic issues for storage of AIX client partitions and HACMP.

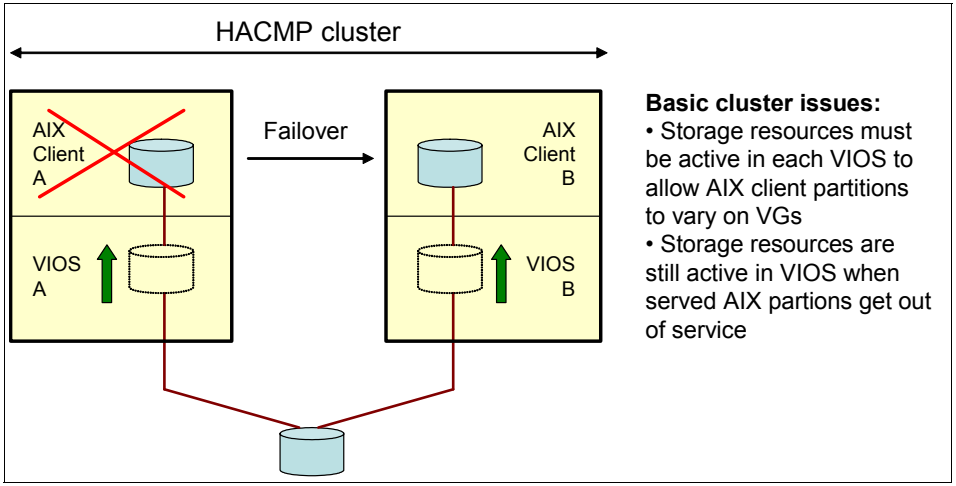


Figure 4-39 Basic issues for storage of AIX client partitions and HACMP

This section outlines the supported configurations for using HACMP with the virtual Ethernet and virtual SCSI features. Such configurations should be accomplished with the minimum levels of software and specific virtual Ethernet and virtual SCSI attached disk configurations.

HACMP and virtual SCSI

The volume group must be defined as Enhanced Concurrent Mode. In general, Enhanced Concurrent Mode is the recommended mode for sharing volume groups in HACMP clusters because volumes are accessible by multiple HACMP nodes, resulting in faster failover in the event of a node failure.

If file systems are used on the standby nodes, they are not mounted until the point of failover, because the volume groups are in full active read/write mode only on the home node; the standby nodes have the volume groups in passive mode, which does not allow access to the logical volumes or file systems. If shared volumes (raw logical volumes) are accessed directly in Enhanced Concurrent Mode, these volumes are accessible from multiple nodes, so access must be controlled at a higher layer, such as databases.

If any cluster node accesses shared volumes through virtual SCSI, all nodes must do so. This means that disks cannot be shared between a partition using virtual SCSI and a server or partition directly accessing those disks.

All volume group construction and maintenance on these shared disks is done from the HACMP nodes using C-SPOC, not from the Virtual I/O Server.

Note: Partitions using virtual I/O cannot be mixed in an HACMP cluster with partitions using dedicated adapters accessing the same shared disks.

HACMP and virtual Ethernet

IP Address Takeover (IPAT) through aliasing must be used. IPAT using replacement and MAC Address Takeover are not supported. In general, IPAT using aliasing is recommended for all HACMP networks that can support it.

All virtual Ethernet interfaces defined to HACMP should be treated as single-adapter networks. In particular, configure the file netmon.cf to include a list of clients to ping. It must be used to monitor and detect failure of the network interfaces. Due to the nature of virtual Ethernet, other mechanisms to detect the failure of network interfaces are not effective.

If the Virtual I/O Server has only a single physical interface on a network (instead of, for example, two interfaces with Ethernet aggregation), then a failure of that physical interface will be detected by HACMP on the AIX client partition. However, that failure will isolate the node from the network. In this case it is recommended to use a second virtual Ethernet on the AIX client partition.

There are several ways to configure AIX client partitions and Virtual I/O Server resources for additional high availability with HACMP. We recommend using two Virtual I/O Servers. This configuration also allows online service for virtual I/O with improved uptime for user applications. In Figure 4-40, there is an example of a HACMP cluster between two AIX client partitions.

