An Introduction to the New IBM eserver pSeries High Performance Switch

Installation and configuration of the IBM eserver HPS

Considerations for configuring applications for the IBM HPS

Experiences with VSD, GPFS and HACMP running over HPS

Octavian Lascu
Zbigniew Borgosz
Josh-Daniel S. Davis
Pablo Pereira
Andrei Socoliuc

Redbooks

ibm.com/redbooks

International Technical Support Organization

An Introduction to the New IBM eserver pSeries High Performance Switch

December 2003

SG24-6978-00

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

First Edition (December 2003)

This edition applies to Version 5, Release 2, Modification 2 of AIX 5L (product number 5765-E62), and Version 1, Release 3, Modification 2 of Cluster Systems Management (CSM) for AIX.

Contents

Notices

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

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

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

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

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

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

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

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

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

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

Trademarks

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

eserver ™
eserver™
eServer™
Redbooks(logo) ™
ibm.com®
iSeries™
pSeries®
xSeries®
AIX 5L™
AIX/ESA®
AIX®
BladeCenter™
DB2®
Electronic Service Agent™
Enterprise Storage Server®
ES/9000®
ESCON®
IntelliStation®
IBM®
LoadLeveler®
PowerPC®
POWER3™
POWER4™
POWER4+™
Redbooks™
RS/6000®
SP1®
SP2®
Tivoli®

The following terms are trademarks of other companies:

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

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

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

Preface

This IBM® Redbook contains information about the first official release of the pSeries® High Performance Switch (HPS) and products that may benefit from this equipment. This book includes detailed information about the hardware and software configuration as well as examples of supported and potential configurations.

This redbook will help you install, tailor, and configure the new switch on IBM's premier UNIX® operating system, Advanced Interactive eXecutive (AIX®) 5.2, and shows the differences from earlier generations of similar technology previously known collectively as the Scalable POWERparallel (SP) Switch. Additionally covered are topics such as Global Parallel File System (GPFS) and High Availability/Cluster Multi-Processing (HACMP).

This redbook gives a broad understanding of this new architecture and its dependencies on the pSeries Hardware Management Console (HMC) and RISC System Cluster Technology (RSCT).

This redbook will help you design and create a solution to exploit the power and performance of the initial release of this new switch and to plan for future generations.

The team that wrote this redbook

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

The team: left to right, standing: Octavian Lascu (Project Leader), Pablo Pereira, Zbigniew Borgosz; sitting: Josh-Daniel Davis, Andrei Socoliuc

Octavian Lascu is a Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of pSeries clusters and Linux. Before joining the ITSO two years ago, Octavian worked in IBM Global Services, Romania as SW and HW Services Manager. He holds a Master's Degree in Electronic Engineering from Polytechnical Institute in Bucharest and is also an IBM Certified Advanced Technical Expert in AIX, PSSP and HACMP. He has worked with IBM since 1992.

Zbigniew Borgosz is a senior systems consultant working for ComputerLand S.A, the IBM Business Partner in Poland. He joined the company in 1998. He has five years of experience in UNIX systems. His areas of expertise include designing and implementing highly available and scalable solutions based on pSeries, SUN Microsystems and Hewlett-Packard hardware and software. He provides technical support to ComputerLand staff as well as to customers.

Pablo Pereira is an IT Specialist for IBM Global Services in Uruguay serving government and telecommunications customers, responsible for design and implementation of complex solutions. He has five years of experience in UNIX and pSeries field. His areas of expertise include AIX, PSSP, HACMP, and TCP/IP.

Josh-Daniel S. Davis is a Staff Software Engineer for IBM Global Services in Dallas, Texas. He has nine years of Information Technology experience and has been with IBM for over five years. His areas of expertise include AIX, Linux and pSeries eServers He is certified for

Tivoli® Storage Manager and is a Certified Advanced Technical Expert for AIX 4.3 and 5L including PSSP and p690 support.

Andrei Socoliuc is a Software Support Engineer in IBM Global Services in Romania. He holds a Master's degree in Computer Science from Polytechnic Institute in Bucharest, Romania. He has six years of experience in the pSeries Clusters field. His areas of expertise include AIX, PSSP, HACMP, TSM, and Linux. He has written extensively on pSeries Clusters managed by PSSP.

Special thanks to **Fernando Pizzano**, IBM Poughkeepsie for his extensive contribution to this project.

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

Marc Gendy
NCAR, Boulder, Colorado

Paul Moyer
IBM Poughkeepsie

Karyn Corneli
IBM Dallas

Nick Rash
IBM Poughkeepsie

Dino Quintero
International Technical Support Organization, Poughkeepsie Center

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 008 Internal Zip MS P/099
  2455 South Road
  Poughkeepsie, New York 12601-5400

Part 1

Technical information

In this part we introduce the IBM @server pSeries High Performance Switch (HPS), provide technical details, and provide further information regarding applications that may benefit from the characteristics of the HPS.

1

Introduction

This introduction is split into two sections:

- In 1.1, “Historical background” on page 4, we provide an historical background on the predecessors of the IBM @server pSeries High Performance Switch (HPS). In it we briefly touch on topics such as supercomputing, the IBM SP, and the Cluster 1600.
- In 1.2, “The pSeries High Performance Switch” on page 9, we continue with the technology that forms the basis of this new switch, its major components, and most significant features. Whether the reader is highly technical or not, this information should assist in an understanding of the capabilities of the pSeries HPS. For greater technical details, see Chapter 2, “Technical overview of the IBM Eserver pSeries High Performance Switch (HPS)” on page 15.

1.1 Historical background

The following section discusses the technological roots of the IBM eserver pSeries High Performance Switch (HPS). Additional historical information is available in other books, including *Inside the RS/6000 SP*, SG24-5145.

1.1.1 HPSSDL

In the late 1980s, IBM intended to build a supercomputer for large technical customers. The High Performance Supercomputer Systems Development Laboratory (HPSSDL) was formed within the IBM Large Systems Division in Kingston and Poughkeepsie, New York. HPSSDL intended to create a supercomputer based on the familiar technology of the IBM ES/9000® vector processing mainframe, depicted in Figure 1-1.

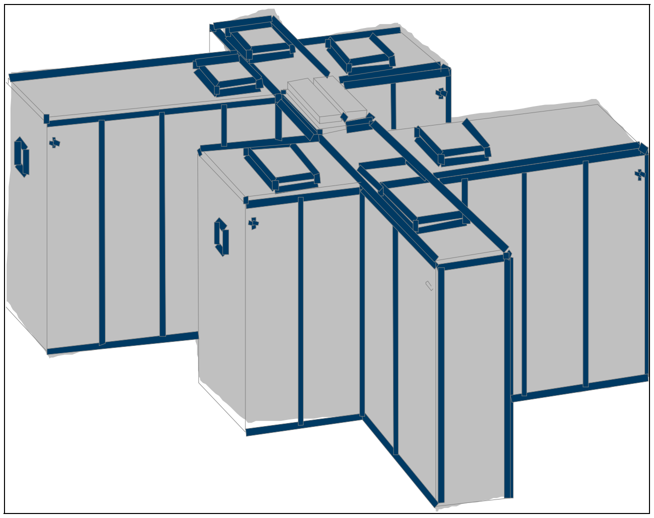


Figure 1-1 Example ES/9000 frame layout

1.1.2 RS/6000®

In 1990, the IBM Advanced Workstation Division in Austin, Texas introduced the RS/6000 (RISC System/6000) family of UNIX-based workstations and servers, such as the one shown in Figure 1-2. Based on Performance Optimization with Enhanced RISC (POWER) CPU architecture, HPSSDL became very interested in these early RS/6000 machines due to high performance floating-point operations and the fact that they ran UNIX, which was popular with large scientific and technical customers.

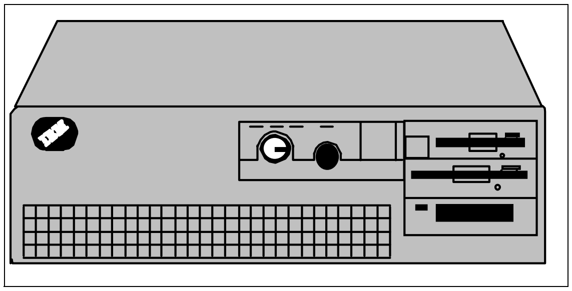


Figure 1-2 RS/6000 7012-3xx Series

At a crossroad with mainframe technology, HPSSDL experimented with off-the-shelf RS/6000 machines interconnected by ESCON® Channel adapters and an ESCON Director. The RS/6000 machines were repackaged as nodes and mounted in drawers, which were then mounted with five drawers to a frame.

1.1.3 SP1® and early switch implementation

IBM in Yorktown, New York was working on a high-speed switch (6 MBps bandwidth and 200 ms latency) while another group on-site developed an eight-drawer frame and the associated management software. In December 1991, these groups came together as HPSSL (the “Development” part of the name was dropped) and were charged with shipping a product within 12 months. Standard RS/6000 workstations were adapted, along with a new version of the switch developed in Yorktown in place of ESCON for reduced latency. See Figure 1-3.

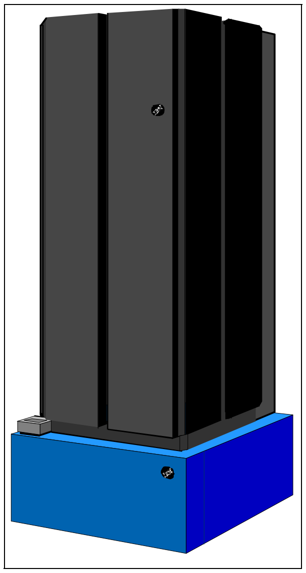


Figure 1-3 SP frame

The total product was introduced to the marketplace as the SP1 in September of 1993. By year’s end, 72 systems had been installed around the world in the scientific and technical community. As the mainframe began losing popularity, commercial customers also began calling on IBM. IBM formed an application solutions group for the SP1, which, among other things, ported a parallel version of Oracle’s database to the SP1.

1.1.4 PSSP and SP2®

In 1994, SP development absorbed personnel from the discontinued AIX/ESA® product. They bolstered the manageability of the system and helped spawn the Parallel System Support Programs (PSSP) software and the SP2 was born.

The Yorktown-developed switch gave way to the High Performance Switch (HiPS), running at 48 MBps bandwidth and 30 ms latency.

The SP2 moved out from under the umbrella of the Large Systems Division to become its own enterprise within IBM. SP2 sales were strong: 352 systems were installed by the end of 1994 and 1,023 by the end of 1995.

1.1.5 The SP Switch

As with most things in the computing industry, SP2 performance continued to grow. In 1996, the SP2 was renamed to simply SP and formally became a product of the RS/6000 Division representing the high-end of the RS/6000 family. SMP nodes were introduced to meet processing demand. The HiPS gave way to the SP Switch, which, by 1999, provided 180 MBps bandwidth and 21 ms latency. See Figure 1-4.

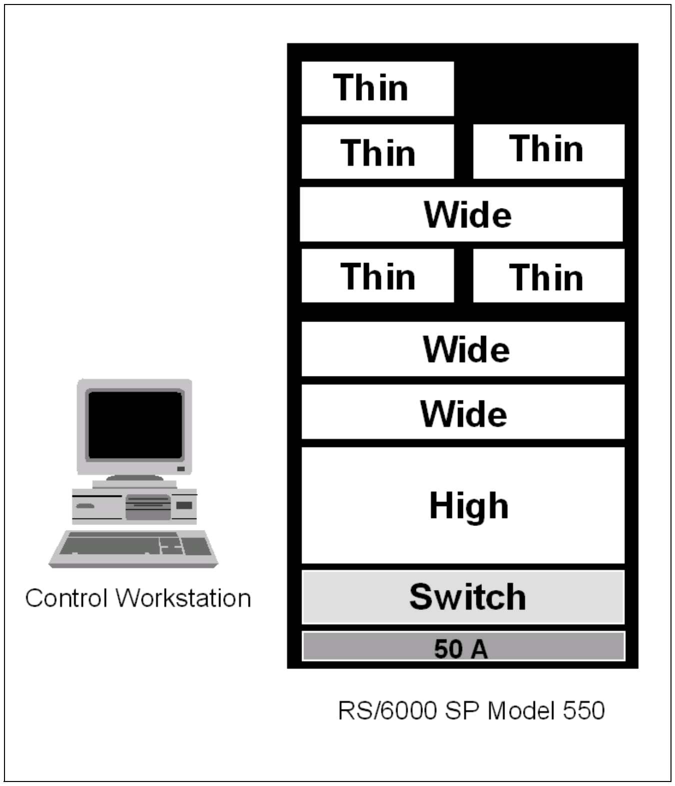


Figure 1-4 SP frame and switch logical view

1.1.6 PCI and Enterprise Servers

In 1998, PCI nodes based on PowerPC® CPUs, known as Silver Nodes, were introduced. To meet the needs of demanding workloads, additional machines were added to PSSP clusters, including the 7017-S70 and S7A, as shown in Figure 1-5 on page 7.

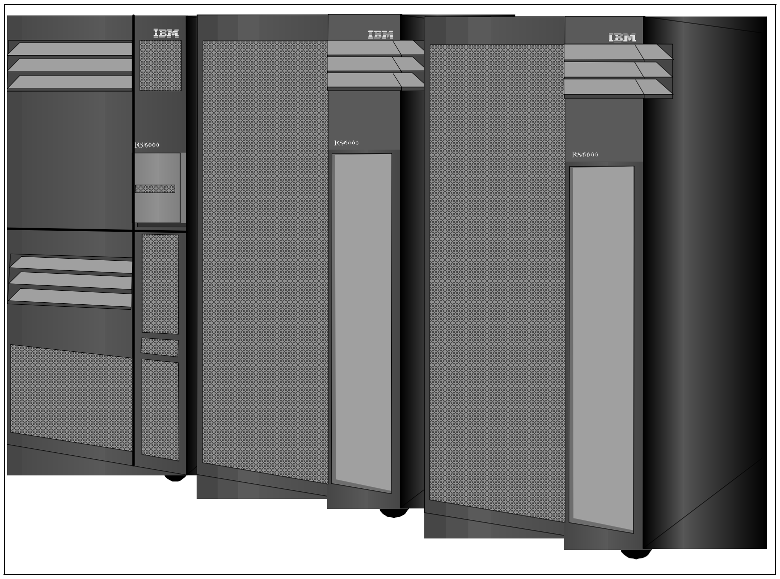


Figure 1-5 7017-S70 and expansion racks

In 1999, Winterhawk and Nighthawk nodes were introduced, bringing POWER3™ architecture SMP nodes and a greater performance I/O subsystem and providing support for Enterprise Server Model 7017-S80.

In the year 2000, Silicon On Insulator and copper interconnect technologies were integrated into the POWER3-II CPU and released in Winterhawk-II and Nighthawk-II nodes. In July of 2000, PSSP 3.2 was introduced and brought about support for Clustered Enterprise Servers. This new version of PSSP made it possible to cluster the 7017 enterprise servers without an SP frame.

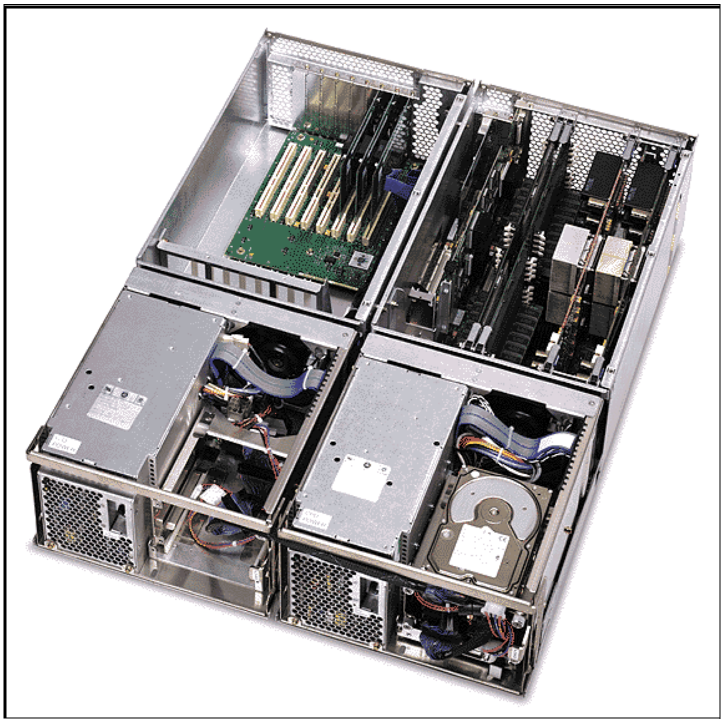


Figure 1-6 375 MHz wide node (Winterhawk II)

1.1.7 The SP Switch 2

To keep up with the higher I/O demand, IBM also released the SP Switch 2. Several attachment buses were supported and two adapters were allowed in the same system for redundancy, an option known as Dual Plane. Under ideal configurations, the SP Switch 2 was able to bring up to 700 MBps bandwidth with 17 ms latency.

1.1.8 pSeries and Cluster 1600

As IBM brand names changed, RS/6000 became eServer pSeries, bringing about a new range of SP Attached Servers. By 2001, IBM extended this tradition to the midrange servers RS/6000 H80, M80, IBM @server pSeries Models p660 (6H1), p680 (S85), and p610.

Furthermore, SP hardware became a function of Cluster 1600 (renaming of Clustered Enterprise Server - CES), shifting the focus of pSeries clusters.

1.1.9 Regatta

Continuing the trend of AIX 5L™ to merge functionality from all of IBM's product lines, the p690 (7040-681) was born as the first IBM LPAR-capable UNIX system. To serve smaller needs, the p670 (7040-671) was released, being identical to the p690 but with support for fewer CPU modules, and therefore fewer L3 cache modules, memory books, and GX buses. See Figure 1-7.

These machines were quickly brought into Cluster 1600 and have been followed by smaller LPAR-capable machines, such as the p655, p650, and p630.

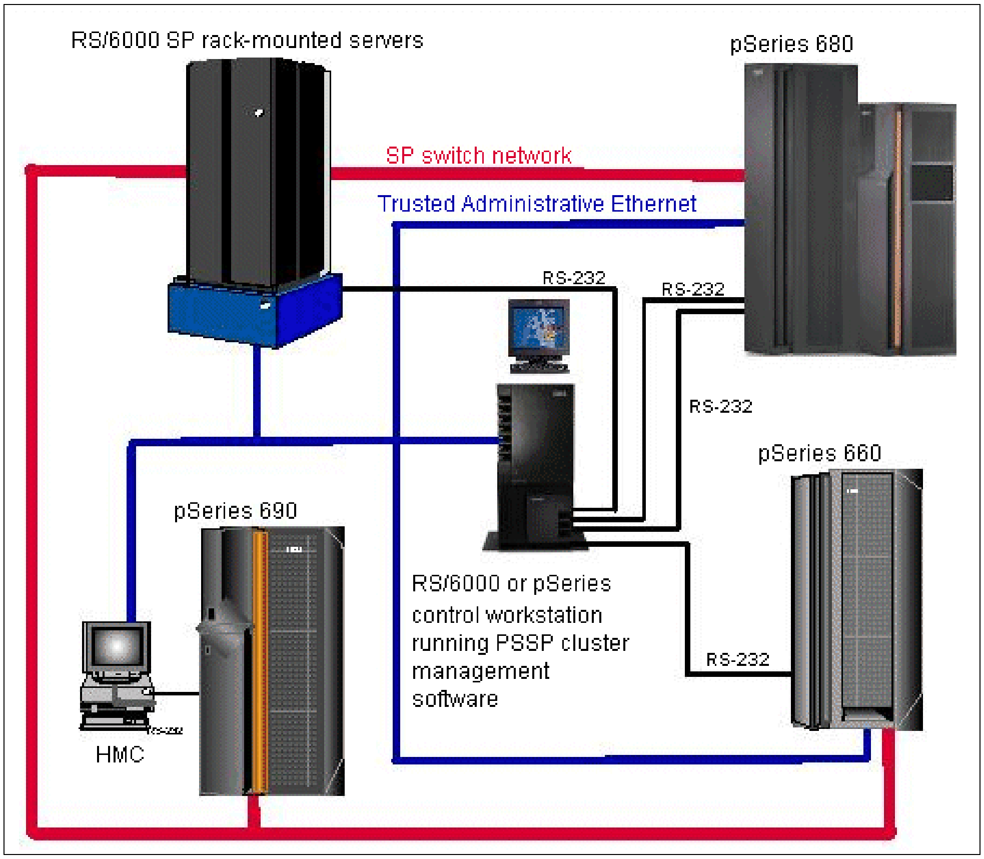

RS/6000 SP rack-mounted servers
pSeries 680
SP switch network
Trusted Administrative Ethernet
RS-232
RS-232
RS-232
pSeries 660
pSeries 690
RS/6000 or pSeries control workstation running PSSP cluster management software
RS-232
RS-232
HMC

Figure 1-7 Cluster 1600 with PSSP and p690

1.1.10 Cluster Systems Manager

As it became obvious that SP hardware could not keep up with the pSeries line and that PSSP lacked Linux Affinity, a major selling point of AIX 5L, Cluster Systems Manager (CSM) was spawned from the IBM AlphaWorks lab and packaged with AIX 5L. CSM has progressed to include support for legacy SP hardware, pSeries, and xSeries® (running Linux) as shown in Figure 1-8.