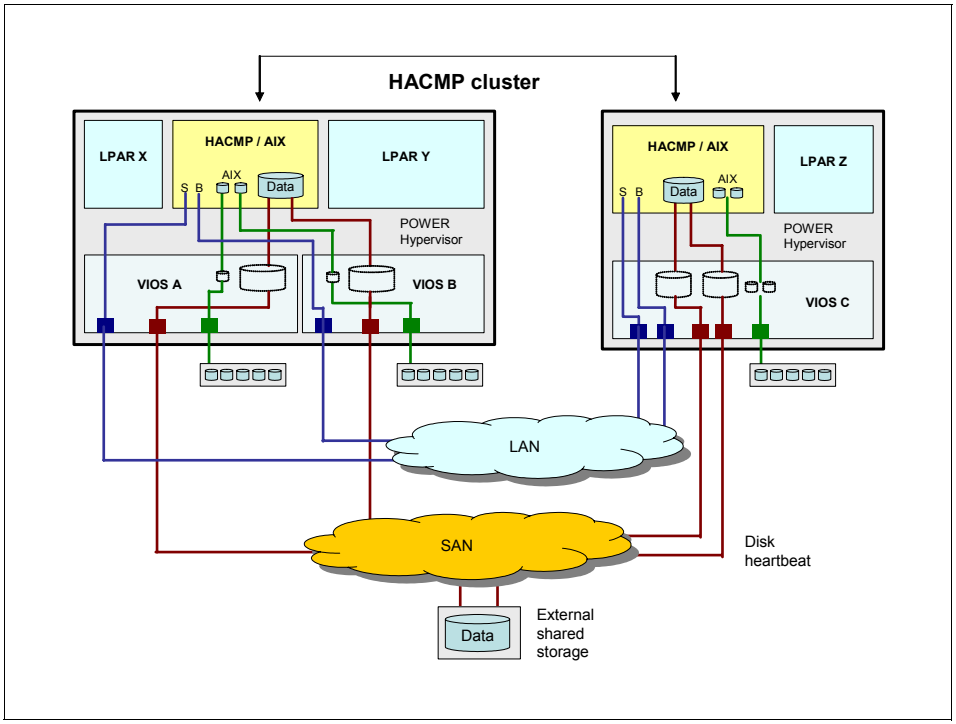


Figure 4-40 Example of HACMP cluster between two AIX client partitions

Figure 4-41 shows the basic considerations when configuring an AIX client partition with HACMP to be part of a high availability cluster of nodes using virtual Ethernet and virtual SCSI services from two Virtual I/O Servers.

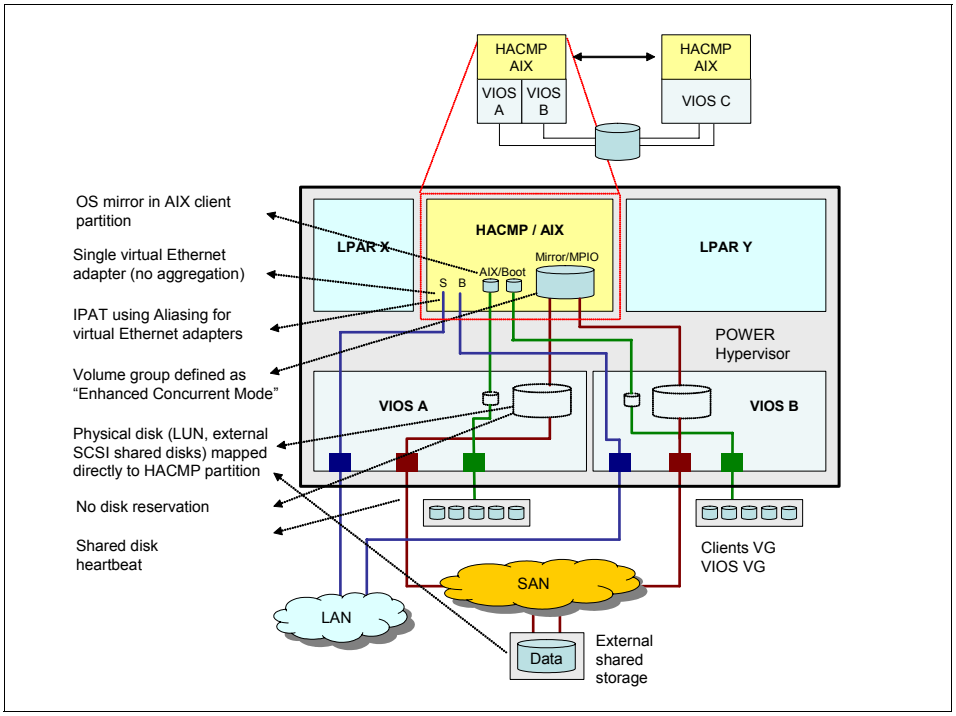


Figure 4-41 Example of an AIX client partition with HACMP using two VIOSs

The following publications can be useful for clients planning to configure HACMP clusters with AIX client partitions:

- *High Availability ClusterMulti-Processing for AIX: Concepts and Facilities Guide*, SC23-4864
- *High Availability ClusterMulti-Processing for AIX: Planning and Installation Guide*, SC23-4861
- *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, SG24-6769

4.14.3 Concurrent disks in client partitions

Normally, a disk is assigned to one machine and this machine is responsible for handling the access to files or raw devices on that disk. Concurrent disks are disks that are accessed from more than one machine at the same time. This happens regularly with clustering software and application software that is

designed to run on more than one machine in parallel. It is possible to implement concurrent virtual disks on one physical machine, using Virtual I/O Servers on that machine. For redundancy reasons with clustering software, you should use at least two machines if a redundant environment is needed.

A storage subsystem is required that can assign a disk to more than one machine. This can be some kind of Fibre Channel disk, for example, a DS4000 disk subsystem. Additionally, for smooth operation, we recommend that the Virtual I/O Servers are connected redundantly to the storage subsystem.

On the operating system, the following has to be installed:

- bos.clvm as the basic layer, to enable concurrent access.
- The cluster software that is responsible for managing concurrent disks.
- The application software that is responsible for handling the locking mechanism on files or raw devices on the disk.

The following steps show how to set up concurrent disk access for a SAN disk that is assigned to two client partitions. Each client partition sees the disk through two Virtual I/O Servers. On the disk an enhanced concurrent volume group is created. This volume group is varied on in active mode in the first partition. This means that updates to the file systems are possible. On the second partition the volume group is varied on in read-only mode. This means that no updates to the file systems are possible. This kind of configuration can be used to build a two-node HACMP test cluster on a single machine.

1. Create the disk on the storage device.

   Follow the directions provided by the storage subsystem provider.

2. Assign the disk to the Virtual I/O Servers.

   How to do that depends on the type of storage subsystem. On a DS4000 type disk subsystem, for example, it would be assigned to a host group.

3. On the first Virtual I/O Server, scan for the newly assigned disk:

   ```
   $ cfgdev
   ```

4. On the first Virtual I/O Server, change the SCSI reservation of that disk to no_reserve so that the SCSI reservation bit on that disk is not set if the disk is accessed:

   ```
   $ chdev -dev hdiskN -attr reserve_policy=no_reserve
   ```

   (where *N* is the number of the disk in question; reservation commands are specific to the multipathing disk driver in use. This parameter is used with DS4000 disks; it can be different with other disk subsystems.)

5. On the first Virtual I/O Server, assign the disk to the first partition:

   ```
   $ mkvdev -vdev hdiskN -vadapter vhostN [ -dev Name ]
   ```

   where *N* is the number of the disk respectively; the vhost in question and the device name can be chosen to your liking but also left out entirely; the system will then create a name automatically.

6. On the first Virtual I/O Server, assign the disk to the second partition:

   ```
   $ mkvdev -f -vdev hdiskN -vadapter vhostN [ -dev Name ]
   ```

   Note: You must specify the -f flag so that an already assigned disk can be assigned again.

7. On the second Virtual I/O Server, scan for the disk:

   ```
   $ cfgdev
   ```

8. On the second Virtual I/O Server, change the SCSI reservation of that disk:

   ```
   $ chdev -dev hdiskN -attr reserve_policy=no_reserve
   ```

9. On the second Virtual I/O Server, assign the disk to the first partition:

   ```
   $ mkvdev -vdev hdiskN -vadapter vhostN [ -dev Name ]
   ```

10. On the second Virtual I/O Server, assign the disk to the second partition:

    ```
    $ mkvdev -f -vdev hdiskN -vadapter vhostN [ -dev Name ]
    ```

11. On the first partition, scan for that disk:

    ```
    # cfgmgr
    ```

12. On the first partition, create an enhanced concurrent capable volume group with a file system:

    ```
    # mkvg -C -y sharedvg hdiskN
    # mklv -y sharedlv -t jfs2 sharedvg 10M
    # crfs -v jfs2 -d sharedlv -m /sharedfs
    # mount /sharedfs
    # touch /sharedfs/testfile
    ```

13. The system will inform you that the volume group is not varied on automatically and that it has to be done manually. Although it is possible to create an enhanced concurrent capable volume group without cluster manager software installed, it is not recommend. All accesses to the enhanced concurrent enabled volume group are not coordinated between both servers, which can lead to data corruption on this volume group.

14. On the second partition, scan for the disk:

    ```
    # cfgmgr
    ```

15. On the second partition, import the volume group:

```
# importvg -n -y sharedvg hdiskN
```

16. On the second partition varyon volume group in read-only mode and mount the file system in read-only mode. You should see the test file in the /sharefs file system.

```
# varyonvg -r sharedvg
# mount -r /sharedfs
# cd /sharedfs
# ls
```

4.14.4 General Parallel Filesystem

General Parallel Filesystem (GPFS) allows the exploitation of Virtual I/O Server configurations. Both the virtual SCSI and the Shared Ethernet Adapter are supported in single and multiple System p server configurations. At the time of writing the support was limited to GPFS nodes that are using AIX 5L V5.3.

All partitions in a System p server that are GPFS cluster members must have all the virtual I/O disks mapped to each LPAR using virtual SCSI. This creates to GPFS a SAN environment where each node has access to disks on a local path without requiring network access. All of the NSDs in these configurations must not be coded with any NSD server associated with them.

For details or updates on supported configurations, see the GPFS FAQ at:

http://publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

Abbreviations and acronyms

ABI	Application Binary Interface
AC	Alternating Current
ACL	Access Control List
AFPA	Adaptive Fast Path Architecture
AIO	Asynchronous I/O
AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report
API	Application Programming Interface
ARP	Address Resolution Protocol
ASMI	Advanced System Management Interface
BFF	Backup File Format
BIND	Berkeley Internet Name Domain
BIST	Built-In Self-Test
BLV	Boot Logical Volume
BOOTP	Boot Protocol
BOS	Base Operating System
BSD	Berkeley Software Distribution
CA	Certificate Authority
CATE	Certified Advanced Technical Expert
CD	Compact Disk
CDE	Common Desktop Environment
CD-R	CD Recordable
CD-ROM	Compact Disk-Read Only Memory
CEC	Central Electronics Complex
CHRP	Common Hardware Reference Platform
CLI	Command Line Interface
CLVM	Concurrent LVM
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSM	Cluster Systems Management
CUoD	Capacity Upgrade on Demand
DCM	Dual Chip Module
DES	Data Encryption Standard
DGD	Dead Gateway Detection
DHCP	Dynamic Host Configuration Protocol
DLPAR	Dynamic LPAR
DMA	Direct Memory Access
DNS	Domain Naming System
DRM	Dynamic Reconfiguration Manager
DR	Dynamic Reconfiguration
DVD	Digital Versatile Disk
EC	EtherChannel
ECC	Error Checking and Correcting
EOF	End of File
EPOW	Environmental and Power Warning
ERRM	Event Response resource manager
ESS	Enterprise Storage Server
F/C	Feature Code
FC	Fibre Channel
FCAL	Fibre Channel Arbitrated Loop