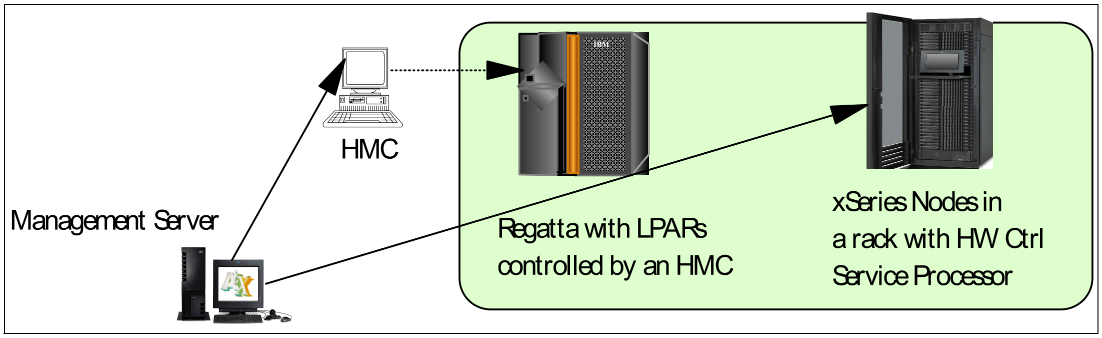


Figure 1-8 CSM Cluster 1600 example

Because of the tight integration of the SP Switch into PSSP, CSM did not include support for the SP Switch. In order to ensure the viability of CSM for Cluster 1600, a new switch was developed.

1.2 The pSeries High Performance Switch

The IBM @server pSeries High Performance Switch (HPS) is IBM's answer for high performance networking outside of an SP complex. As you can see from Figure 1-9 on page 10, the logical design of the HPS is very similar to that of the SP Switch.

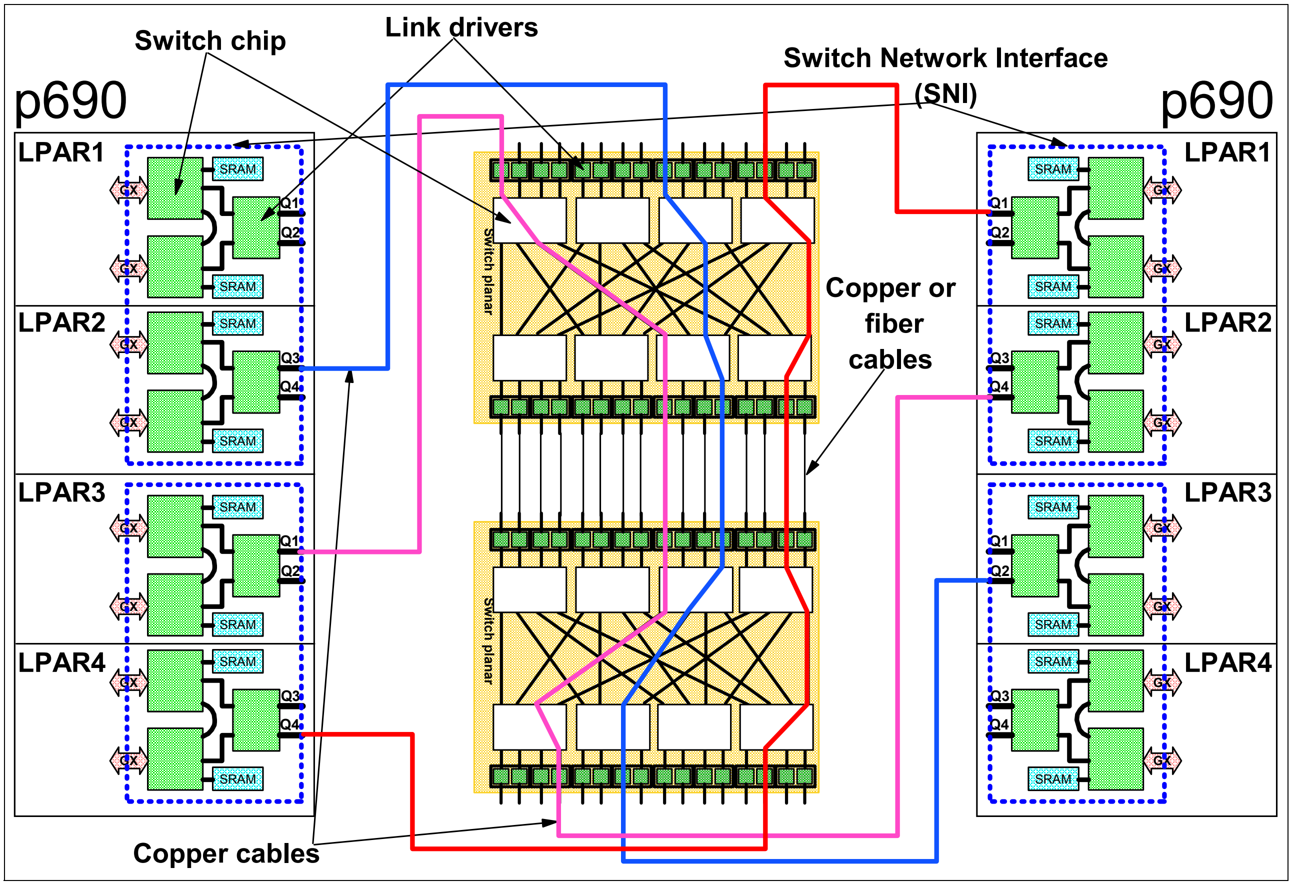


Figure 1-9 Switch hardware diagram

While building on the technology of the SP Switch 2, the IBM @server pSeries High Performance Switch (HPS) originated as a NUMA Shared Memory Adapter (SMA). Because of this, NUMA requirements dictate its functions. NUMA requires a shared memory segment between two systems so that they may coordinate and operate as one. While IBM has not announced any pSeries NUMA offering, the technological foundations remain, thereby providing a low latency, high speed, shared memory network.

Since NUMA requires its network to be active prior to the operating system, Communication Subsystem Support (CSS) lacked the features required by the SMA. The only code available in the design of pSeries servers that was capable of driving such an adapter was the Hypervisor. Hypervisor is the portion of system microcode that provides Runtime Abstraction Services (RTAS) and Partition Management in LPAR mode. Currently the 7039 and 7040 machine types possess both Hypervisor and sufficient I/O resources to drive the Shared Memory Adapters.

1.2.1 Naming the technology

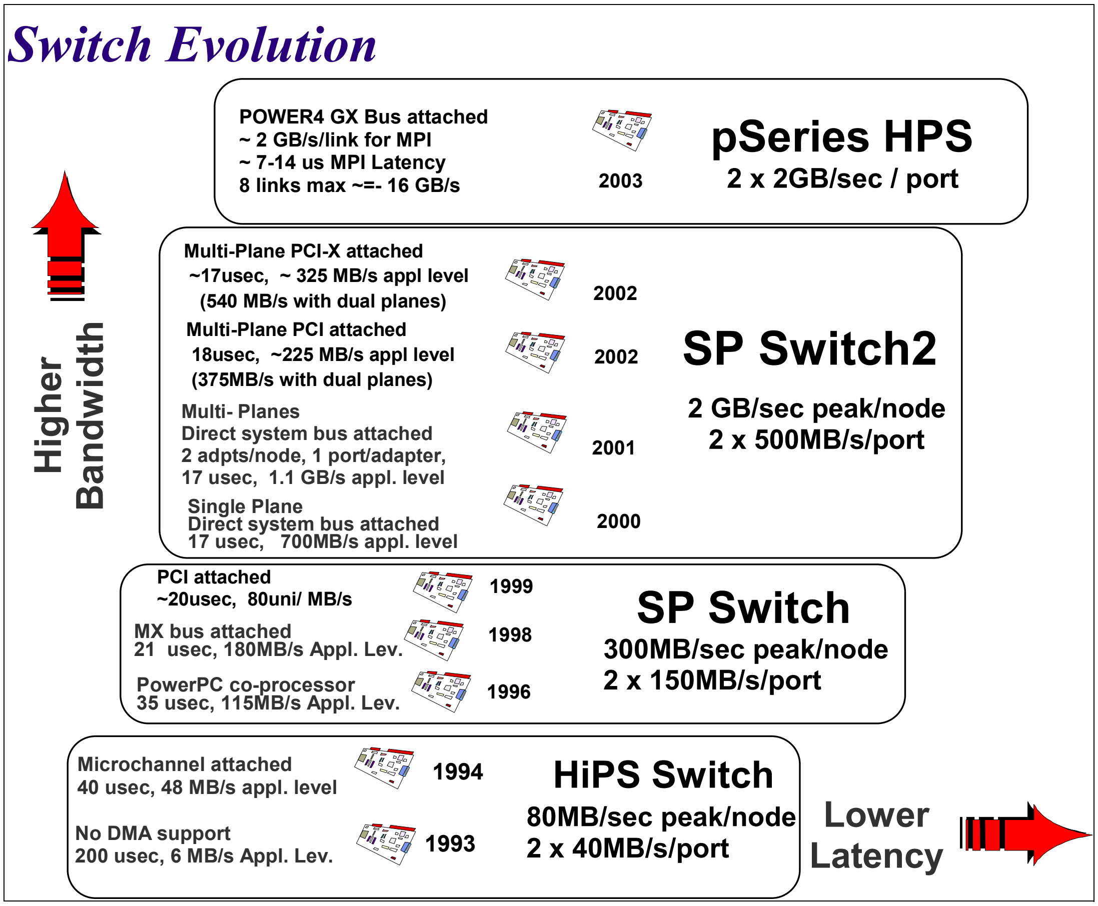
Because of the rich history of the HPS, there are many names for its different components, most commonly including those in Table 1-1 on page 11. See the glossary for more terminology.

Table 1-1 HPS common terms

FNM	See fnmd or SNM.
fnmd	This is the daemon that manages topology, master selection, and initialization of the switch network. This daemon performs its functions through hrdw_svr and presently runs on the HMC.
HPS	Standing for High Performance Switch, HPS is the shortest form of the official name “IBM @server pSeries High Performance Switch”.
HSC	Hardware Service Console. Also known as Hardware Management Console.
pSeries HPS	See HPS.
SMA	Shared Memory Adapter, an obsolete name based on its technical function, can still be found in some reference materials.
SNI	Switch Network Interface is the current name for server-side components.
SNM	Switch Network Manager, used in the GUI to reference functions of the FNM.

1.2.2 Progression of switch technology

The IBM @server pSeries High Performance Switch (HPS) is considered to be the next progression of the SP Switch. While the HPS cannot be used with PSSP or an SP complex, its internal design is very similar. See Figure 1-10 on page 12 for the logical progression of IBM switch technology.



Switch Evolution

POWER4 GX Bus attached
~ 2 GB/s/link for MPI
~ 7-14 us MPI Latency
8 links max ~=- 16 GB/s
2003
pSeries HPS
2 x 2GB/sec / port

Multi-Plane PCI-X attached
~17usec, ~ 325 MB/s appl level
(540 MB/s with dual planes)
2002
Multi-Plane PCI attached
18usec, ~225 MB/s appl level
(375MB/s with dual planes)
2002
Multi- Planes
Direct system bus attached
2 adpts/node, 1 port/adapter,
17 usec, 1.1 GB/s appl. level
2001
Single Plane
Direct system bus attached
17 usec, 700MB/s appl. level
2000
SP Switch2
2 GB/sec peak/node
2 x 500MB/s/port

PCI attached
~20usec, 80uni/ MB/s
1999
MX bus attached
21 usec, 180MB/s Appl. Lev.
1998
PowerPC co-processor
35 usec, 115MB/s Appl. Lev.
1996
SP Switch
300MB/sec peak/node
2 x 150MB/s/port

Microchannel attached
40 usec, 48 MB/s appl. level
1994
No DMA support
200 usec, 6 MB/s Appl. Lev.
1993
HiPS Switch
80MB/sec peak/node
2 x 40MB/s/port

Higher Bandwidth

Lower Latency

Figure 1-10 Switch evolution

1.2.3 Major HPS components

The HPS is based on several major components such as described in Table 1-2. Familiarity with these components will greatly assist you in understanding the requirements and operation of a pSeries High Performance Switch. They will be covered in much greater detail in Chapter 2, “Technical overview of the IBM Eserver pSeries High Performance Switch (HPS)” on page 15.

Table 1-2 HPS components

Component	Purpose
Trusted Network	Used to maintain privacy of administrative commands between the CSM Master and the HMC.
Switch	The HPS is a 4U drawer with 16 slots, which can hold up to 32 switch port s (two ports per card adapter), and can be placed in a single or dual frame switch-only rack, or in the bottom position of an existing p690 or p655 rack.
Switch Network Interface (SNI)	This is the card that plugs into a supported CEC, currently p690 and p655 systems, and that provides one or two pairs of network links. Configuration is provided via the HMC serial network and via AIX device drivers.

Component	Purpose
Hypervisor	This is the portion of pSeries microcode allowing logical partitions and is required when using the HPS SNI adapters.
Frame	This is the equipment that will contain either the pSeries server or the HPS, or both, and will often refer to the power subsystem as well.
Switch Power Control Network (SPCN)	Functionally, this is both the supervisor and distribution of the power system within a frame. All switch commands are passed here.
Central Electronics Complex (CEC)	This is the core of a pSeries server and refers to the unit that contains CPUs, memory, firmware, and other components.
Switch Network Manager (SNM or FNM)	This is a set of programs and an administrative interface that has been added to the pSeries HMC. It is the enabling software for an HPS.
Hardware Management Console (HMC)	This is the Linux-based IBM PC system used to administer LPAR-capable pSeries machines. This also provides access to Switch Network Management functions.
Cables	There are many new cables used in an HPS environment including high-density copper link cables, LC fiber link cables, and RS422 serial cables for access to the Switch Power Control Network.
Switch Port Connector Cards (SPCCs or Risers)	These are basically the line drivers for switch link cables that are installed into the face of a pSeries High Performance Switch. These allow the same switch to be used with both fiber and copper link cables.

1.2.4 IBM @server pSeries High Performance Switch (HPS) features

Great reliability

Unlike the SP Switch, errors are retried at each point within the network making this version of the switch a more reliable transport. Additionally, the pSeries HPS continues the tradition of dual-plane support.

Highly scalable

The HPS is cascadable up to 13 switch chips. What this means is that you can start small with one switch, and grow as you need up to 96 switches, or up to 1024 connections, all joined as one large network.

Reliable and secure

Routing changes are performed through out-of-band (serial) connections. This provides enhanced security, since the switch routing table (also known as the WORM table) cannot be modified from inside the network. Also modifications to the routes do not require a node connected to the switch to be up and running. Should you choose to use Cluster 1600 administration features, the recommended configuration puts sensitive administrative data outside of the reach of your LPARs.

Out-of-band monitoring

Each switch chip maintains its own link and chip performance and error statistics and is monitored externally to the switch network. This enables the Switch Network Manager to show network details with low performance impact.

Low latency

Each adapter is populated with routes from Hypervisor, allowing the use of source routing at the interface level to help keep latency low. Broadcast functions were dropped to reduce network contention. However, ARP broadcast is still provided transparently via multicast.

Each physical link can provide up to eight logical channels, which allows for prioritization of data. This can be used to prevent individual network sessions from using all of the bandwidth. Each logical channel has its own buffer, plus access to a central buffer and direct access to outbound ports to reduce latency caused by blocked packets.

Even with all of these functions and enhancements, latency has been reduced to a variable rate of between 7 and 14 microseconds, with potential for further reduction.

Fast transfers

The pSeries HPS has a raw bandwidth of 2 GBps per link per direction. Direct memory copy applications under optimal conditions are required to achieve maximum bandwidth. While the scientific community will benefit most from this, regular TCP/IP applications will also see a marked improvement over other currently available technologies.

2

Technical overview of the IBM @server pSeries High Performance Switch (HPS)

In the never-ending quest for speed, IBM has produced a newer, higher-speed version of the Shared Memory Adapter previously known as the SP Switch 2 adapter. With a new operating structure and new requirements, even the most experienced SP administrator will need a compare-and-contrast document to get started.

If you have not already done so, please familiarize yourself with the concepts presented in 1.2, “The pSeries High Performance Switch” on page 9 prior to continuing with this chapter. For those who are in a hurry, a quick summary is that the HPS is derived from SP Switch technology and designed to be a NUMA adapter. NUMA was abandoned for POWER4™. However, the constraints of that architecture moved many of the administrative components into firmware.

This chapter organizes and presents the many facets of technology behind the IBM @server pSeries High Performance Switch (HPS). Where possible, we have included both an overview and detailed practical information.

2.1 Hardware components

In addition to the seemingly standard switch network components shown in Figure 2-1, the IBM @server pSeries High Performance Switch (HPS) relies on a wide variety of both software and hardware components. These vary greatly from the structure of the Communication Support Subsystem (CSS) of the SP.