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

FDX	Full Duplex
FLOP	Floating Point Operation
FRU	Field Replaceable Unit
FTP	File Transfer Protocol
GDPS®	Geographically Dispersed Parallel Sysplex™
GID	Group ID
GPFS	General Parallel File System™
GUI	Graphical User Interface
HACMP	High Availability Cluster Multiprocessing
HBA	Host Bus Adapters
HMC	Hardware Management Console
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
I/O	Input/Output
IBM	International Business Machines
ID	Identification
IDE	Integrated Device Electronics
IEEE	Institute of Electrical and Electronics Engineers
IP	Internetwork Protocol
IPAT	IP Address Takeover
IPL	Initial Program Load
IPMP	IP Multipathing
ISV	Independent Software Vendor
ITSO	International Technical Support Organization
IVM	Integrated Virtualization Manager
JFS	Journaled File System
L1	Level 1
L2	Level 2
L3	Level 3
LA	Link Aggregation
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LMB	Logical Memory Block
LPAR	Logical Partition
LPP	Licensed Program Product
LUN	Logical Unit Number
LV	Logical Volume
LVCB	Logical Volume Control Block
LVM	Logical Volume Manager
MAC	Media Access Control
Mbps	Megabits Per Second
MBps	Megabytes Per Second
MCM	Multichip Module
ML	Maintenance Level
MP	Multiprocessor
MPIO	Multipath I/O
MTU	Maximum Transmission Unit
NFS	Network File System
NIB	Network Interface Backup
NIM	Network Installation Management
NIMOL	NIM on Linux
NVRAM	Non-Volatile Random Access Memory
ODM	Object Data Manager
OSPF	Open Shortest Path First
PCI	Peripheral Component Interconnect
PIC	Pool Idle Count
PID	Process ID
PKI	Public Key Infrastructure
PLM	Partition Load Manager

POST	Power-On Self-test
POWER	Performance Optimization with Enhanced Risc (Architecture)
PPC	Physical Processor Consumption
PPFC	Physical Processor Fraction Consumed
PTF	Program Temporary Fix
PTX	Performance Toolbox
PURR	Processor Utilization Resource Register
PV	Physical Volume
PVID	Physical Volume Identifier
PVID	Port Virtual LAN Identifier
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
RCP	Remote Copy
RDAC	Redundant Disk Array Controller
RIO	Remote I/O
RIP	Routing Information Protocol
RISC	Reduced Instruction-Set Computer
RMC	Resource Monitoring and Control
RPC	Remote Procedure Call
RPL	Remote Program Loader
RPM	Red Hat Package Manager
RSA	Rivet, Shamir, Adelman
RSCT	Reliable Scalable Cluster Technology
RSH	Remote Shell
SAN	Storage Area Network
SCSI	Small Computer System Interface
SDD	Subsystem Device Driver
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor
SMS	System Management Services
SMT	Simultaneous Multithreading
SP	Service Processor
SPOT	Shared Product Object Tree
SRC	System Resource Controller
SRN	Service Request Number
SSA	Serial Storage Architecture
SSH	Secure Shell
SSL	Secure Socket Layer
SUID	Set User ID
SVC	SAN Virtualization Controller
TCP/IP	Transmission Control Protocol/Internet Protocol
TSA	Tivoli System Automation
UDF	Universal Disk Format
UDID	Universal Disk Identification
VIPA	Virtual IP Address
VG	Volume Group
VGDA	Volume Group Descriptor Area
VGSA	Volume Group Status Area
VLAN	Virtual Local Area Network
VP	Virtual Processor
VPD	Vital Product Data
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
VSD	Virtual Shared Disk
WLM	Workload Manager

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 356. Note that some of the documents referenced here may be available in softcopy only.

- *AIX 5L Practical Performance Tools and Tuning Guide*, SG24-6478
- *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038
- *i5/OS on eServer p5 Models A Guide to Planning, Implementation, and Operation*, SG24-8001
- *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, SG24-6769
- *Introduction to pSeries Provisioning*, SG24-6389
- *Linux Applications on pSeries*, SG24-6033
- *Managing AIX Server Farms*, SG24-6606
- *NIM: From A to Z in AIX 4.3*, SG24-5524
- *NIM from A to Z in AIX 5L*, SG24-7296
- *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615
- *Integrated Virtualization Manager on IBM System p5*, REDP-4061
- *IBM AIX Version 6.1 Differences Guide*, SG24-7559
- *Introduction to Workload Partition Management in IBM AIX Version 6.1*, SG24-7431
- *IBM System p Live Partition Mobility*, SG24-7460
- *Integrated Virtual Ethernet Adapter Technical Overview and Introduction*, REDP-4340
- *PowerVM Virtualization on IBM System p Managing and Monitoring*, SG24-7590

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

- *Hardware Management Console V7 Handbook*, SG24-7491

Other publications

These publications are also relevant as further information sources:

- The following types of documentation are located through the Internet at the following URL:

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

  - User guides
  - System management guides
  - Application programmer guides
  - All commands reference volumes
  - Files reference
  - Technical reference volumes used by application programmers
- Detailed documentation about the PowerVM feature and the Virtual I/O Server:

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

Online resources

These Web sites are also relevant as further information sources:

These Web sites and URLs are also relevant as further information sources:

- AIX and Linux on POWER community

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

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

- IBM PowerVM on IBM System p Web page

  http://www.ibm.com/systems/p/apv/
- IBM System p and AIX Information Center

  http://publib16.boulder.ibm.com/pseries/index.htm
- IBM System Planning Tool

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/
- IBM Systems Hardware Information Center

  http://publib.boulder.ibm.com/infocenter/systems/scope/hw/index.jsp
- IBM Systems Workload Estimator

  http://www-304.ibm.com/jct01004c/systems/support/tools/estimator/index.html
- Latest *Multipath Subsystem Device Driver homepage*

  http://www-1.ibm.com/support/docview.wss?uid=ssg1S4000201
- Novell SUSE LINUX Enterprise Server information

  http://www.novell.com/products/server/index.html
- SCSI T10 Technical Committee

  http://www.t10.org
- SDDPCM software download page

  http://www.ibm.com/support/docview.wss?uid=ssg1S4000201
- Service and productivity tools for Linux on POWER

  https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html
- Virtual I/O Server supported environment

  http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html
- Virtual I/O Server documentation

  http://techsupport.services.ibm.com/server/vios/documentation/home.html
- Virtual I/O Server home page