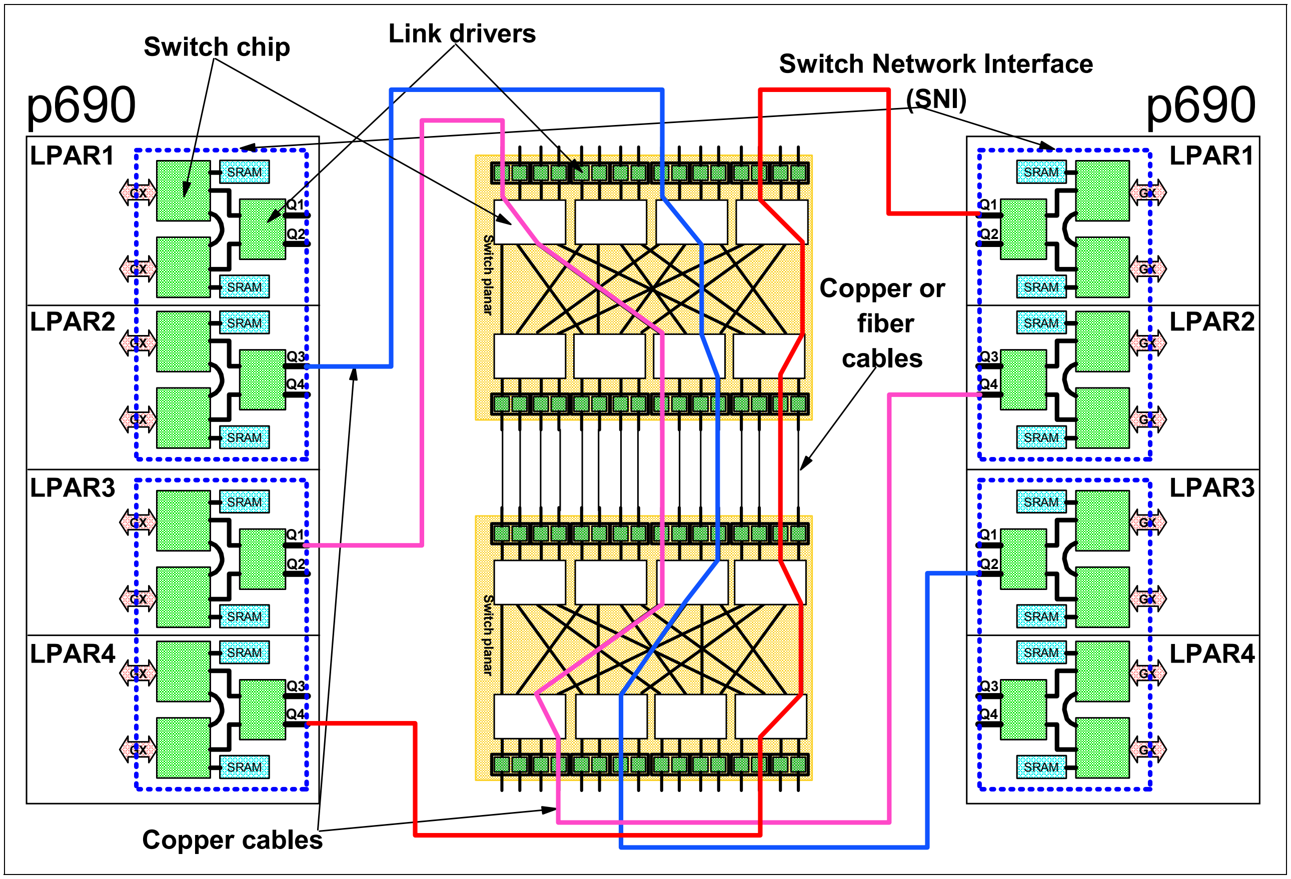


Figure 2-1 Switch hardware diagram

In the above diagram, the general structure appears to match that of an SP Switch except for one seemingly minor difference: we list a GX bus rather than MX or PCI. Because of the heavy requirements of the new HPS, only systems with two available GX buses will be able to take advantage of the performance it can give.

In addition to the systems, the pSeries HPS requires a Switch Network Interface (SNI), which is basically a multi-port network card. These ports are connected via copper cables to the switch assembly. The switch is attached to the frame via Universal Power Interface Connection (UPIC) cables. The frame obtains new cable connections to increase the performance of firmware operations and administrative commands to the switch.

All of these components and more are discussed further in this chapter.

2.1.1 Systems

During the initial offering, both the pSeries 690 and pSeries 655 machines are supported for attachment to the IBM @server pSeries High Performance Switch (HPS). See Figure 2-2 on page 17. These systems provide both mid-sized and enterprise server solutions with the ability to support GX attachment for high-performance peripherals such as the pSeries HPS.

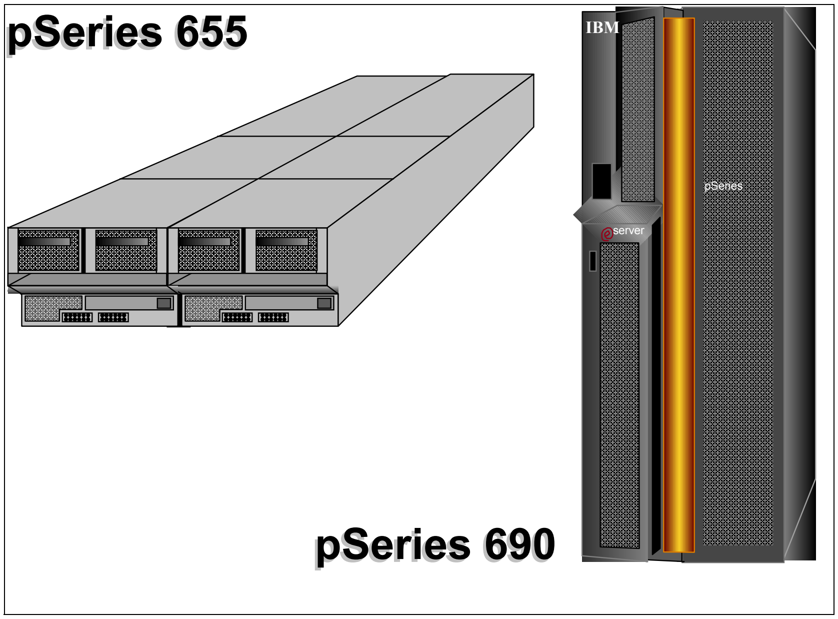


Figure 2-2 IIBM @server pSeries HPS capable machines

7039-651 Model p655

Each mid-level p655 server added to an HPS network brings with it a single four-way or eight-way MCM and up to 32 GB of primary storage. In addition, external Remote I/O (RIO) may be used to add a variety of peripherals beyond the integrated PCI-X and SP-attachment slots. The following are p655 CPU and RAM options:

- Four-way 1.3 GHz POWER4 SMP (one four-way MCM)
- Four-way 1.7 GHz POWER4+ SMP (one four-way MCM)
- Eight-way 1.1 GHz POWER4 SMP (one eight-way MCM)
- Eight-way 1.5 GHz POWER4+ SMP (one eight-way MCM)
- 4 to 32 GB or RAM

The p655 servers are fit two to a drawer and up to eight drawers per rack. Such a high-density configuration offers a total of 128 CPUs and 16 LPARs per rack to the HPS network. While the p655 server does support more than one LPAR, it only has bus resources to attach one LPAR to the HPS per p655 processor subsystem (CEC).

7040-681 Model p690

For customers who can afford the space on their raised floor or who otherwise need the additional processing power, the p690 can be attached to an HPS. Each p690 rack supports only one processor subsystem. However, each processor subsystem supports up to 32 processors at up to 1.7 GHz in clock speed (currently). The following are p690 CPU and RAM options:

- 32-way 1.1 GHz or 1.3 GHz POWER4 SMP (four eight-way MCMs)
- 16-way 1.1 GHz or 1.3 GHz POWER4 HPC/CUoD[1] (four four-way MCMs)
- 32-way 1.5 GHz or 1.7 GHz POWER4+ SMP (four eight-way MCMs)
- 16-way 1.5 GHz or 1.7 GHz POWER4+ HPC/CUoD[1] (four four-way MCMs)

[1] High Performance Computing is available as Capacity Upgrade on Demand. By disabling half of the CPUs, the L3 cache is shared by fewer number of CPUs, the Fabric Distributed Controller is less saturated, and the spare CPUs are available to take over, should a primary CPU fail.

- 8 to 512 GB of RAM

While each p690 supports up to 32 LPARs (with HMC 3.2 or higher and RIO-2), only four LPARs per processor subsystem can be attached to the HPS. In general, customers needing the performance of the HPS will find this acceptable. In fact, many of the preliminary customers made use of the aggregation abilities of the HPS by using all eight HPS connections in one p690 for a single LPAR. For those wondering why there are eight connections but only enough for four LPARs, this is because links are assigned in pairs for redundancy. We cover this and other issues in more detail later in this chapter.

2.1.2 Supported frames

The pSeries HPS is a 4U rack-mounted drawer that must be installed into a supported frame. Essentially, the supported frames are frames that come standard with a compatible Bulk Power Assembly (BPA) and would otherwise be capable of supporting a compatible host system. Supported frame configurations are shown in Table 2-1. The frames are serially cabled similarly to p655 cabling requirements in *IBM eServer pSeries 655 Service Guide*, SA38-0618. We cover this more in detail in 2.2, “Switch connectivity” on page 34.

Table 2-1 Supported switch frames

Frame type	Host system	Maximum switches
7040-61R	7040-681/671	1
7040-W42	7039-6M2	1
7040-W42	switch-only single	8
7040-W42	switch-only dual	16*

System frames

When the HPS is installed into a system frame, it must be installed in the bottom drawer location. Only one switch is allowed in a system frame at this time. In general, the switch will redundantly connect to the power subsystem. Cable clearance must be provided for the two stiff copper cables from the switch to each HPS-attached LPAR. Specifics of the cabling are discussed further in 2.2, “Switch connectivity” on page 34.

Switch-only frames

The HPS can be installed into a dedicated switch-only frame. This installation may be called for when system rack space is at a premium, when large numbers of switches are required, or when customers are planning for future growth.

Switch-only frames are the same frames used for p655 servers. Generally, the HPS placement locations are the same as p655 locations. However, placement rules are more flexible. One major difference is that when more than two switches are installed into the same frame, the included 4" frame extender must be replaced with a 24" Frame Extender (FC6234). This upgrade provides the mechanical support and organization for the bulky copper cables, as can be seen in Figure 2-3 on page 19, and will require additional floor space planning.

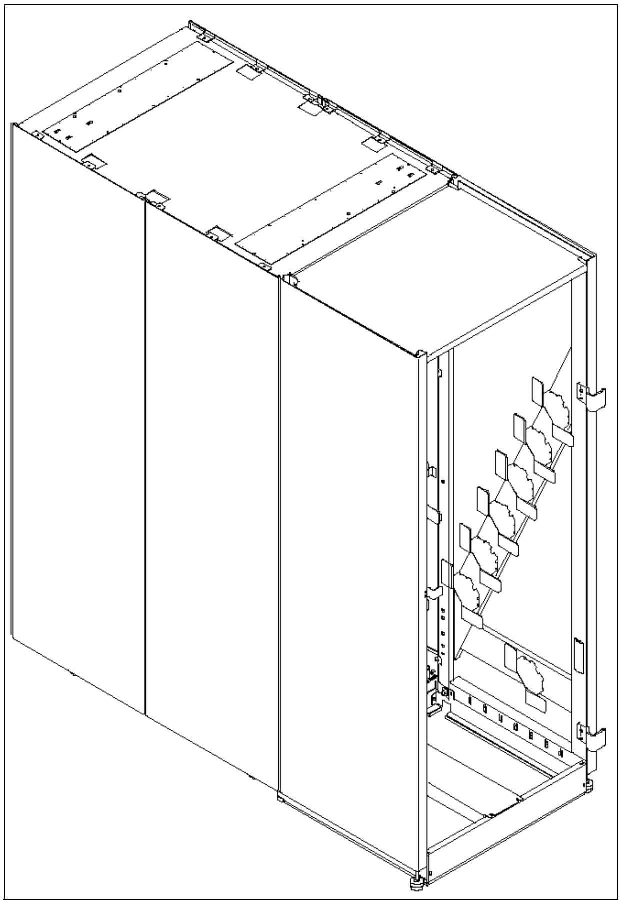


Figure 2-3 Frame with 24" extender

Switch placement

In a switch-only frame configuration, up to two frames may be used. Frame 1 is called the A frame and frame 2 is called the Z frame. There are no frames in between and only two frames may be joined. If additional frames are required, a new A frame will be required.

When identifying frames from the front, the A frame is on the right and contains the power subsystem. The Z frame is on the left and is a slave to the A frame for power connections. For an example, see Figure 2-4 on page 20.

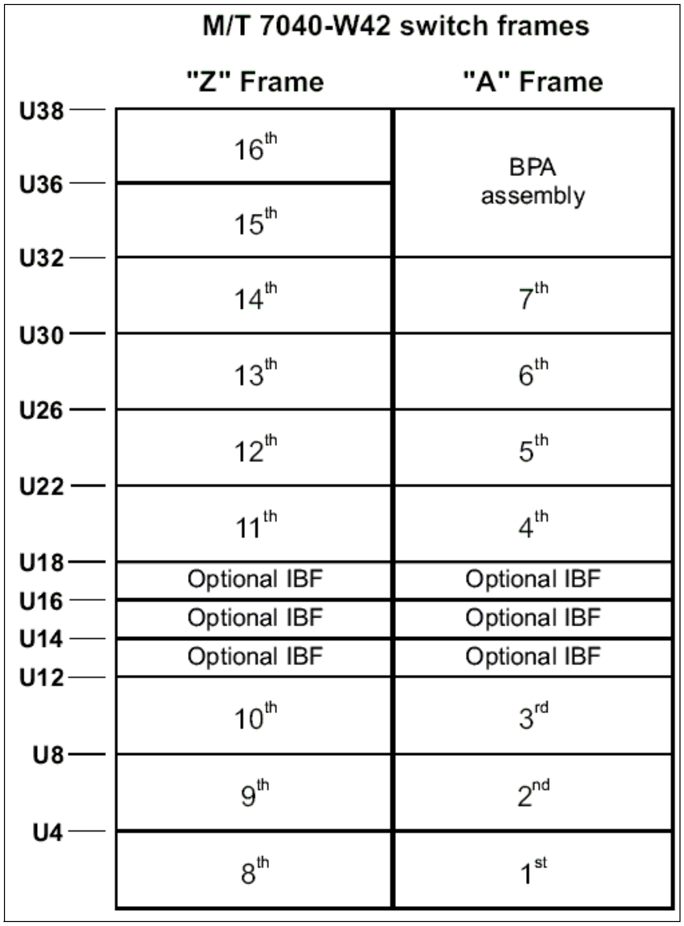


Figure 2-4 Switch-only frame example

When installing switches into a switch-only frame, it is recommended that ISBs be installed alternatively, such as in positions 1, 3, 2 and 4. This allows for better cable clearance prior to expansion. It is also recommended that Node Switch Boards (NSBs) be installed into the top four slots and ISBs in the bottom four slots. This will also help keep cables tidy, since the NSB cables need to exit the frame through the bottom of the rack.

The estimated maximum size for a dual-frame switch-only, fully populated configuration, including frame extenders, is shown inTable 2-2.

Table 2-2 Maximum anticipated specifications for switch-only frame

Measurement	**Metric SI**	**US Imperial**
Height	202.3 cm	79.72 inches
Width	157.5 cm	62 inches
Depth	210.3 cm	82.80 inches
Weight	5184 kg	11429 pounds

Should additional switch-only frames be needed, another set of A and Z frames can be installed separately. While they will be administered as a separate set of frames, they can be logically connected with the same HPS network assuming all cabling and administration requirements are met. At this time, a special bid is required for any configuration exceeding two switches or 32 server ports.

Power supply

According to the formal documentation, redundant power is a requirement for proper operation. We highly recommend this for stability. However, the systems will continue to operate with only single power. If it is necessary to operate without redundant power for any length of time, be aware that firmware updates will require manual intervention as per A.2.4, “Upgrading the frame microcode using the instfru command” on page 229.

Large numbers of switches may require additional BPRs (Bulk Power Regulators) in the BPA. A detailed diagram of BPA components is shown in Figure 2-5.

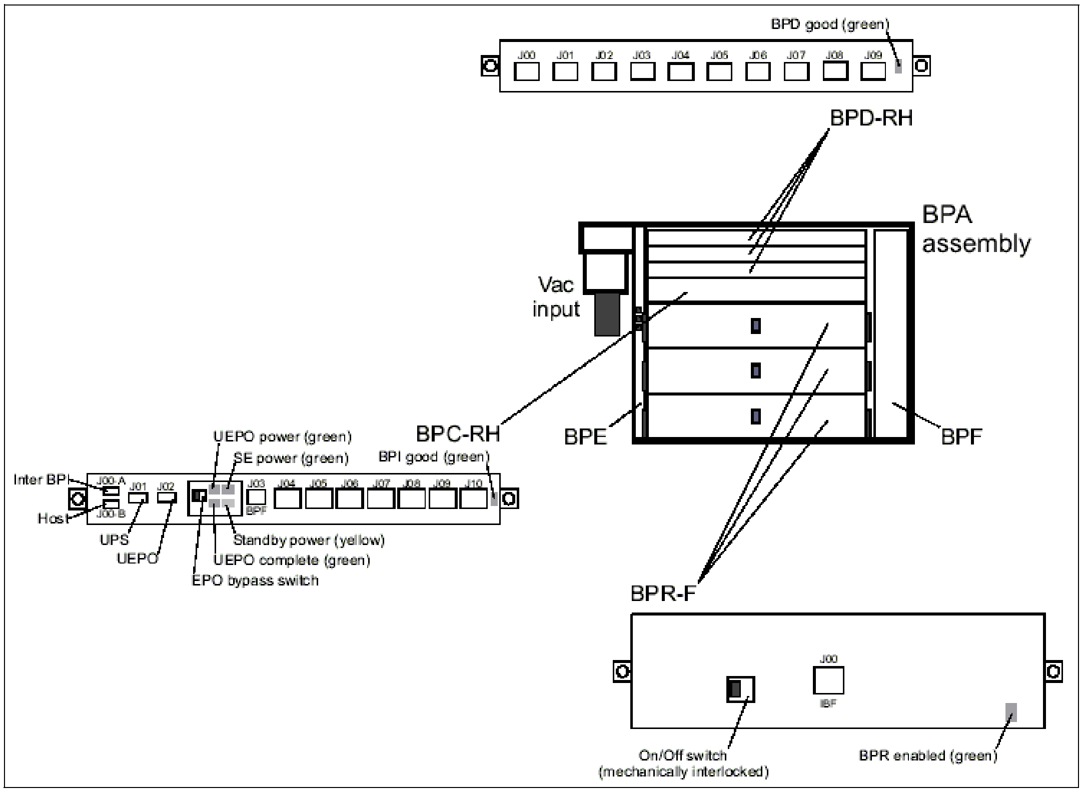


Figure 2-5 Components of a Bulk Power Assembly (BPA)

2.1.3 Switch overview

The pSeries HPS is a four-EIA-unit subsystem, so it occupies the same space in a rack as an I/O drawer or a p655 processor drawer. The HPS may be installed either in the bottom location of the p690 and p655 frames, or it may be installed into an empty p655 frame used as a switch-only frame.

The switch drawer contains all of the components of the switch itself: the riser cards, the switch board, the Distributed Conversion Assembly for Federation (DCA-F), several fans, and the drawer housing. A detailed diagram of these components is shown in Figure 2-1 on page 16.

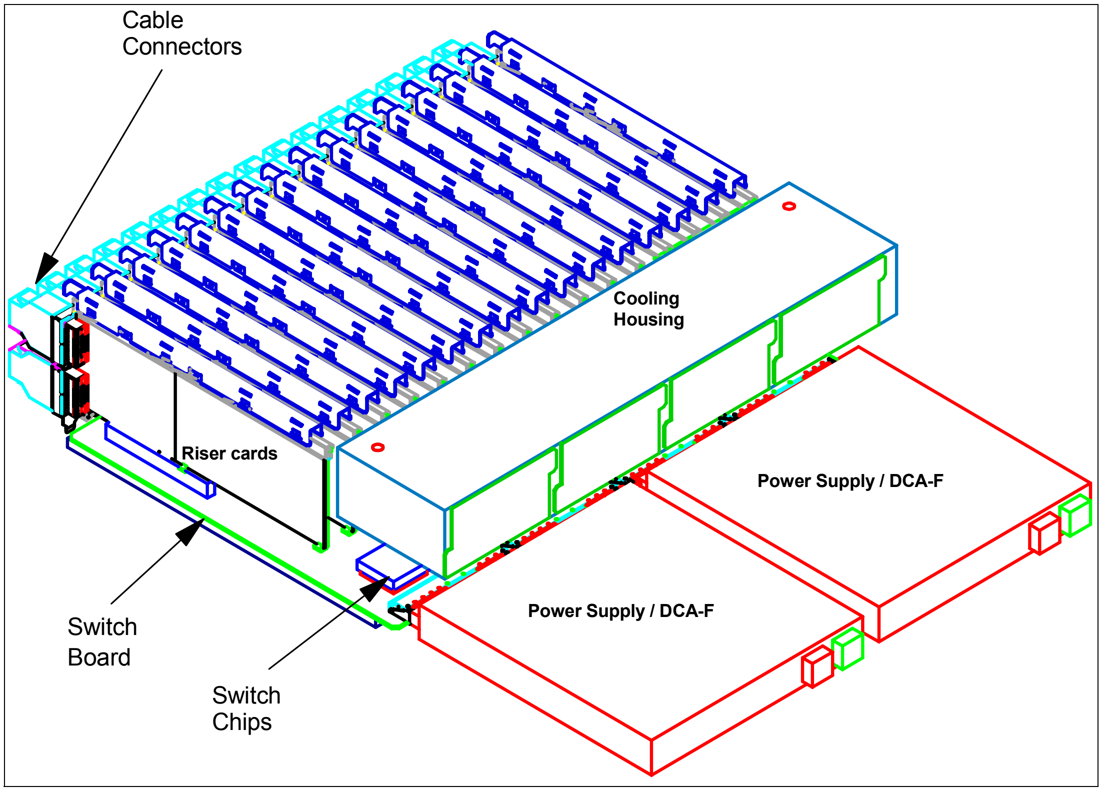


Figure 2-6 pSeries High Performance Switch diagram

The switch itself is required for an HPS network. During NUMA development, point-to-point connections were tested and used. Since then, this ability has been abandoned due to severe technical constraints. As such, when planning for an HPS network, plan on the switch.