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html
- Virtual I/O Server fixpack download page

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

How to get Redbooks

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

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols
.sysplan file extension 224

A
access external network flag 179, 200, 252
adapter
  list of supported 142
  network 90
  physical 32
  virtual SCSI 77
Advanced POWER Virtualization 3, 20
  Enterprise Edition 20, 24, 102
  feature codes 26
  IVM 71
  Live Partition Mobility 25, 102
  ordering 20
  Standard Edition 20, 63
  virtual Ethernet 83
  virtual SCSI 75
Advanced POWER Virtualization editions 4
  APV-EE 5
  APV-SE 4
Advanced System Management Interface 24
advanced virtualization configurations 243
affinity 31
AIX
  boot image 120
  EtherChannel 90
  Ethernet adapters and interfaces 90
  features 12
  link aggregation 90
  monitoring 32
  network adapter 90
  network interface backup 90
  SMIT 122
  Version 5.3 39
alt_root_vg command 147
APV 3
APV Enterprise Edition 147
ARP 94
assigned processor 111
autoyast 297
availability of virtualized systems 15

B
bindprocessor command 39
boot 72
boot image 120
boot mode, SMS 176
bootlist command 294
BootP 86
bosboot command 121, 294
bridging
  virtual Ethernet 92, 97
broadcast 94

C
cache 119
capacity
  upper boundary 38
Capacity Upgrade on Demand 123
  APV feature 4
  backup 124
  licensing 114
  on/off 123, 125
  permanent 123
  POWER5 14
  POWER6 15
  reserve 124
  trial 124
  utility CUoD 124
capped 129
capped micro-partition 36, 43, 127–128
cfgassist command 134, 182, 268
cfgdev command 183, 191
cfgmgr command 81, 191, 215, 218
chdev command 248–249, 277
chhwres command 315
chpath command 247, 286
chtcpip command 142
client adapter
  virtual SCSI 76
client partitions
  creating client partitions 197
command line interface
  IOSCLI 134
commands

© Copyright IBM Corp. 2004, 2005, 2007, 2008. All rights reserved.

AIX
arp 94
bindprocessor 39
bootlist 294
bosboot 120–121, 294
cfgmgr 81, 191, 215, 218
chdev 248
chpath 247, 286
chtcpip 142
chvg 143
crontab 142
dsh 192
errpt 295
ethchan_config 275
lparstat 32
lscfg 77, 191
lsdev 77, 142
lspath 283, 286
lspv 283, 295
lstcpip 142
mirrorvg 293
mkbdsp 142
mkvdev 191
mpstat 32
rset 39
schedo 39
smctl 120
smitty 122
smitty installios 139
smtctl 120–121
startnetsvc 142
stopnetsvc 142
syncvg 295
topas 142
varyonvg 245, 295
viostat 142
wkldagent 142
HMC
chhwres 315
installios 232
lshwres 316
mksysplan 148
OS_install 229
Linux
arp 94
brctl 94
vconfig 105
Other
fget_config 188, 278
VIOS
alt_root_vg 147
cfgassist 134, 182, 268
cfgdev 183, 191
chdev 249, 277
chsp 145
chtcpip 142
crontab 142
entstat 269
extendvg 195
help 135
installios 139–140, 173
license 178
lscfg 285
lsdev 142, 179, 183, 191, 250, 267
lslv 82
lsmap 183, 186, 281
lssp 146
mirrorios 196
mkbdsp 142, 146
mklv 184
mksp 145
mktcpip 182, 268
mkvdev 180, 219, 266, 272
mkvg 184
oem_setup_env 188
repphyvol 147
rmbdsp 146
rmdev 184
save_base 147
startnetsvc 142
stopnetsvc 142
su 147
tee 135
viosecure 142
wkldagent 142
wkldmgr 142
wkldout 142
concurrent volume groups 345
configuration
IVM 72
virtual Ethernet 97
virtual SCSI 76
console 27, 32
IVM 74
continuous availability 15–16, 244, 299, 303, 306
Create Logical Partition Wizard 155
crontab command 142
CUoD See Capacity Upgrade on Demand

D

Dead Gateway Detection, DGD 252
dedicated memory 35
dedicated processor 35, 40, 112, 126
dedicated-processor partition 40–41, 127
defaultid 272
DefaultPool 309
defining the Virtual I/O Server partition 153
device
  optical 33, 74
  virtual SCSI 76
device mapper multipath 306
DHCP 86
dispatch 29
DLPAR 9
  virtual SCSI 214
dsh command 192
DSH_LIST variable 192
DSH_REMOTE_CMD variable 192
dynamic LPAR 111, 126
  dynamic LPAR operations 127
  IVM 127, 145
  RMC 127
  SLES 109
  virtual SCSI 81
dynamic reconfiguration 9
dynamic resources 126
  considerations 128
  dynamic LPAR operations 127
  micro-partitions 127
dynamic routing protocols 252
dyntrk 249, 277