The HPS may be ordered in one of two “personalities”. One is the Node Switch Board (NSB), which may also be referenced with an interim name as Server Switch Board (SSB). The NSB may be configured with up to 16 ports for server attachments and 16 ports for switch-to-switch attachments. Ports are provided via riser cards marketed as switch port connector cards. Not all riser cards must be installed. However, unused slots must be filled with the appropriate spacers for adequate airflow.

The other personality of the HPS is an Intermediary Switch Board (ISB). ISBs are used when more than three NSBs are to be joined. They are ordered in sets of two or four in order to ensure adequate inter-switch connectivity. An ISB may contain up to 32 switch-to-switch ports. Generally this switch looks nearly identical to the NSB.

In case you are wondering, there are no switch configurations supporting all 32 ports as server connections. This is assumed to be a limitation imposed to prevent overloading chip-to-chip connections within the switch.

Switch chip

Switch chips exist both on switch boards and on Switch Network Interfaces. They are the core of the entire technology that makes the HPS possible. Each switch chip has eight inputs and eight outputs. Each input supports up to eight virtual channels per link, which can be used for prioritizing traffic. Each channel is allocated a portion of the 8 KB link-level buffer, which is carved into four 2 KB buffers by default. As the data is received, it is immediately passed to the outbound port, or if in use, to the central buffer. If a packet is already queued up, the new incoming packet is blocked. A diagram of these structures is shown in Figure 2-7 on page 23.

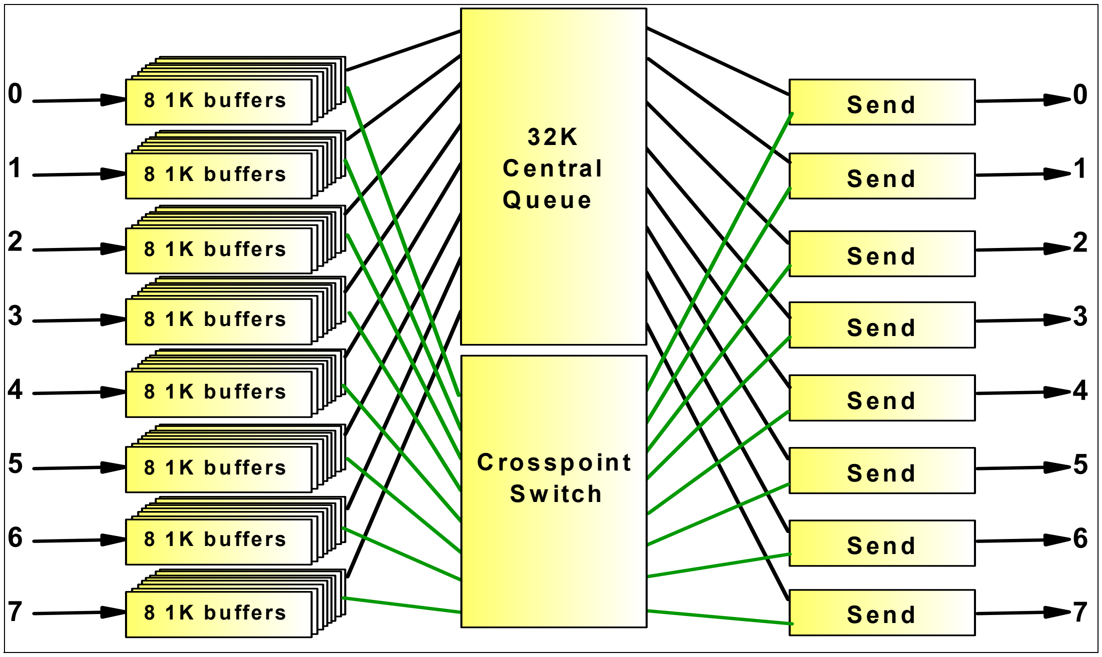


Figure 2-7 Internal diagram of the switch chip

Switch board

The switch board is the key to an HPS network. Each switch board has eight Switch Interface Chips (SICs). Each chip is connected to four other chips. Logically, this is viewed as connecting to each of the chips on "the other side of the board" as shown in Figure 2-8.

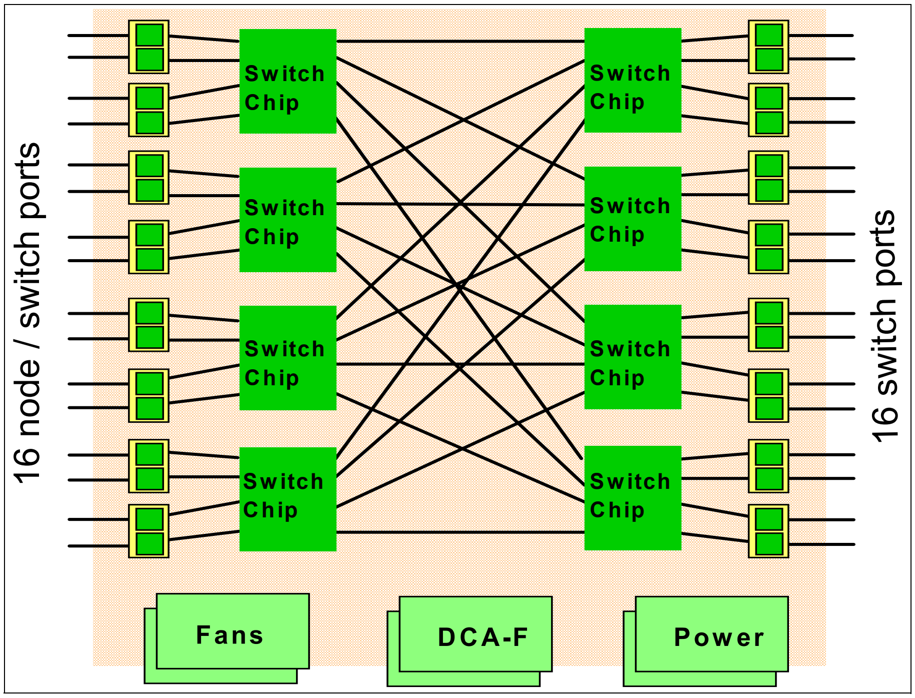


Figure 2-8 Logical view of HPS switch board

This allows cross-chip traffic an additional buffering stage to reduce blocking under heavy load. The difference between an ISB and an NSB is primarily in topology. Physically, they are virtually identical, as shown in Figure 2-9.

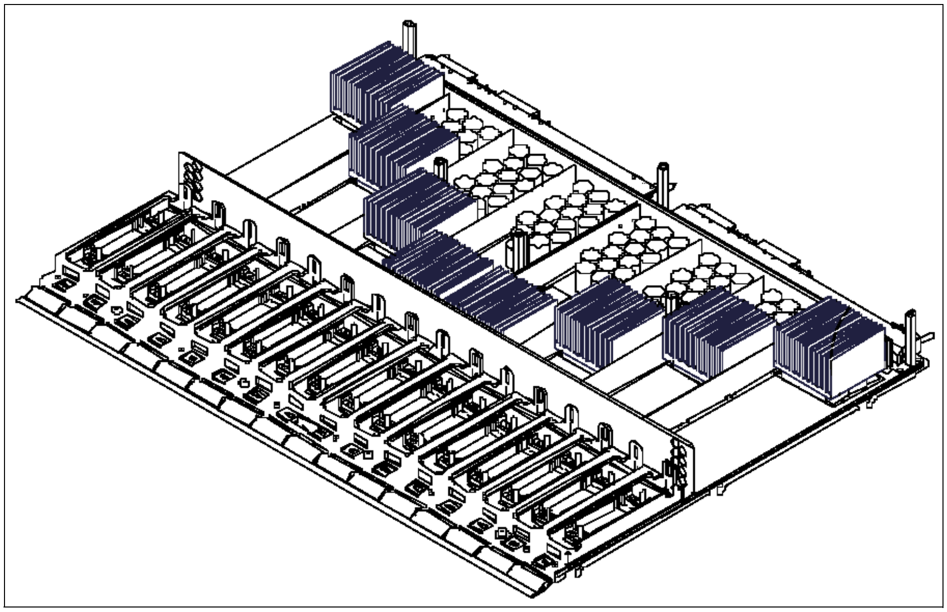


Figure 2-9 Physical diagram of an HPS switch board

Switch port connector cards

Switch port connector cards, or riser cards, are simply the link drivers for the switch board. Each riser provides two links. Each link pair on a riser card is generally mated with a link pair on a Switch Network Interface (SNI).

Risers currently include fiber optic interfaces for switch-to-switch only, copper interfaces for switch or server interconnects, and blanks to fill empty slots. It is important to remember that empty slots *must* contain blank risers to maintain proper airflow, without which the switch may not maintain adequate cooling. See Table 2-3 for switch riser part numbers.

Table 2-3 Switch riser part numbers

Feature code	Part Number	Description	Purpose
FC6433	44P4320	Two-plug, dual-link, copper	Server or switch to switch
FC6436	44P4321	Four-plug, dual-link, fiber	Switch to switch
FC6435	44P4606	Zero-plug, filler	Fill empty slots for airflow

Example diagrams of these riser cards are shown in Figure 2-10 on page 25.

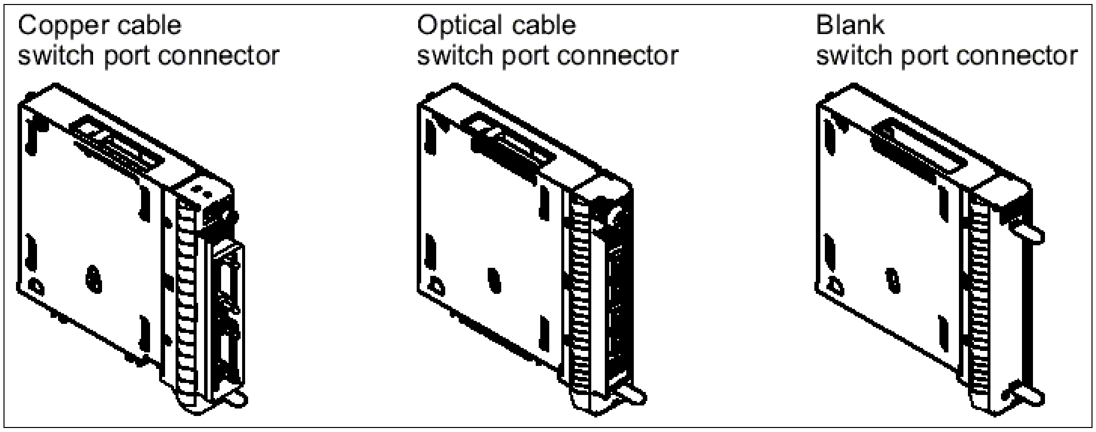


Figure 2-10 Switch port connector cards

In some instances, it is acceptable to stagger connections of a link pair between two switches. When this option is not chosen, riser cards need not be installed sequentially. In order to give cable clearance for visibility of link lights, skip riser slots when possible. Riser card slot numbers are shown in Figure 2-11.

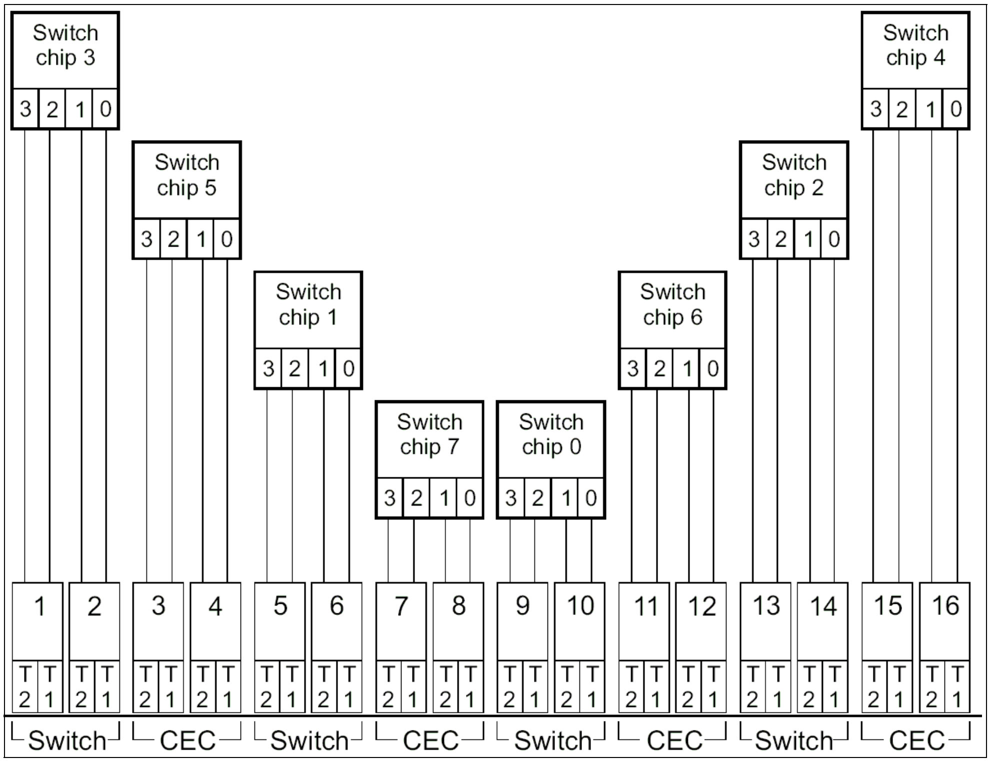


Figure 2-11 Switch port connector card numbering from front view

Redundancy

All internal components of the switch and the SNIs are redundant. This includes switch service processors and inter-chip links. In order to maintain this redundancy, links are always assigned in pairs. If one link fails, then the next shortest path is chosen. Generally, this would be another link on the same SNI. We tested this by pulling a cable from our SNI, and traffic resumed. See Figure 5-2 on page 194 for our results.

2.1.4 Switch Network Interfaces

The network adapter used to provide access to the HPS is known as a Switch Network Interface (SNI) or a Shared Memory Adapter (SMA). Each server that requires an HPS connection will need one or more SNIs. Each LPAR that requires an HPS connection will need one or more link pairs. Each link requires one GX bus. These rules define the scaling limitations, which are explained later in this chapter. A general diagram of the internals of an SNI are shown in Figure 2-12. Details specific to p655 and p690 are contained in the next two sections.

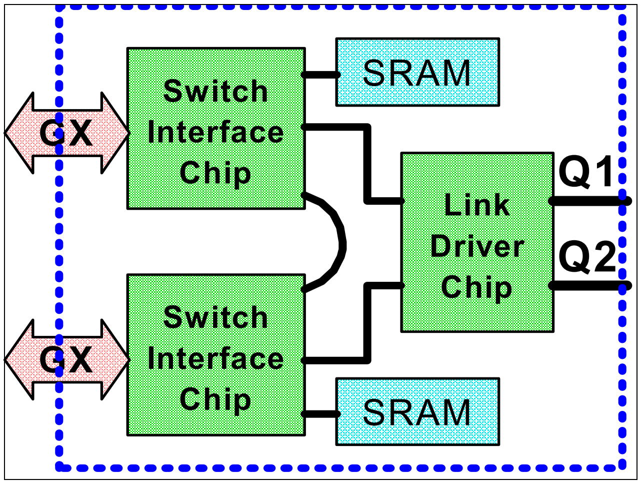


Figure 2-12 SNI logical view

Links

A link refers to a physical connection, usually between an SNI and a switch. The initial offering is limited to 32 server-to-switch links plus 32 switch-to-switch links. This is equivalent to 16 LPARs and two switches. Larger networks are available via special bid, and it is anticipated that the standard offering will grow beyond this initial limitation.

Links are assigned in pairs. This means that the initial offering HPS is limited to 16 servers and two switches. Again, special bids may be made for larger configurations.

Links are limited by the number of GX buses installed in a system. One p690 is limited to four switch-attached LPARs when all four MCMs are installed. A p670 would be limited to half that number. A p655 is limited to only one switch-enabled LPAR. It is possible to configure fewer than the maximum number of switch-enabled LPARs on machines with more than one link pair. When multiple link pairs are assigned to the same LPAR, the single multilink (ML) interface spans all of the links available to that LPAR.

2.1.5 p655 Switch Network Interface

The p655 connects to the High Performance Switch using a Switch Network Interface card attached to the GX bus via the SP Adapter Connector card slot. Only one feature code is available for this attachment: FC6420, a 2-link GX bus mounted card, as shown in Figure 2-13 on page 27.

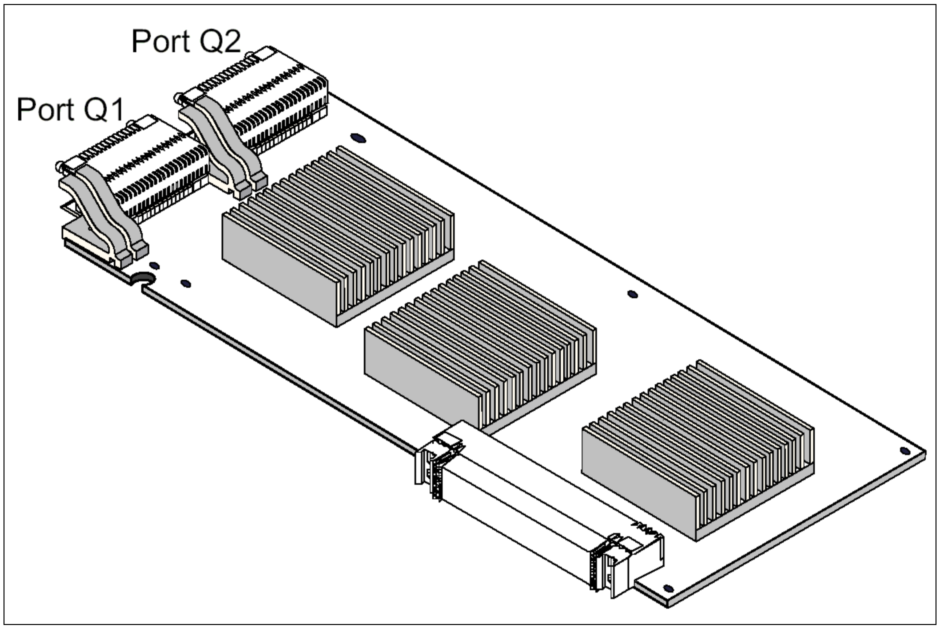


Figure 2-13 p655 two-port GX card

p655 card placement

Each SNI port requires its own GX bus due to bandwidth constraints. The p655 has two GX buses and two GX slots. See Figure 2-14 for details.

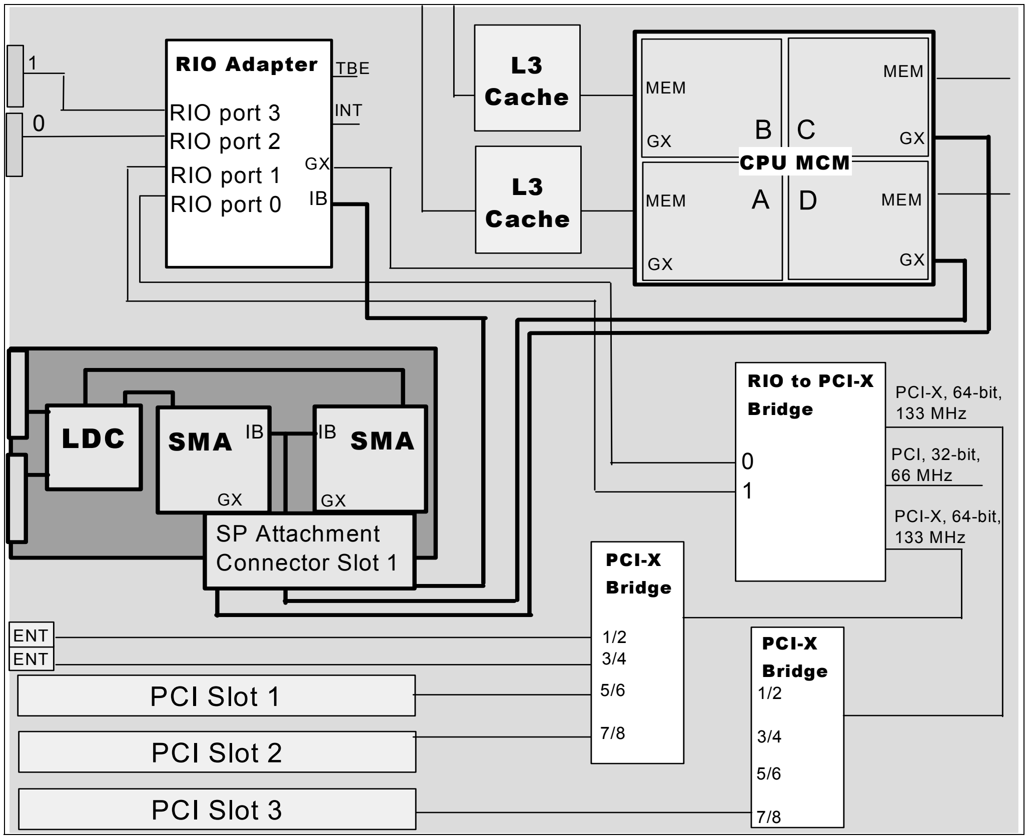


Figure 2-14 p655 logical bus architecture

Furthermore, switch chips produce a good deal of heat and therefore require a direct-attached heat-sink. Because of these conditions, the SNI for a p655 must always be mounted in GX slot 1, and the second GX slot must be empty.

While the logical diagram shows the adapter beside the PCI slots physically, the GX bus connector places the card in the same place as would be occupied by PCI slot 1, as shown in Figure 2-15.

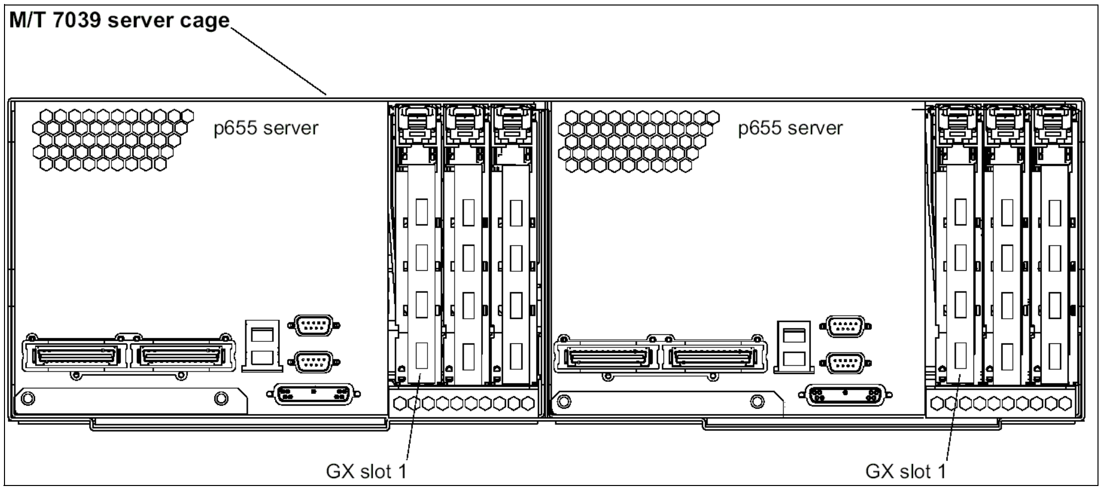


Figure 2-15 p655 physical rear diagram

The maximum number of switch links for a 7039 server is two per CEC, using a 2-link card (FC6420).

p655 SNI link numbering

SNI links, as all devices in pSeries servers, have a variety of device names. The card installed in GX slot 1 is given a physical location of Ux.y-P1-H1, where *x* and *y* depend on the frame location of the p655. The link ports on the SNI are numbered Ux.y-P1-Hz-Q#, where # is either 1 or 2.

2.1.6 p690 Switch Network Interfaces

The p690 connects to the High Performance Switch using a Switch Network Interface (SNI) book. An I/O book is simply a metal carrier that provides proper alignment for the I/O card. The SNI book is attached to the GX buses by installing into GX slot 1 or slot 3. GX slot 0 may not be used because it contains the primary I/O book. The primary I/O book contains system firmware, service processor, and all other system resources other than CPU, memory and other GX cards. There are two feature codes available for this attachment:

- FC6434 IBM 4-Link Switch Network Interface for HPS
- FC6432 IBM 2-Link Switch Network Interface for HPS

See Figure 2-16 on page 29. The two-port SNI contains only ports Q3 and Q4 but is otherwise identical to the four-port card.

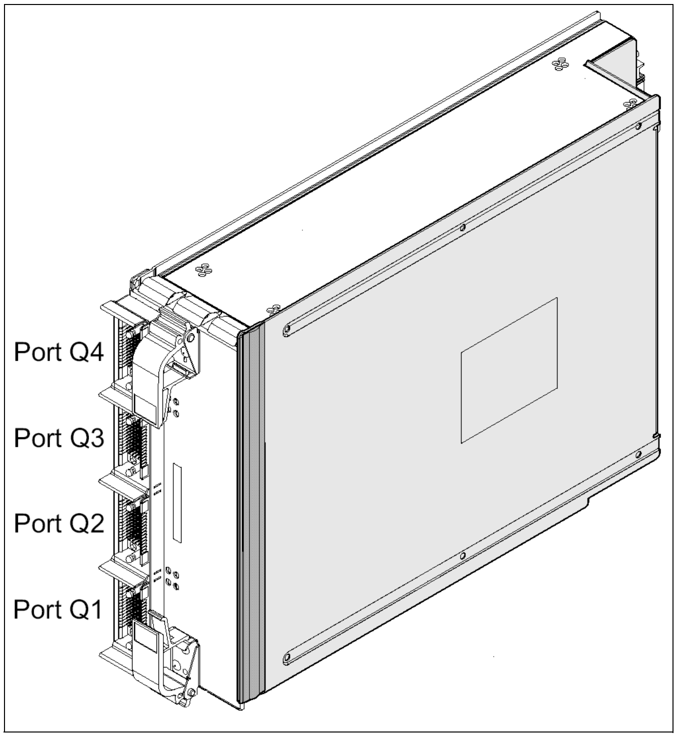


Figure 2-16 Four-port SNI book in alignment for slot 1

p690 architecture primer

p690 I/O architecture is different from most common systems. The system planar is a cross connect with a small amount of logic. This cross connect accepts several major components. On the front edges are the memory buses and slots. In the center are the CPU modules and the oscillator. Between them are the L3 memory cache modules. On the rear are power cards. Last, but not least, also on the rear edges are the I/O slots, known as GX slots.

Each GX slot contains four GX buses. 7040 machines contain one GX bus per CPU. Each MCM contains four CPU chips (single or dual core). The GX buses on each MCM are split between two GX slots. Therefore, a system fully populated with MCMs will present each GX slot with four GX buses total.

GX slot 0 is special and contains bypass power connections. Slot 0 must always contain the primary I/O book. This book contains two boards. One board provides four Remote I/O ports plus six Vital Product Data (VPD) sockets. Each MCM requires a matching VPD card. The other two are used for L3 cache and Capacity Upgrade on Demand (CUoD).

GX slot 0 also contains the first two GX buses. As such, CPU0 is always attached to the primary I/O book.

GX slots 1 through 3 are available for system I/O books. There are currently two types:

- Two-sided remote I/O (RIO) books
- SNI books

GX buses are similar to memory buses in that missing MCMs leave dead buses. See Figure 2-17 on page 30 for which GX buses connect to which CPUs.

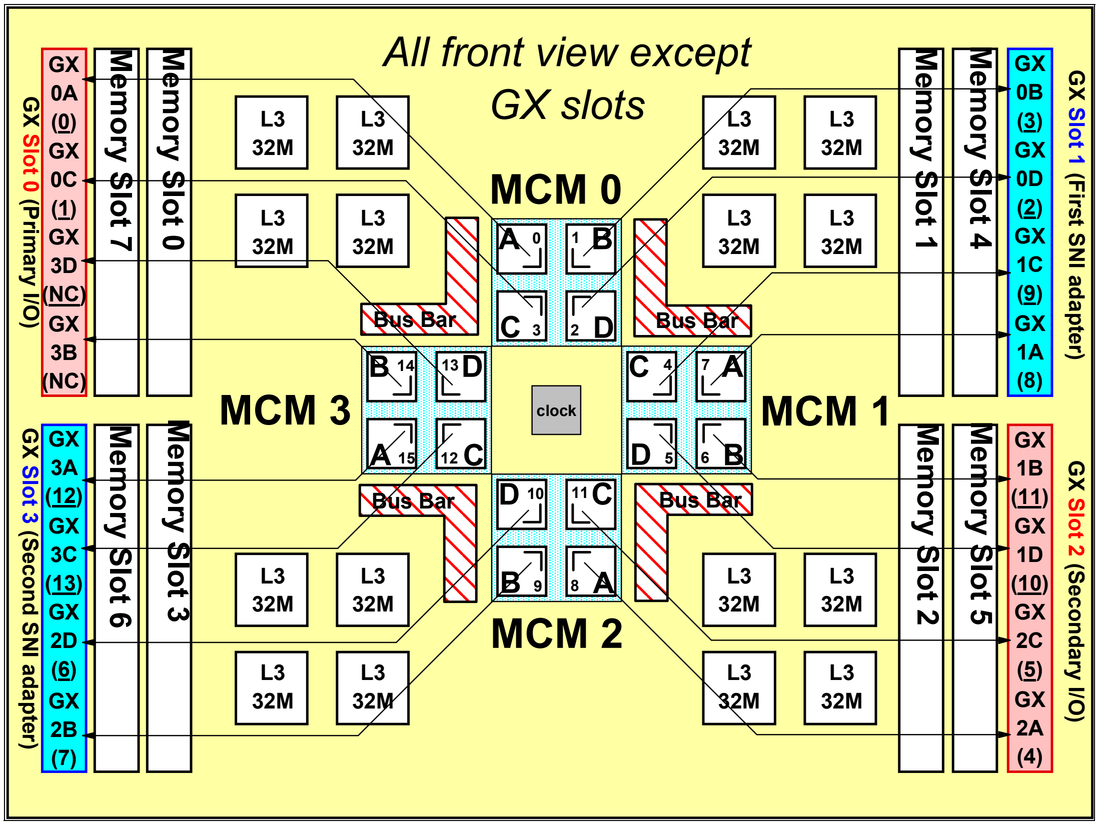


Figure 2-17 MCM to GX bus activation chart

Note: GX buses should match the physical alignment as viewed from the back of the frame; that is, slot 1 has Q4 at the top and slot 3 has Q4 at the bottom. Viewed from the rear of the p690 frame, slot 1 is to the left.

p690 SNI book placement

For a summary of the rules for book placement, see Table 2-4. Lesser configurations are supported even if more MCMs are installed.

Table 2-4 Valid configurations for SNI books in p60 servers

Number of links	Required MCMs	SNI in GX1	SNI in GX3
2	1	FC6432 2-link book	None
4	2	FC6432 2-link book	FC6432 2-link book
4	3	FC6434 4-link book	None
6	3	FC6434 4-link book	FC6432 2-link book
8	4	FC6434 4-link book	FC6434 4-link book

An easy way to determine the number of HPS-enabled LPARs possible for a 7040 is one partition per MCM. While the LPAR is not required to be on the same MCM as your link pair, this may reduce latency. See 2.4.4, “Affinity LPARs” on page 48 for more details.

A summary diagram is shown in Figure 2-18 on page 31.

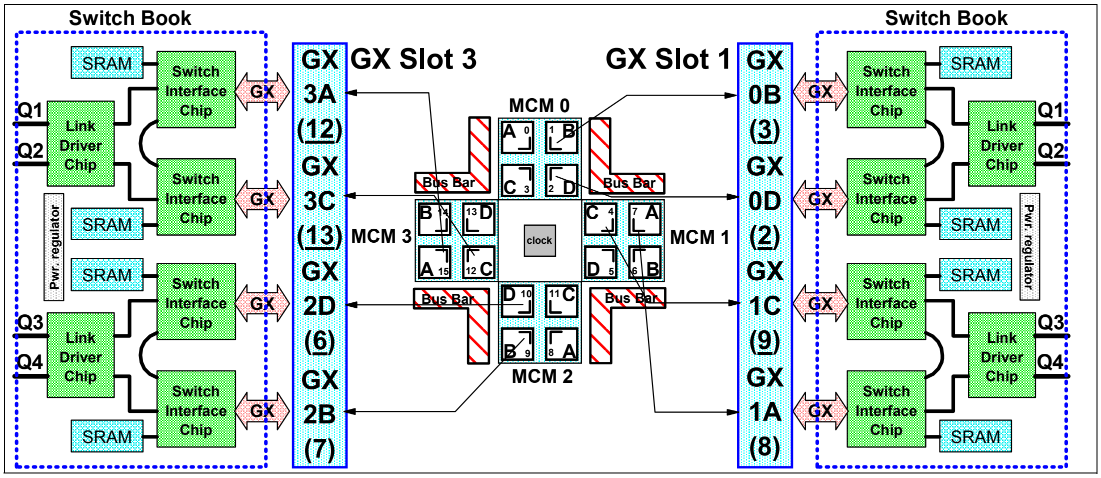


Figure 2-18 SNI-to-MCM relationship (front view)

GX slot ordering

GX slot 1 must be populated prior to GX slot 3. We tested this and the system failed to power up to LPAR ready. The LED panel would display B00E for a longer time, power cycle, go back to B00E, and finally fail with a MOPS 8-digit LED code.

When both a four-port and a two-port SNI book are used in the same CEC, the four-port must be in slot 1 and the two-port must be in slot 3. We did not have an opportunity to test this configuration. However, we believe that this may fail with a manual operations (MOPS) error because of the lack of firmware support for this configuration.

GX slot 2

GX slot 2 may not be used for an SNI. The primary reason for this is internal bandwidth constraints within the Fabric Distributed Controller (FDC). The FDC is the switch internal to each MCM that allows MCM-to-MCM communication. Slots 0 and 2 are kept for relatively low throughput devices. The primary I/O book only uses two of the four GX buses and the secondary I/O book has two per half without any of the special devices that exist in the primary I/O book.

Even so, we wanted to know if it would technically work in an unsupported environment. When we tested this, the system failed to boot and presented a MOPS 8-digit (0x40000000) LED code. Since we only had two MCMs in our system, this might have been because the first two links were not active on the card due to the missing MCM 1. It might also have failed due to lack of firmware support for an SNI in GX slot 2.

Four-port SNI with only one or two MCMs

The system will still operate if a four-port SNI adapter is installed where insufficient MCMs are available. LPAR management will show all four links listed. However, only two links will have global identifiers. Links without global identifiers are not usable. If unidentified links are assigned to an LPAR, that LPAR will fail to activate due to unavailable resources.

p690 SNI link port numbering

SNI book and port numbering for a p690 are similar to that of a p655. The primary difference is that a p690 has four GX slots and the adapters are contained in I/O books.

The card installed in GX slot 1 is given a physical location of Ux.y-P1-H1, where x and y are based on the frame location of the processor subsystem. Generally for machine type 7040, this should be U1.18.

The card in GX slot 3 is numbered Ux.y-P1-H4. If this doesn't seem to make sense, that's because it does not. Using Figure 2-19 as reference, H2 is GX slot 0, H1 is GX slot 1, H3 is GX slot 2, and H4 is GX slot 3.

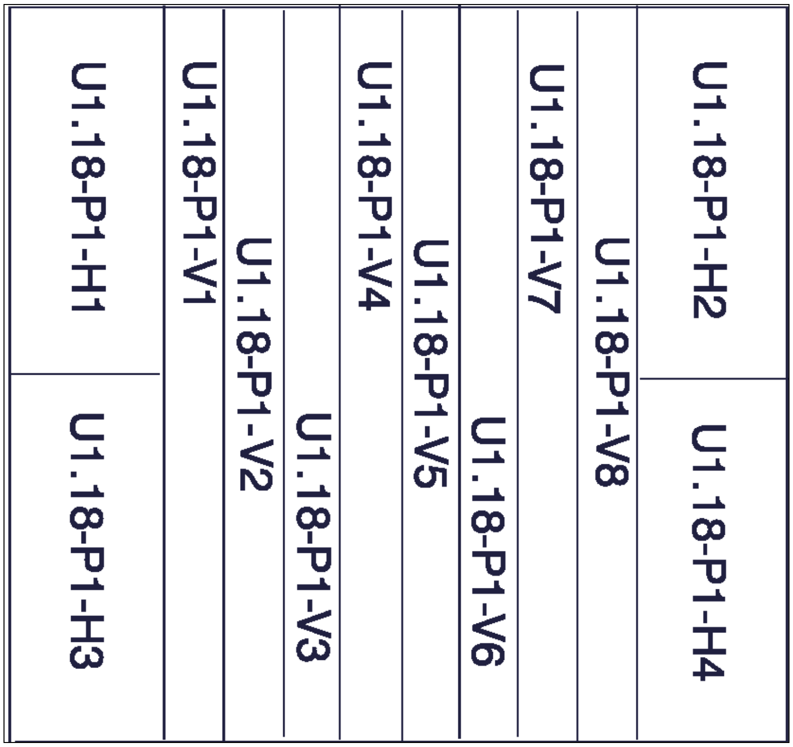


Figure 2-19 7040 CEC rear locations

The link ports themselves are numbered, straightforwardly, Ux.y-P1-Hz-Q#. # is from 1 to 4, although it is important to note that an I/O book installed on the right side is rotated 180 degrees from one installed on the left. Q4 is always towards the outer corner and Q1 is always towards the center, as shown in Figure 2-20 on page 33.