E

enhanced concurrent volume groups 341, 345
entitled capacity 6
Entitled Pool Capacity 309
entstat command 269
errorlog 286, 295
errpt command 295
ethchan_config command 275
EtherChannel 66, 90, 255
Etherchannel 272
Ethernet 32
  adapter 90
  EtherChannel 90
  link aggregation 90
  network interface backup 90
  TCP Segmentation Offload 144
  virtual Ethernet 92
  VLAN 27
Ethernet connection bonding 299
extendvg command 195

F

fast_fail 249, 277
fc_err_recov 249, 277
fget_config command 188, 278
firewall 93
firmware 27, 38, 128
fixpack 141

G

General Parallel Filesystem, GPFS 348
GNU Public License, GPL 118

H

HACMP 252, 341, 346
Hardware Management Console, HMC 67
  dynamic partitioning 81, 214, 327
  IVM 70
  virtual console 33
hcall 28, 32
hcheck_mode 247
help command 135

I

IBM Passport Advantage 115
IBM Tivoli Application Dependency Discovery Manager, TADDM 69, 146
IBM Tivoli Monitoring enhancements (ITM) 146
IBM Tivoli Monitoring System Edition, ITMSE 68
IBM Tivoli Monitoring, ITM 150
IBM Tivoli Usage and Accounting Management, ITUAM 69
IBM Tivoli Usage and Accounting Manager (ITUAM) 146
IBM TotalStorage Solutions, supported configuration 335
IEEE 802.1Q 87
IEEE 802.1Q compatible adapter 166
installios command 139–140, 173, 232
instruction retry recovery, IRR 126
Integrated Virtual Ethernet, IVE 130, 146
  features 130

logical Ethernet adapters 130
logical ports 131
overview 130
Integrated Virtualization Manager, IVM 10, 68, 192
APV feature 5
HMC 11
Integrated Virtual Ethernet 74
introduction 70
partition
configuration 73
physical adapters 74
processors 73
sample configuration 71
setup guidelines 70
virtual console 33
virtual Ethernet 73
Virtual I/O Server 70
virtual SCSI 74
VMC 72
inter-partition networking 83, 91
virtual Ethernet 91
interrupts 28
IOSCLI 134
IP Address Takeover, IPAT 252
IP filtering 93
IP fragmentation 96
IPSec 93

J

jumbo frames
virtual Ethernet 92
VLAN 86

K

kernel 2.6
Linux 104
kickstart 297

L

layer-2 bridge 94–95
license command 178
licensing
Linux 118
Lightweight Directory Access Protocol client (LDAP) 146
Link Aggregation 90, 96, 258
Cisco EtherChannel 259
EtherChannel 259
IEEE 802.3ad 259
LACP 259
Link Aggregation Control Protocol 259
NIB 272
of virtual Ethernet adapters 260
Linux 104
automated installation 297
considerations 109
continuous availabilty 299
device drivers 104
device mapper multipath 306
distributor 118
Ethernet connection bonding 299
GPL 118
installation 296
kernel 2.6 104
LVM 108
MPIO 105
RAID 107
software licensing 118
software RAID mirroring 303
VIO client 104
virtual console 104
virtual Ethernet 105
virtual I/O 103
virtual SCSI 105
Live Partition Mobility 11, 102, 147
active migration 102
hardware infrastructure 103
inactive migration 102
minimum requirements 148
Multiple Shared-Processor Pools 57
Virtual I/O Server 103
Logical Partitions, LPAR 6
micro-partitioning 6
Logical Remote Direct Memory Access, LRDMA 80
logical volume 81
Logical Volume Manager
Linux 108
lscfg command 191, 285
lsdev command 142, 179, 183, 191, 250, 267
lshwres command 316
lsmap command 183, 186, 281
lspath command 283, 286
lspv command 283, 295
lstctpip command 142
LVM mirroring 251
scenario 288

testing 294

M

MAC address 84, 95
maintenance 260
Maximum Pool Capacity 309
microcode 141
Micro-Partitioning
  entitled processor capacity 42
micro-partitions 6, 34
  APV feature 4
  capped 7, 36, 43, 127–128
  configuration attributes 35
  considerations 128
  dedicated memory 35
  entitled processor capacity 30, 42, 44
  firmware enablement 20
  I/O requirements 35
  introduction 27
  licensing
    capped 112
    entitlement capacity 112
    uncapped 113
  maximum 35
  mode 36
  overview 33
  physical shared-processor pool 6
  POWER5 34
  POWER6 34
  processor entitlement 33
  share of the unused capacity 36
  uncapped 7, 36, 127–128
    uncapped weight 36
  virtual console 27
  virtual SCSI 27
  VLAN 27
migration
  Virtual I/O Server 82
mirroring the Virtual I/O Server rootvg 214
mirrorios command 196
mirrorvg command 293
mkbdsp command 142
mklv command 184
mksysplan command 148
mktcpip command 182, 268
mkvdev command 180, 219, 266, 272
mkvg command 184
monitoring 32
MPIO 246
  Linux 105
  scenario 275
  SDD 337
  SDDPCM 250, 337
  testing 286
  third-party multipathing software 250
MTU 86
multicast 94
Multi-Path I/O
  Linux 105
multiple operating system support 11
Multiple Shared-Processing Pools, MSPPs 7
  APV feature 5
  architectural overview 8
  configuring 308
  default Shared-Processor Pool 8
  entitled capacity 7
  Maximum Pool Capacity 7
  physical shared-processor pool 7
  Reserved Pool Capacity 7
Multiple Shared-Processor Pools, MSPPs 20–21, 44
  architecture 45
  capacity redistribution 48
  capacity resolution 53
    Level0 53, 55
    Level1 53–54
  capped 62
  capping 61
  ceded processor capacity 48
  creating 51
  default Shared-Processor Pool 45, 49
  deleting 57
  dynamic adjustment 55
  dynamic movement 55
  Entitled Pool Capacity 46, 54
  examples 57
    database 60
    Web-facing 58
  HMC 51, 55
  Live Partition Mobility 57
  maximum 49
  Maximum Pool Capacity 46, 57
  micro-partition uncapped weights 54
  Physical Shared-Processor Pool 45
  Reserved Pool Capacity 46–47
  scope (Level1) 55
  set of micro-partitions 44

Shared-Processor Pool ID 46

N

NAT 86
NDP 94
network
adapter 90
interface backup 90
Network Interface Backup, NIB 255
advantages 257
errorlog 273
Etherchannel 272
scenario 270
testing 273
when to use 256
no_reserve 279

O

oem_setup_env command 188
Open Firmware prompt 298
operating system 27
AIX
boot image 120
monitoring 32
SMIT 122
reboot 120
Red Hat AS 118
SLES 118
operating system support
AIX 12
AIX 5L 12
Linux 13
optical device 33
IVM 74
OS_install command 229
OSI-Layer 2 318
OSPF 252

P

padmin
userid 135, 183
partition
boot 72
configuration
IVM 72–73
dedicated 127
definition 33
DLPAR 111, 123, 126
virtual SCSI 81, 214
dynamic
IVM 127
SLES 109
inter-partition networking 83, 91
licensing
capped 112
entitled capacity 112
uncapped 113
profile 42
partition deployment 222
Partition Migration 157
Partition Mobility 147
minimum requirements 148
Passport Advantage 115
performance
SMT 122
virtual SCSI 82
Performance Toolbox 68
physical adapters 127
physical devices 32
physical processor 29, 128
physical shared-processor pool 39–40
affinity scheduling 41
POWER5 40
physical shared-processor pools
POWER6 45
Ping test 213
Port Virtual LAN ID, PVID 88, 96, 98
POWER Hypervisor 5
abstraction layer 28
calls 32
dispatch cycle 59
dispatch mechanism 29
firmware 28
inter-partition networking 91
introduction 27
partition integrity 28
physical shared-processor pool 29–30, 41
processor affinity 31
processor dispatch 28
processor sparing 126
RDMA 80
virtual I/O adapter types 32
virtual I/O operations 32
virtual TTY console support 33
VLAN 88
POWER6 31

POWER6 Blade JS22 147
PPPoE 88
processing unit 35, 38, 126, 128
processor
affinity 31
assigned 111
de-allocation 126
dedicated 35, 40, 126
dispatching 29
physical 29, 34, 128
POWER5 27, 29, 31
POWER6 44, 126, 130
sparing 126
unassigned 111
unit 35, 38, 126
virtual 27, 29, 38, 42, 127–128
folding 39
licensing 113
protocol
TCP/IP 84
virtual SCSI 76

Q

Quality of Service, QoS 93

R

RAID 107, 333
RDAC 338
reboot 120
Red Hat AS
licensing 118
Redbooks Web site 356
Contact us xix
redundant Virtual I/O Servers 17
repphyvol command 147
reserve_policy 250, 340
Reserved Pool Capacity 309
restricted Korn shell 135
RMC 100, 127
rmdev command 184, 191
routing
virtual Ethernet 92, 97
rset command 39

S

save_base command 147
schedo command 39
SCSI mappings
defining 185, 189
SCSI Remote Direct Memory Access, RDMA 80
SDD 337
SDDPCM 337
SEA failover
control channel 252
delay 254
priorities 253
scenario 264
testing 268
when to use 256
security 18
certifications 18
separating traffic 262
server adapter
virtual SCSI 76
service processor
IVM 70
serviceability 16
Shared Dedicated Capacity 8, 21, 63
APV feature 4
Shared Ethernet Adapter 66, 93, 100, 221
advantages 257
bridging 92, 97
configuring for VLANs 325
considerations 329
creating a shared Ethernet adapter 179
external networks 100
failover 66
filtering 93
firewall 93
GARP VLAN Registration Protocol 66
inter-partition networking 99
introduction 92
IPSec 93
Link Aggregation 66
QoS 93
routing 92, 97
TCP segmentation offload 66
Shared Ethernet Adapter failover, SEA failover 252
Shared-Processor Pool 36, 308
Simple Network Management Protocol (SNMP)
146
simultaneous multithreading 6, 118
AIX 119
disabled 120
enabled 120
instruction cache 119

Linux 122
mode setting 120
POWER5 119
POWER6 119
single-threaded execution mode 119
SLES
licensing 118
smitty installios command 139
SMS boot mode 176
SMT 6
POWER5 6
POWER6 6
smtctl command 120–121
software
SSH 142
software licensing 23, 61, 109
Advanced POWER Virtualization 23
capped micro-partitions 112
CUoD 114–115
dedicated processor 112
factors 110
IBM software 116
licensing factors 111
licensing methods 109–110
Linux 118
Multiple Shared-Processor Pools 109, 113
on demand 115
Passport Advantage 115
planning 114–115
Red Hat AS 118
shared dedicated partitions 112
SLES 118
sub-capacity program 115
enrollment 116
sub-processor licensing 110
summary 113
uncapped micro-partition 113
software maintenance 23
software RAID mirroring 303
SSA support 81
stale partitions 245, 269
startnetsvc command 142
stopnetsvc command 142
storage pools 145
storage virtualization 10
su command 147
supported commands 135
switch
virtual Ethernet 91
SWMA 23
syncvg command 295
System p servers 16
System Plan 148
Viewer 149
System Planning Tool, SPT 67, 148, 222