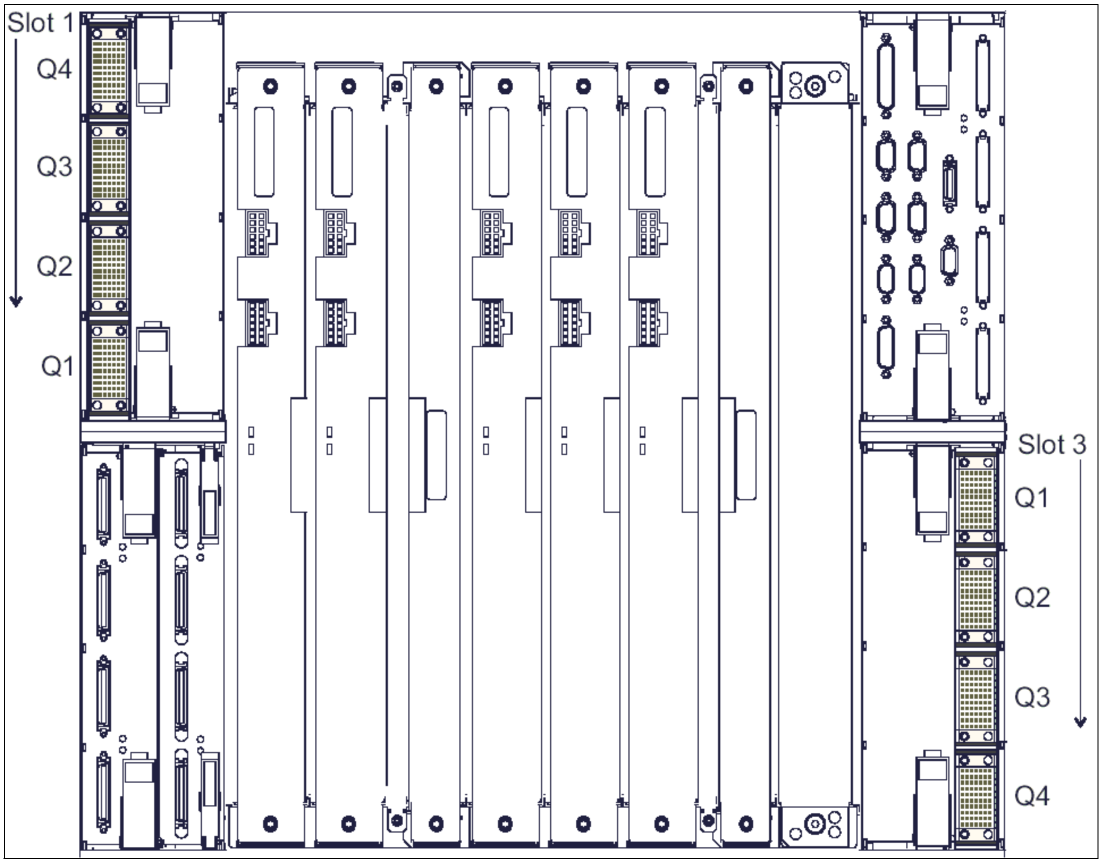
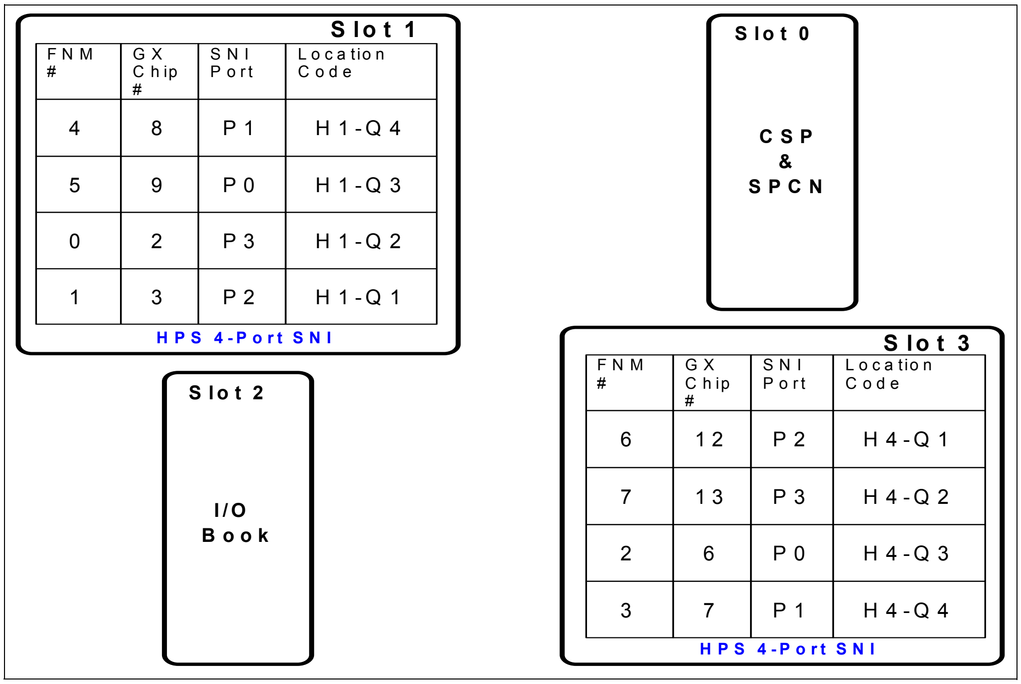


Figure 2-20 7040 CEC rear diagram

To help cross-reference the maps and charts, see Figure 2-21.



Slot 1

FNM #	GX Chip #	SNI Port	Location Code
4	8	P1	H1-Q4
5	9	P0	H1-Q3
0	2	P3	H1-Q2
1	3	P2	H1-Q1

HPS 4-Port SNI

Slot 0

CSP
&
SPCN

Slot 2

I/O
Book

Slot 3

FNM #	GX Chip #	SNI Port	Location Code
6	12	P2	H4-Q1
7	13	P3	H4-Q2
2	6	P0	H4-Q3
3	7	P1	H4-Q4

HPS 4-Port SNI

Figure 2-21 SNI to GX bus map

Table 2-5 decodes the information from Figure 2-21 on page 33.

Table 2-5 Field descriptions within Figure 2-21 on page 33

Term	Definition
SNI Port	This is the SNI port number as referenced internally to the SNI book.
FNM #	Describes the Switch Network Manager's logical designator for the chip connected to the port.
GX Chip #	Internal system number used in CPU controls for that I/O chip based on its location on the GX bus.
CPU #	The CPU number from Figure 2-17 on page 30.
Location Code	Physical location of the slot and port within HMC I/O operations.

2.2 Switch connectivity

Because of the major design changes within the HPS, as compared to the SPS, there are many new cables. In this section, we discuss the cabling requirements and recommendations for an HPS.

2.2.1 Administrative networking

To reduce network contention and improve security between Switch Network Managers (SNMs), it is recommended that a separate, trusted network be installed. The trusted network should connect all HMCs and CSM masters and contain no other nodes. Use this network when defining the IP of the HMC to CSM for power control. This will prevent LPARs from having direct access to SNM out-of-band control data and CSM to HMC passwords. The Hardware Server process running on HMC will automatically use this network for its own traffic.

If you choose to use this configuration, and the CSM Master is also an LPAR, then that LPAR should have three or more Ethernet segments:

- HMC to LPAR LAN
- SNM Trusted LAN
- Public LAN

If the CSM master allows user logins, then it is recommended that access controls of the customer's design and choosing be implemented to limit user access to the trusted network. There are no automatic methods for this, so it is recommended that the CSM master not be allowed for everyday logins.

Presently, the trusted LAN is restricted to 10 Mbps or 100 Mbps Ethernet using the adapters listed in Table 2-6.

Table 2-6 Supported HMC network features

FC4962 10/100 Ethernet/LAN encryption (Ethernet II) type A-F, 32-bit
FC2969 Gigabit Ethernet Fiber, 64-bit
FC2975 Gigabit Ethernet (UTP) 1000BT, 64-bit

For a 7040-61D I/O drawer, 32-bit cards should generally be restricted to slots 8, 9, 10, 18, 19 and 20. While not a requirement, this will leave 64-bit slots available for other resources.

Details of PCI adapter placement may be found in *RS/6000 and @server pSeries Adapter Placement Reference for AIX*, SA38-0538.

Should you choose not to use a separate trusted LAN, it is generally acceptable to use the HMC administrative LAN, provided that an Ethernet switch is used. Please note that there are scaling limitations with the number of HMCs and LPARs on the same network segment due to RMC broadcast traffic. There is no official rule or scaling model for this, although practical experience has shown that more than 32 RSCT peers on the same 10 Mb subnet would likely be a problem.

2.2.2 Serial cables

Rather than a supervisor network as in the SP, pSeries servers have the System Power and Control Network (SPCN). The SPCN network generally consists of serial communications embedded within the DC power cables, one RS232 per Bulk Power Controller (BPC) and two RS422 ports per BPC. The RS232 connection is already used by the CEC, and the CEC-to-HMC connection is already strained at 19200 bps.

In order to provide direct access to the SPCN, the HMC requires an RS422 connection to each Bulk Power Controller. SPCN code can be updated through this connection via the HMC frame microcode download. However, this will still typically be performed via download from the LPAR with Service Authority defined in its profile.

Power commands will now be issued through the RS422 connection, as is the case already for p655 servers. Switch programming is performed through both the existing CEC serial connection for the SNIs and through the RS422 connections for the switches themselves. An example is shown in Figure 2-22.

.

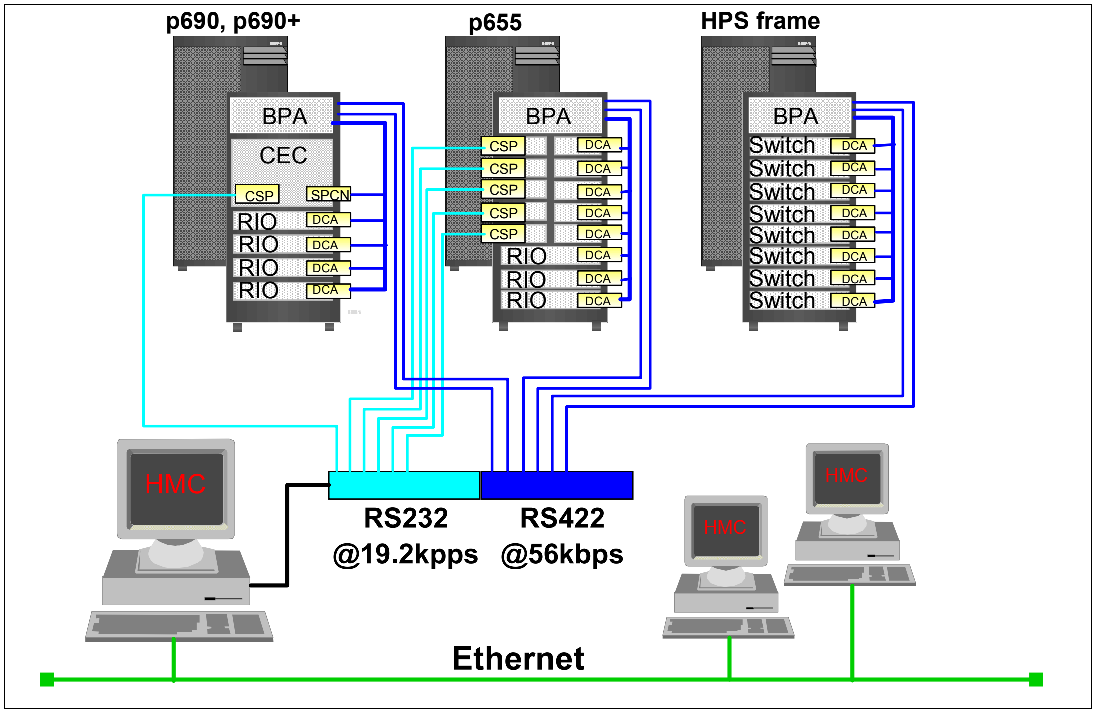


Figure 2-22 General HPS cabling example

Since the HPS requires two RS422 serial connections per frame and one RS232 serial connection per processor subsystem, the HMC requires a multi-port asynchronous serial adapter capable of both RS232 and RS422. See Table 2-7 for a list of parts.

Table 2-7 Serial cable feature codes

Feature	Description
FC 8123	RS-422 cable, 15 m, HMC to frame
FC 8122	RS-422 cable, 6 m, HMC to frame
FC 2943	8-port async adapter
FC 2933	128-port async adapter
FC 8137	2.4 Mbps RAN, 16 ports, stand-alone
FC 8136	2.4 Mbps RAN, 16 ports, rack-mount
FC 8131	1.2 Mbps controller cable, 4.5 m, async-to-RAN
FC 8132	1.2 Mbps controller cable, 23 cm, async-to-RAN
FC 2934	2.4 Mbps serial cable, EIA232
FC 3124	2.4 Mbps serial cable, crossover, EIA232, short
FC 3125	2.4 Mbps serial cable, crossover, EIA232, long

CEC- to-HMC connection

A CEC-to-HMC connection is an RS232 connection,running at 19200 bps, with no control lines. The communication protocol uses Serial Line Internet Protocol (SLIP) framing to the Common Service Processor (CSP) for controlling both Central Electronics Complex (CEC) and Shared Memory Adapter (SMA), also known as Switch Network Interface (SNI) or Switch book.

Note: The HMC's second serial port should be used for the modem and not for an additional CEC. This is for clarity and is not an absolute requirement.

BPC-to-HMC connection

A BPC-to-HMC connection is an RS-422 connection, running at 57600 bps. There are two connections per frame containing a switch (one to each BPC).

You will spend many hours troubleshooting if you don’t verify that both of your BPC cables are set to RS422 by using the `configAsync` command on the HMC after the `digiConf` command has been run. After configuring the serial ports and connecting the serial ports, reload the HMC serial drivers. This may be done by rebooting the HMC or from a root shell on the HMC by running `/etc/init.d/epca restart`.

In a non-HPS environment, power firmware was updated through the CEC serial connection, which would pass the firmware onto the System Power Control Network (SPCN), which would then pass into the Bulk Power Controller (BPC), and finally to the respective components. Each Bulk Power Controller in a redundant system has one connection to the CEC and two RS422 connections between each other.

In an HPS environment, the RS422 cross connects are removed, and one connection from each BPC is routed to the HMC. The Hardware Server process provides monitoring and firmware services for both Bulk Power Assemblies (BPAs) through this RS422 connection.

The major benefit is throughput (57600 bps). However, this also allows Hardware Server process to become a peer on the SPCN, which is required for JTAG access to the HPS.

Unofficially (thus unsupported), a single RS422 cable per rack may be used, and a wrap cable may be left in place between the two BPCs.

For details on how communications actually reach the switch, see 2.5.5, “Switch network manager (fnmd) daemon” on page 53.

> **Note:** While the eight-port asynchronous adapter natively supports both RS232 and RS422 modes, the 128-port RAN only supports RS232. By the time the HPS is Generally Available (GA), an interposer to convert between the two should be available to ship as part of the RS422 cable set.

2.2.3 Switch cables

Physical cables embody server-to-switch and switch-to-switch links. Physical links are presently cabled in pairs. Current limitation is 32 hardware (switch to server) links (plus 32 switch-to-switch links). Single link adapters are anticipated to become available. However, these will be created by wrapping one plug in a link pair.

Switch-to-node connections

On POWER4 and POWER4+™ architecture, server-to-switch connections are made with fairly bulky copper cables as shown in Figure 2-23. Fiber to the SNI may come about in the next line of hardware, but no official statement has been made to that effect.

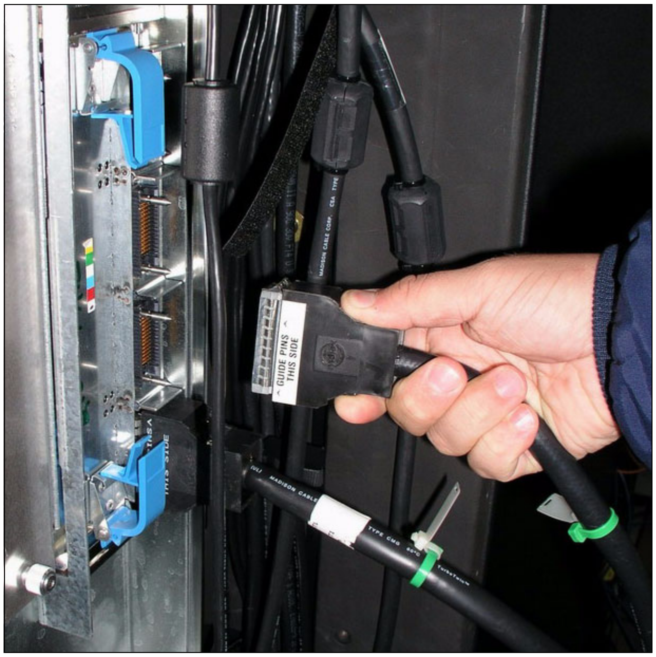


Figure 2-23 Cable connection

Switch-to-switch connections can be via copper or fiber. The available features are listed in Table 2-8 on page 38.

Table 2-8 Switch cable feature codes

Feature	Description
FC 3257	Switch cable pair, 40 m, fiber optic
FC 3256	Switch cable pair, 20 m, fiber optic
FC 3167	Switch cable, 10 m, copper
FC 3166	Switch cable, 3 m, copper
FC 3161	Switch cable, 1.2 m, copper

A link pair may be converted to single-link use by installing an FC 6437 converter wrap plug into the Q2 port of an FC 6420 2-link SNI card or the Q4 port in an FC 6432 2-link SNI book. This feature is not available with the initial offering and is not compatible with 4-link SNI cards. An example of the converter with a 2-link book is shown in Figure 2-24.

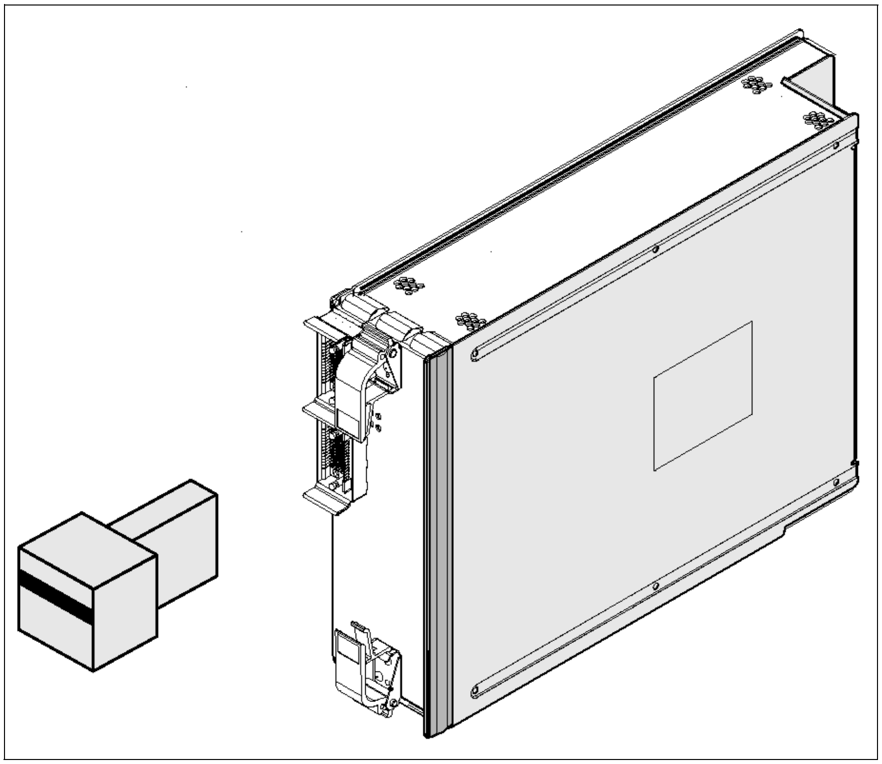


Figure 2-24 Single-link converter plug

When available, the converter will enable the LPAR link pair to use only one cable. The purpose of this is to reduce the number of switch-side ports required for servers. Two LPARs will be able to run from the same switch port connector card. This option will allow a single Node Switch Board (NSB) to serve up to 16 single-link LPARs rather than eight.

This option does not allow a server-side link pair to be split between LPARs. In other words, when you configure the network, both links of the converted link pair must be assigned to the same server or LPAR. When using single-link nodes, all cabling must be sequentially ordered.