T

tagged packets 89
tagged port 88
target 76, 80
target device 185
TCP Segmentation Offload 144
tee command 135
third-party multipathing software 250
thread 128
Tivoli Enterprise Management Server 68
Tivoli Enterprise Portal 68
Tivoli Identity Manager, TIM 69, 151
Tivoli Storage Manager, TSM 69, 146
Tivoli System Automation, TSA 252
topas command 142
trunk flag 95, 252
trunk priority 167
TTY 33
IVM 74

U

unassigned processor 111
uncapped 36, 113, 127–128
weight 36, 128
untagged packets 88
untagged port 89
unused processor capacity 36
updating the Virtual I/O Server using fix packs 152

V

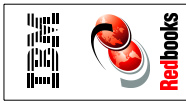
value of virtualization 2
varyonvg command 245, 295
viosecure command 142
viostat command 142
VIPA 252
virtual console 27, 32
IVM 74
Linux 104
virtual Ethernet 9
AIX adapters and interfaces 90

ARP 94
bridging 92, 97
broadcast 94
configuration 97
Etherchannel 96
filtering 93
firewall 93
flowchart 319
HACMP 343
IEEE VLAN header 318
IEEE802.1Q 88
inter-partition networking 91, 98
introduction 83, 269
IPSec 93
IVM 72–73
jumbo frames 92
layer-2 bridge 95
link aggregation 96
Linux 105
multicast 94
NDP 94
POWER Hypervisor switch implementation 318
PVID 96, 98
QoS 93
routing 92, 97
switch 91
trunk adapter 318
trunk flag 95
VLAN 98
virtual Ethernet adapter
access external network flag 252
creating a virtual Ethernet adapter for the VIOS 166
maximum number 329
maximum number of VLANs 329
trunk flag 252
virtual I/O
client 103
Linux 103
considerations 109
server 103
virtual I/O adapters 32
Ethernet 32
SCSI 32
TTY 33
virtual console 32
Virtual I/O Server 10, 32, 63, 141–142, 146–147
Advanced POWER Virtualization Enterprise Edition 63
Advanced POWER Virtualization Standard Edition 63
APV feature 5
command line interface 67
disk presentation 76
functions 64
HMC integration 67
installation 82
installation and migration 82
IVM 71
logical volume 81
maintenance 260
mirroring 333
MPIO 335
network security 66
ordering 21
Performance Toolbox support 68
rebooting 261
RMC 100
separating traffic 262
Shared Ethernet Adapter 66
Shared Ethernet Adapters 10
software installation 173
storage pools 145
storage virtualization 64
supported platforms 64
system maintenance 260
System Planning Tool support 67
Tivoli support 68
upgrading 261
using multiple 244, 339
using single 336
version 1.5 64
virtual SCSI devices 65
virtual LAN, VLAN 27, 84, 98
AIX support 86
benefits 86
IEEE 802.1Q 87
overview 84
POWER Hypervisor 88
PPPoE 88
PVID 88
tagged port 88
untagged port 89
Virtual Management Channel, VMC 72
virtual processor 29, 42, 127–129
licensing 113
reasonable settings 129
virtual processor folding 39

default mode 39
vpm_xvcpus 39
virtual SCSI 9, 27
client adapter 76
client partition 77
client/server architecture 76
configuration 76
considerations 81
devices 65, 76
dynamic partitioning 81, 214
failover 143
growing disks 143
HACMP 341
initiator 76, 80
introduction 75
IVM 72, 74
Linux 105
logical volume 81
mirrored devices 331
multiple backing devices 78
new features in VIO 1.3 143
optical device 74
performance considerations 82
protocol 76
queue depth 143
redundancy 245
server adapter 76–77
SSA support 81
target 76, 80
transport protocols 80
virtual SCSI Server adapter 219
virtual TTY 33
VLAN 317
BootP 86
DHCP 86
DLPAR 327
extending into client partition 327
HMC 327
MAC address 84
MTU 86
NAT 86
scenario 320
VNC 298

W

wkldagent command 142
wkldout command 142
wklmgr command 142



PowerVM Virtualization on IBM System p: Introduction and Configuration

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

PowerVM Virtualization on IBM System p:
Introduction and Configuration
Fourth Edition



Basic and advanced configuration of the Virtual I/O Server and its clients

Updated to include new POWER6 technologies

The next generation of Advanced POWER Virtualization

This IBM Redbooks publication provides an introduction to PowerVM virtualization technologies on IBM System p servers. The Advanced POWER Virtualization features and partitioning and virtualization capabilities of IBM Systems based on the Power Architecture have been renamed to PowerVM.
PowerVM is a combination of hardware, firmware and software that provides CPU, network and disk virtualization. The main virtualization technologies are:

- POWER6 and POWER5 hardware
- POWER Hypervisor
- Virtual I/O Server

Though the PowerVM brand includes partitioning, software Linux emulation, management software, and other offerings, this publication focuses on the virtualization technologies that are part of the PowerVM Standard and Enterprise Editions.
This publication is also designed to be an introduction guide for system administrators, providing instructions for:

- Configuration and creation of partitions and resources on the HMC
- Installation and configuration of the Virtual I/O Server
- Creation and installation of virtualized partitions

While discussion is focussed on IBM System p hardware and AIX, the basic concepts can be extended to the i5/OS and Linux operating systems as well as the IBM System i hardware.
This edition has been updated with the new features available with the IBM POWER6 hardware and firmware.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

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

For more information:
ibm.com/redbooks

SG24-7940-03 ISBN 0738485306