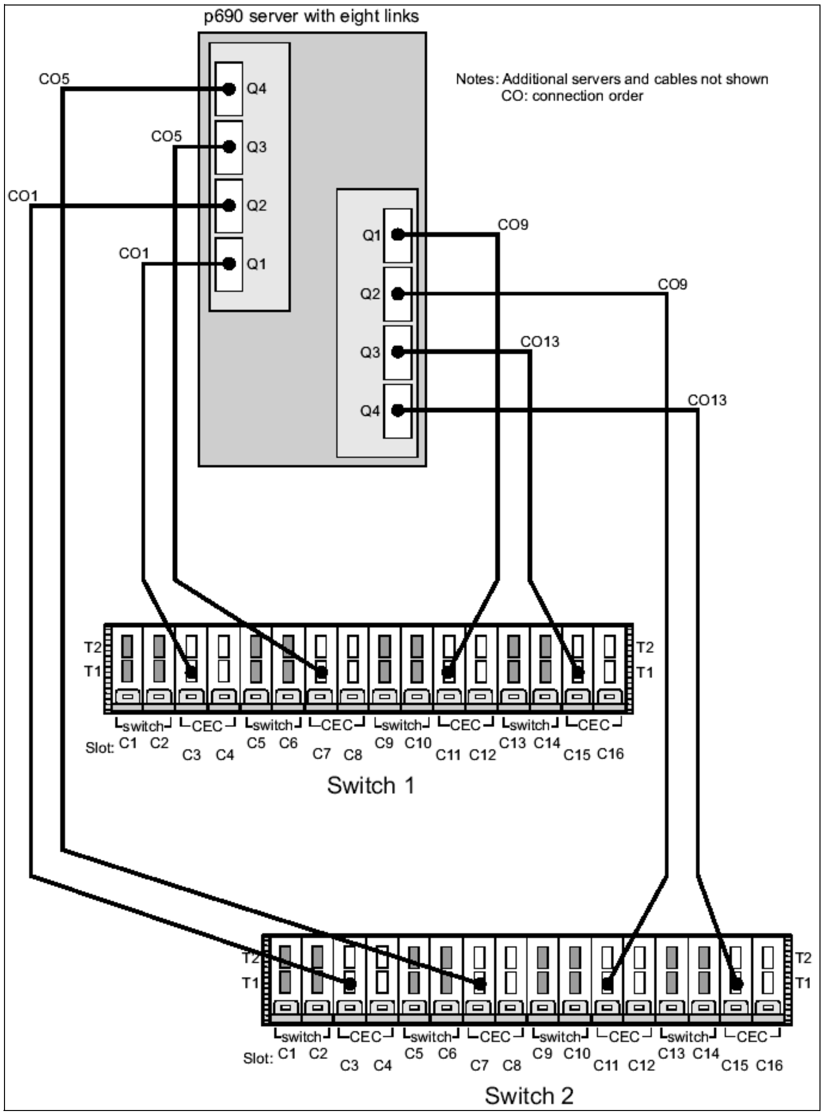
Cable planning

At one point, documentation stated that link pairs on the same switch must connect to the same SPCC. However, this restriction can no longer be found in the documentation. Notice in

Figure 2-26 on page 40 from the service guide that this appears to no longer be a requirement.

Dual plane

In the development lab, an experimental dual-plane configuration splits the link pair between non-interconnected switches. An example of this configuration is shown in Figure 2-25.



p690 server with eight links
Notes: Additional servers and cables not shown
CO: connection order
CO5
CO5
CO1
CO1
Q4
Q3
Q2
Q1
Q1
Q2
Q3
Q4
CO9
CO9
CO13
CO13
T2
T1
switch
CEC
Slot: C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16
Switch 1
Switch 2

Figure 2-25 Dual-plane configuration

Notice how the ports are staggered. This is to ensure that each link pair has connectivity to each switch.

Single switch

Some networks are required to be cabled sequentially. Networks with any single links or an odd number of links require sequential port population, and networks with even number links but odd number of Server Switch Boards (SSBs) must also be cabled sequentially. See Figure 2-26 on page 40.

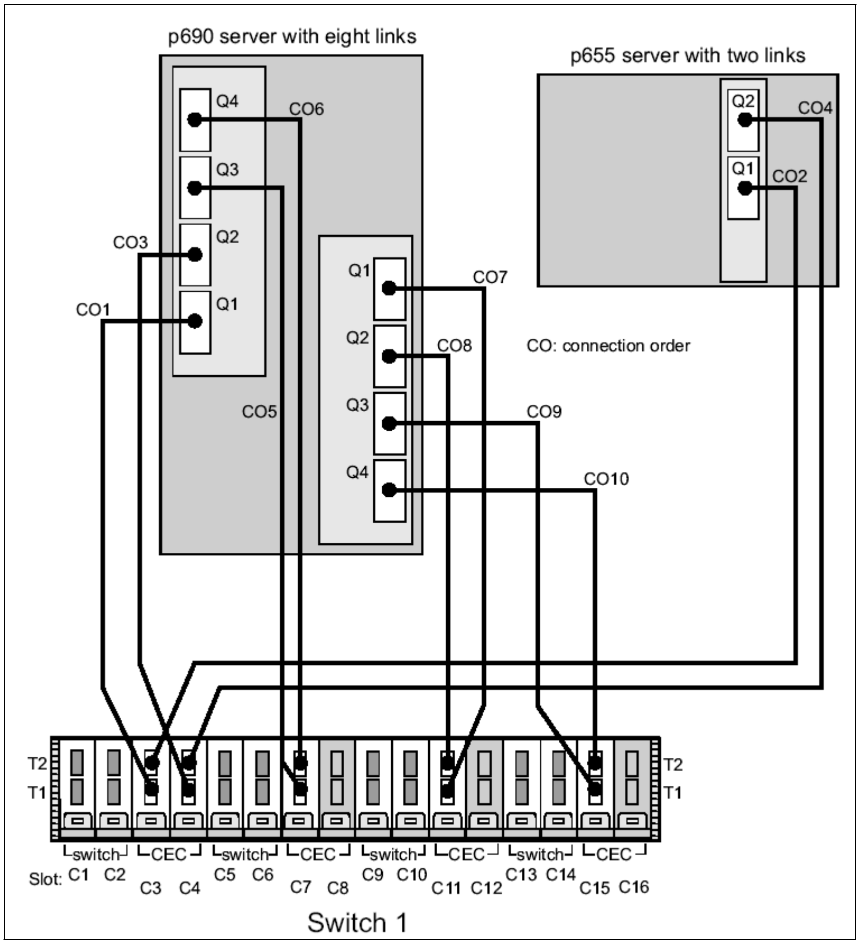


Figure 2-26 Sequentially cabled switch network

Please notice that the riser ports are cabled numerically from the bottom up, as shown in Figure 2-26.

Dual switch, single plane

It is acceptable in a two-switch, single-plane, all dual-link configuration to connect each link within a pair to the same port on different switches, thereby preventing a switch failure from crippling a server. The cable connections would be identical to those in Figure 2-25 on page 39, with the addition of switch-to-switch connections.

> **Tip:** During our testing, it was not necessary to cable unused SNI ports, despite what the service guide states.

Other topologies

Dual-plane, quad-switch and dual-plane, non-staggered configurations are not documented. It is feasible that these may work along the same principles as above. However, it is unknown whether they would work or would be supported. For the time being, we recommend using only the documented cabling schematics as shown in *pSeries High Performance Switch Planning, Installation and Service*, GA22- 7951.

Switch-to-switch connections

When using more than one switch per plane, each switch must have switch-to-switch riser slots occupied. These are used to drive the lines that interconnect the switches. Official documentation states that networks with switch-to-switch connections will require *all* switch-to-switch riser slots to be filled and all cross connecting cables to be installed.

In the lab, we found that you only need as many switch-to-switch risers as you have switch-to-node risers. If the switches have an uneven number of switch-to-node risers installed, we recommend choosing the larger number when planning your switch-to-switch risers. You will need the same number of switch-to-switch risers installed in each of the cross-connected switches.

Tip: When possible, both for visibility and mechanical stability, we recommend using fiber connections between switches.

2.3 Firmware

Firmware, microcode, ucode, licensed internal code, or any other name you give to it is the machine code that allows the hardware to operate. It provides for all interaction between the operating system and the physical components of a system. Needless to say, firmware is crucial to HPS functions.

During our testing, we found the available firmware to be intermittently stable. Some versions were fine, while others were not. In several instances, it was necessary to back-level the firmware in order to resolve problems. The most common firmware problems were duplicate frames, missing frames, and IP packet loss and duplication.

The required levels of system firmware and SPCN code are 030922 and 2597 respectively. The GA code was anticipated to be F3.41.2. However, we had trouble with it and expect it to change. Special attention has been given to new HPS installations in order to resolve some of these issues.

Note: Firmware references such as F3.41.2 are decoded to mean Federation, 2003, week 41, build 2. This is an entire package containing system firmware, system power code, device drivers, and SNM code for the HMC. F3.41.2 includes system firmware dated 031024 and switch microcode level 25a7.

2.3.1 System firmware

System firmware, also referred to as General Firmware (GFW), resides in the primary I/O book. We recommend that this be upgraded first. The minimum firmware version required for installing the pSeries HPS is:

- pSeries 690: 3H030922
- pSeries 690+: 3P030922
- pSeries 655: 3J030922

The GA level was anticipated to be 031024 or later and is *highly recommended* to be at the October 2003 or later levels. The IBM service personnel installing your HPS should have it available for installation. System firmware can be downloaded from:

http://techsupport.services.ibm.com/server/mdownload2

Updating system firmware

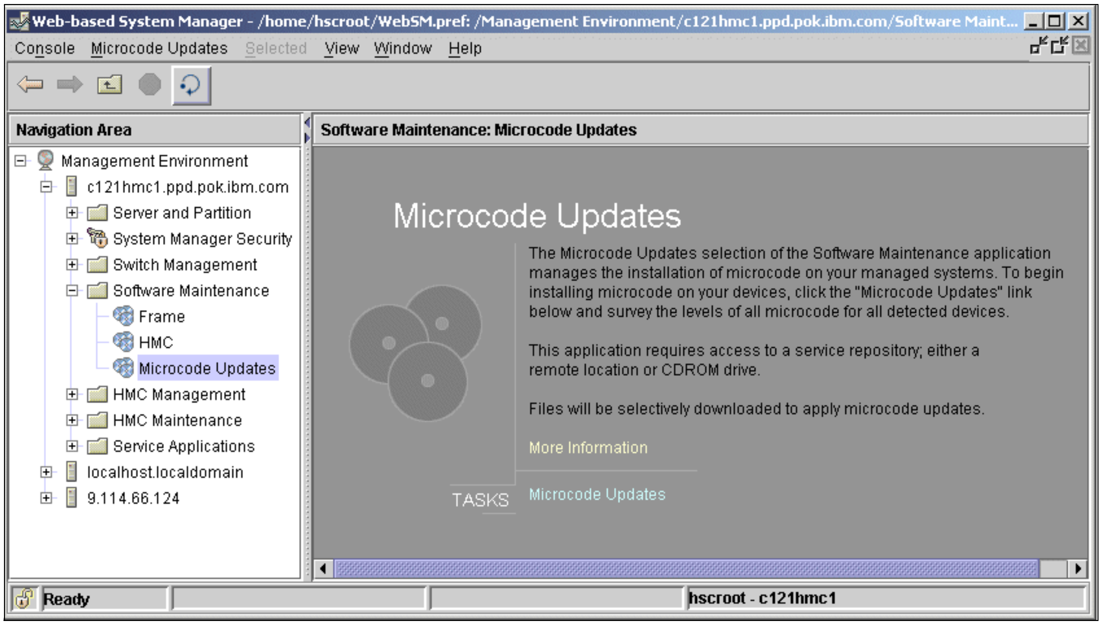
Note: In some instances, namely when firmware update problems arise, it may be required to update the Global Firmware (GFW or CEC firmware) prior to upgrading the HMC.

After a firmware update, the frame must be power cycled.

There are several ways to update the system firmware, depending on the pSeries type, as including:

- Via HMC GUI

  Not mentioned very often, the HMC GUI has a provision to use Inventory Scout to update system firmware, as shown in Figure 2-27.



The Microcode Updates selection of the Software Maintenance application manages the installation of microcode on your managed systems. To begin installing microcode on your devices, click the "Microcode Updates" link below and survey the levels of all microcode for all detected devices.

This application requires access to a service repository; either a remote location or CDROM drive.

Files will be selectively downloaded to apply microcode updates.

Figure 2-27 HMC Microcode Updates menu

  This will scan for updates and present a list of components that you may choose to update.

  Note: Devices in use, such as rootvg disks, will not be updated.

- Online diagnostics

  From the LPAR with service authority, as root, from a terminal, issue the **`diag`** command. Choose **Task Selection -> Update System** or **Service Processor Flash**.

  NIM Client Options may be used to place the node into diagnostic mode if no operating system is installed.

- Shutdown method

  From the LPAR with service authority, as root, issue:

```
# shutdown -u 3H031024.img
```

  The `-u` flag is an option used by diagnostics to update the flash memory and reboot. It cannot be used in conjunction with any timed power on, since its very nature is to update the flash-memory then reboot the system.

On platforms that do not have flash memory, the reboot command and system call will treat this option as a noop (no operation) and just do a normal reboot as if the -r flag had been used. It is not necessary to specify the -r flag when the -u flag is used.

The argument to this flag is the file containing the binary flash memory update image. It must be in the /var filesystem and be present if the -u flag is specified, even if the system has no flash memory. No check is made to ensure the file is in /var, but if it is not, then the update will fail since only the /, /usr, and /var filesystems are present at the time the reboot command is executed.

- update_flash method

  From the LPAR with service authority, as root, issue:

  ```
  # /usr/lpp/diagnostics/bin/update_flash -f /tmp/fwupdate/3H031024.img
  ```

- Service processor (7040 only)

  Prepare firmware diskettes and install from service processor menus per instructions included with the firmware package or from:

  http://techsupport.services.ibm.com/server/mdownload2/7040681F.html

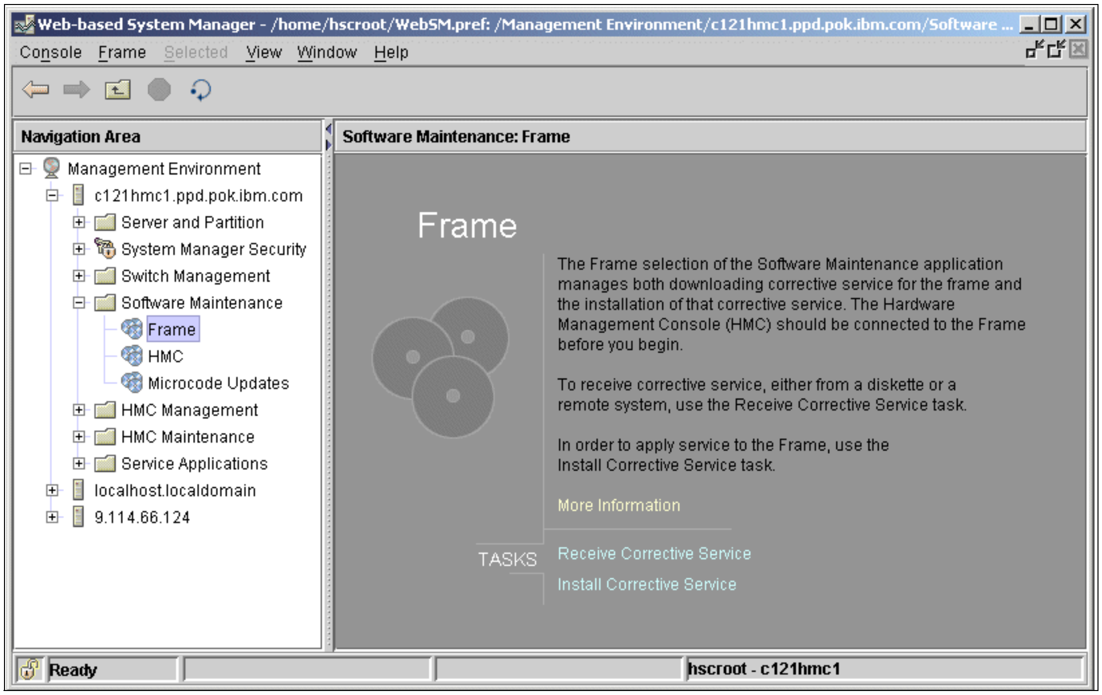
2.3.2 Frame microcode

Functionally, the HPS is a part of the power subsystem. Its code is updated via frame microcode update utilities. The HMC GUI, HMC command **frucode**, HMC command **instrfu** and a full CEC firmware image load should update the microcode levels. The generally accepted method is through a whole CEC firmware update. However, using the **instfru** command, we were able to update all 20 FRUs in under 30 minutes.

Note: Before the installation of the frame firmware, make sure that the CEC is logically powered off and the LCD panel shows **OK**.

HMC GUI

The frame microcode option in the GUI shown under Software Maintenance in Figure 2-28 on page 44 refers to the power subsystem firmware. This can be used to perform power code updates without reloading system firmware.



The Frame selection of the Software Maintenance application manages both downloading corrective service for the frame and the installation of that corrective service. The Hardware Management Console (HMC) should be connected to the Frame before you begin.

To receive corrective service, either from a diskette or a remote system, use the Receive Corrective Service task.

In order to apply service to the Frame, use the Install Corrective Service task.

Figure 2-28 HMC frame menu

Since the frame is attached directly to the HMC, it is also possible to use the Microcode Updates menu. This will survey the frame, the CEC, and all of the LPARs via Inventory Scout and prompt you to automatically download. A list will be presented with recommended actions for each device found. Note that the SNIs will not be listed because they are part of the system firmware.

Using instfru

The HMC GUI retrieves firmware from a public Internet-accessible FTP server and calls **frucode** to install it. If redundant power subsystems are not available, this will fail and the **instfru** command must be used manually from the HMC. The **instfru** command is briefly documented in A.2.4, “Upgrading the frame microcode using the instfru command” on page 229.

In early firmware versions, power subsystem updates through the SPCN from the CEC would fail for this new code. Effectively, the Bulk Power Controllers (BPCs) and the CEC would fight each other over power subsystem firmware loads. In such cases, you had to remove all but one serial connection from the BPCs and to use **instfru** to update the firmware. This should be resolved by the time this redbook is published, but on your initial HPS firmware update, be sure to verify that the SPCN code did actually update.

In some instances with **instfru**, we found that the SPCN code needed to be updated several times in order to update all components. This was due to the BPC updating first and resetting, causing the automatic load to fail. After this, **instfru** showed only three FRUs. The second load would update the second BPC and all of the FRUs would once again be shown. The third load would finish updating all components.

2.3.3 Hypervisor

Hypervisor is the glue that holds all of the HPS components together. It exists within the CEC as Licensed Internal Code and is part of the firmware image. When in LPAR Ready mode, Hypervisor owns all system resources and provides an abstraction layer through which device access and control is arbitrated. It is because of these functions that Hypervisor was chosen to handle the HPS.

The HPS was originally intended to be a NUMA fast-cache adapter. To function as such, it needed to be available prior to system boot. This required much of the switch chip functionality to become part of system firmware, and only Hypervisor had the necessary hooks to provide abstraction of system resources. While the NUMA on POWER4 plans have been dropped, the HPS remains. There is insufficient room in the switch microcode to offload these functions from Hypervisor, and as such, the HPS will always require that the CEC be in LPAR Ready mode and even if there were room, it would be a complete redesign.

Once the switch is installed, fully configure the Switch Network Manager prior to powering up the CEC. Failure to do so may prohibit use of the switch due to improper frame identification within the HMC, thereby crippling its ability to identify components within the switch network. SNM configuration can be done within the HMC GUI, or through /opt/hsc/data/HmcNetConfig.

Restriction: Due to their reliance upon Hypervisor functionality, switch adapters cannot be dynamically allocated (DLPAR).

2.4 Hardware Management Console (HMC)

To use an IBM @server pSeries High Performance Switch (HPS), a 7315-C01 equivalent or better HMC is required. In supporting this class of machine, the initial HPS offering was limited to 32 switch nodes except by special bid. This is primarily because of HMC performance considerations. This limitation is expected to be removed by the next major HMC release and may require the next HMC GA level.

The version of the HMC code required for installing the pSeries HPS is R3V2.4 or later. We highly recommend V3.2.5 or later, since its Switch Network Manager is more stable. Check for the latest version of HMC code at:

https://techsupport.services.ibm.com/server/hmc

Initial releases of the HPS require that Switch Network Manager (SNM) corrective service be applied to the HMC, This corrective service, internally known as update.zip, will be tailored to match the firmware and device driver levels provided. Eventually we expect that this will either be synchronized and stabilized within HMC updates or will be provided as a separately downloadable package.

Sizing and performance

The Hardware Server (hrdw_svr) subsystem can coordinate multiple hrdw_srv processes running on different HMCs (as a single virtual hrdw_srv image). This will allow more than one HMC to share the load of larger HPS installations.

In the future, a non-HMC machine may be allowed to be the Switch Network Manager and Hardware Server. To help facilitate this, new firmware images may be made available that preconfigure the LPARs and remove the need for the HMC.

In order to increase performance of the overall HMC interface, much of the CIMOM code was moved from disk to RAM.

2.4.1 HMC required components

Service Focal Point, Service Agent, and Inventory Scout must to be configured on the HMC for an HPS installation. Actual responsibility of this task changed during our residency, so unless otherwise noted in official documentation, assume that it is your responsibility to configure them.

Service Focal Point (SFP)

The Service Focal Point (SFP) application runs on the HMC and provides a user interface for viewing events and performing problem determination. SFP resource managers monitor the system and record information about serviceable events. The application filters data and groups information related to the triggering event. When appropriate, SFP also initiates a call to the service provider.

By grouping related service information, SFP reduces the complications of diagnosing problems in a partitioned system. Problem determination in a partitioned system is complicated by the fact that each partition functions independently but may share resources with another partition. As a result, more than one partition may report the same error. Service Focal Point recognizes repeating errors, filters them, and reports one serviceable event instead of a long list of repetitive call-home information.

The SFP is able to provide the Field Replaceable Unit (FRU) identification codes for the pSeries HPS in case of a failure in the switch network. Look here for MP_FAIL and MP_DOWN events.

Electronic Service Agent™

Electronic Service Agent (also referred to as Service Agent) is installed on the HMC and monitors the system for hardware errors. Service Agent uses the diagnostic functions provided by AIX, the service processor, and Service Focal Point. Service Agent functions include:

- Providing user-definable thresholds for error reporting
- Automatic problem analysis
- Automatic problem reporting
- Automatic customer notification

The pSeries HPS forwards its Service Agent messages to the Hardware Management Console. Depending on your configuration, the HMC transmits these error messages to the System Administrator or through the system modem to IBM.

> **Tip:** Previously, an HSCP0025 during “Save Upgrade Data” almost always indicated a Service Agent configuration problem. In an HPS environment, this may also be caused by /var being filled with automatic snaps. Use /var/hsc/log/SaveUpgradeFiles.log on the HMC to look for large numbers of switch snaps or other files to be removed prior to a retry of Save Upgrade Data.

Inventory Scout

Inventory Scout surveys your system for information about installed hardware and software. Inventory Scout also generates reports about current microcode levels and the need to install software updates or patches. Two of the primary Inventory Scout functions are:

- Microcode Discovery Service

  MDS generates a real-time comparison report showing subsystems that may need to be upgraded. For further information about Microcode Discovery Service, visit the following URL:

http://techsupport.services.ibm.com/server/aix.invscoutMDS

- VPD Capture Service

  VPD capture service transmits the vital product data (VPD) information for your server to IBM. For further information about VPD Capture Service, visit the following URL:

  http://techsupport.services.ibm.com/server/aix.invscoutVPD

2.4.2 WebSM (optional) remote management

For remote management, WebSM can be used to perform all of the switch management features available from the HMC console. WebSM 5.2.0.10 or later for Microsoft Windows® can be downloaded from your HMC or an AIX server via:

- `http://hmcname/remote_client.html` for the application
- `http://hmcname/remote_client_security.html` for the SSL module

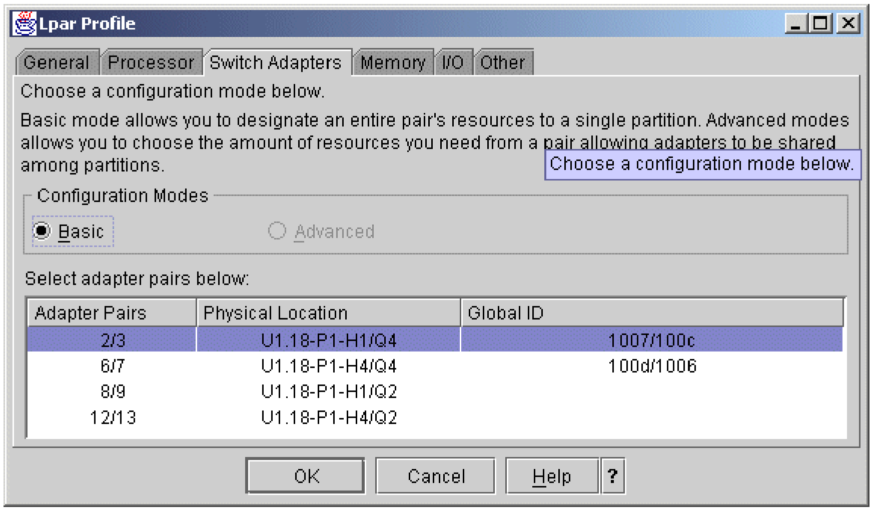
For this to be available from an AIX server, WebSM will need to be installed. WebSM 5.2.0.10 for AIX also works and has been used in the lab. However, we have found no official documentation of support.

2.4.3 LPAR definitions

At this time, link pairs may not be split between LPARs. Even once the single-link converter becomes available, the wrapped link will still be required to be assigned to the same LPAR as its link mate. In the future, there may be support for assigning only a single link to an LPAR but such plans have not been announced.

Tip: It is acceptable to have both switched and non-switched LPARs on the same CEC.

SNI interface allocation is provided through the HMC LPAR definition menus. Rather than showing up under the I/O resources, SNIs have a separate properties tab, as shown in Figure 2-29.



Choose a configuration mode below.

Basic mode allows you to designate an entire pair's resources to a single partition. Advanced modes allows you to choose the amount of resources you need from a pair allowing adapters to be shared among partitions.

Choose a configuration mode below.

Configuration Modes

● Basic ○ Advanced

Select adapter pairs below:

Adapter Pairs	Physical Location	Global ID
2/3	U1.18-P1-H1/Q4	1007/100c
6/7	U1.18-P1-H4/Q4	100d/1006
8/9	U1.18-P1-H1/Q2	
12/13	U1.18-P1-H4/Q2	

Figure 2-29 SNI allocation

Currently, Advanced is not supported and will not be selectable. This is because sharing a link pair between LPARs is currently not supported.

2.4.4 Affinity LPARs

GX buses are I/O resources that will have a natural affinity for specific CPUs. Figure 2-17 on page 30 shows which MCMs are required for which GX buses, and Figure 2-21 on page 33 shows which GX buses are required for each link.

Important: The default assignment of SNIs to Affinity Partitions is incorrect. You must allow the defaults, then go back and change them. If you correct them during the initial definition, your LPARs will fail to start. This is a known defect and will be resolved at some point in the future.

When assigning I/O to Affinity LPARs, always allow the Default SNI configuration. Allow this to continue, then edit the partition to correct the SNI choices, as shown in Table 2-9.

Table 2-9 Proper Affinity Partition SNI allocation

Affinity LPAR[a]	Link 1	Link 2
LPAR 1	2	3
LPAR 2	12	13
LPAR 3	8	9
LPAR 4	6	7

[a] This assumes you are not splitting your MCMs.

2.4.5 Hardware Server

Since the HPS is only connected to the power subsystem, this interface is used to perform all microcode, routing, and monitoring functions. Due to the limits of the CEC RS232 interface, the power system management was moved to the HMC via RS422. This coincides with p655 cabling, since multiple p655 servers can be installed in the same frame, negating the value of a CEC-to-SPCN serial cable as was used for 7040 servers.

Hardware virtualization

Hardware Servers form a distributed, system-wide layer via the service Ethernet. Physical serial port connections to BPAs and CECs are abstracted as virtual serial ports to the Switch Network Manager (SNM) and other HMC software.

All serially attached hardware within the system is assigned a virtual port number that is unique within the entire cluster. This allows all HMCs within the same cluster to access all other hardware by virtual port number regardless of which HMC actually hosts the physical attachment. See Figure 2-30 on page 49 for a logical diagram.

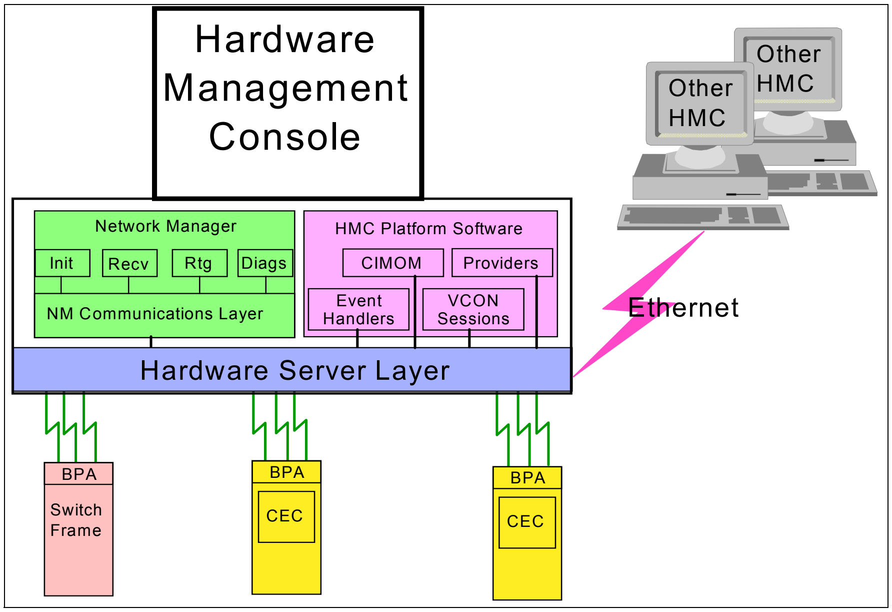


Figure 2-30 Hardware Server network

Note: Single-image (virtual) hardware management *only* affects the SNM and not the HMC's server and partition manager. You still must choose the correct HMC to manage partition resources.

The hrdw_svr daemon communicates on the administrative network via TCP port 8877 for sync and async notifications. This allows the HMCs to communicate about which portions of the HPS network they host, and allows SNM to manage all of these components from any of the HMCs. Clients that use this interface include the Switch Network Manager, the Bulk Power Assembly logger (BPA logger), Microcode Download, the command-line JTAG utility (hmc# jt), and fnm_test.

Logging facilities

The Hardware Server subsystem uses the HMC tracing and logging facilities. HS produces system events with information about serial port communication and peer-to-peer HMC communications. Events can by viewed from the command line using the **showlog** command, or from the HMC GUI by selecting **HMC Maintenance -> System Configuration -> View Console Events**. See Figure 2-31 on page 50 for a logical flow of these components.

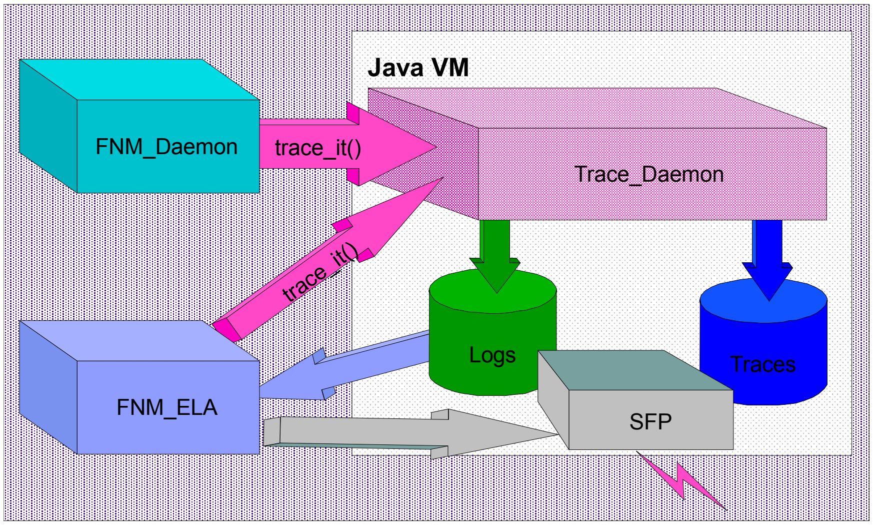


Figure 2-31 HMC FNM logging and diagnostics

Time Of Day Master

A registration list function is provided for use by FNM and BPA logger. These are sorted by uptime, with the longest uptime at the top of the list. The client with the longest uptime becomes the Time Of Day Master, which is functionally equivalent to the switch primary in SP Switches. All other clients function as Time Of Day backups.

2.5 Switch Network Manager

The Switch Network Manager (SNM) resides on the HMC and provides all of the APIs, commands, and administrative functions required of the HPS. Generally this relates to the SNM daemon (fmnd) as referenced in Example 2-2 on page 52.

Note: Any references to FNM, Federation Network Manager, or Fabric Network Manager are references to the Switch Network Manager. Some components are still internally named with an F for Federation.

2.5.1 SNM major components

The major components are shown in Figure 2-32 on page 51. For detailed information regarding SNM administration, see 4.3, “Switch administration” on page 180 for both GUI and command-line examples.

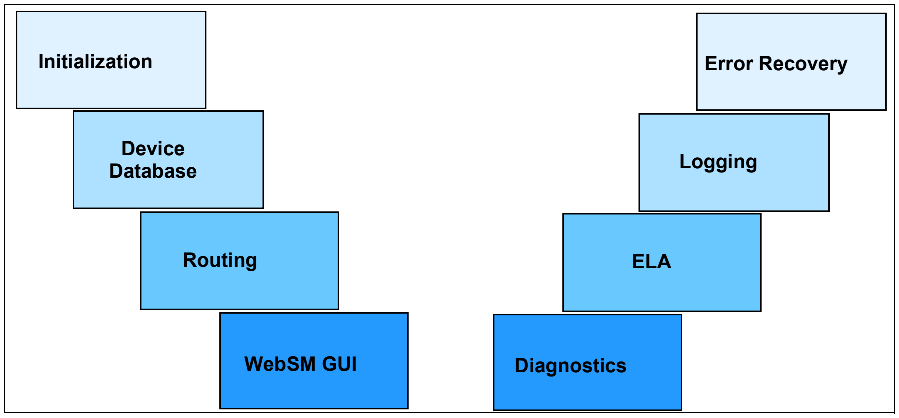


Figure 2-32 SNM overview

2.5.2 HMC SNM filesets

Initially, SNM is comprised of the three RPMs listed in Example 2-1 added to the HMC.

Example 2-1 SNM filesets

```
$ unzip -t update.zip | grep rpm
IBMhsc.SNM-1.1.0.1-1.i386.rpm 10/23/2003 12:56:08 pm 5,205,480 5,035,992 DeflatedN
IBMhsc.SNM_GUI-1.1.0.1-1.i386.rpm 10/23/2003 12:56:08 pm 232,584 228,195 DeflatedN
IBMhsc.ucode-mgr-1.1.0.1-1.i386.rpm 10/23/2003 12:56:08 592,536 586,312 DeflatedN
```

Performance

SNM code consists of C++ with only enough Java™ to tie it into the existing HMC interface. Polling of the switch is offloaded to the DCA-F processor, which will provide alerts back to the Hardware Server, reducing load on the manager. With these performance enhancements, and those made to the HMC CIMOM handling, a single HMC with 32 CECs and full SNM tracing only used 50% of the HMC CPU.

2.5.3 Switch Manager GUI

The Switch Network Manager GUI is accessed through the Switch Management menu on HMC 3.2.4 (or later) as shown in Figure 2-33 on page 52.

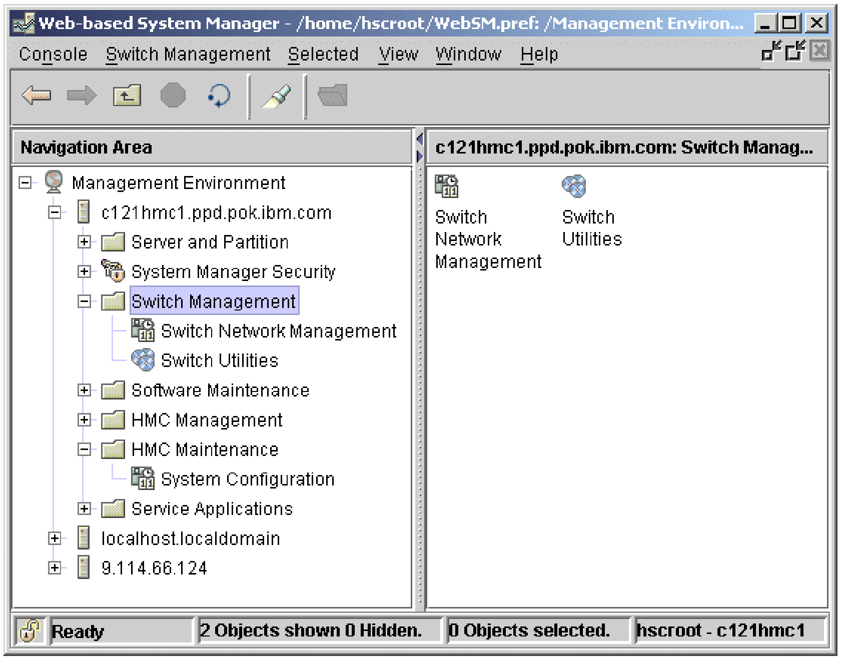


Figure 2-33 Switch management

Switch Network Management GUI submenu

The Switch Network Manager submenu provides most of the standard functions required for administration of the switch, shown in Figure 2-34.

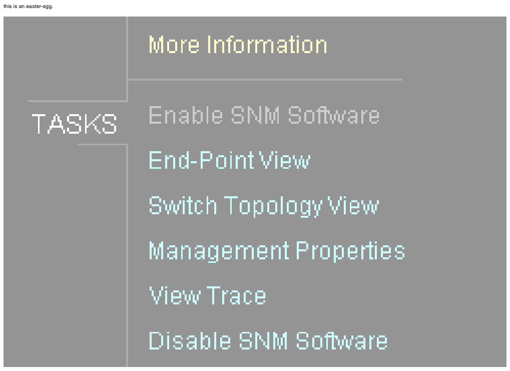


Figure 2-34 SNM task listing

Many of the switch management functions are handled silently and automatically. As such, none of the traditional CSS E-commands exist for the HPS. Most of the commands are display-based and are primarily tabular. The online description of these functions are shown in Example 2-2.

Example 2-2 Text of Switch Network Manager GUI

```
Use the Switch Network Management application to manage the High Performance Switch (HPS)
networks in your switched clusters.

Use the enable SNM Software task to uncomment the entry in /etc/inittab for the Switch
Netwrk Management daemon. This will start the daemon and allow you to use the rest of the
tasks on this panel. After invoking this task, either use the View -> Reload drop-down
selectio or press F5 to refresh this panel in order to enable the other selections.
```

```
To view information about servers, adapters and ports and to perform diagnostic operations
on adapter ports, use the End-point View task.

To power on and off switches, to view information about switches, chips, and links, and to
perform diagnostic operations on switch links, use the Switch Topology View task.

Use the Management Properties task to look at:
   Information about Hardware Management Consoles in your switched cluster
   Summary information about your High Performance Switch (HPS) networks
   Version information about the switch network management software.

To look at trace files from the switch network management software, use the View Traces
task.

Use the Disable SNM Software task to comment out the entry in /etc/inittab for the Switch
Network Management daemon. This will stop the darmon and disable this and other tasks on
this panel. After invoking this taks, either use the View -> Reload drop-down selection or
press F5 to refresh this panel.
```

2.5.4 Switch Network Management CLI

All of the GUI commands are available from the HMC command line. Briefly, the commands are described as follows:

- **lsswtopol**: This provides the same output as the topology view in the GUI.
- **verifylink**: Concurrent verification of a link.
- **testlinecont**: Test link continuity.
- **chswpower**: Power on and off switches.
- **lsswenvir**: Display switch environmentals.
- **lsswmanprop**: Display SNM properties.
- **lsswtrace**: View SNM trace log.
- **chswnm**: Enable and disable SNM.

For full details, see 4.3, “Switch administration” on page 180.

2.5.5 Switch network manager (fnmd) daemon

The fnmd (Federation Network Manager Daemon) relies on the Hardware Server for communication. The Hardware Server provides program access to the serial lines of the CEC and the BPA. The RS232 is used to communicate with the Common Service Processor (CSP) within the CEC. This communication channel is used to access the JTAG interface of the Switch Network Interface (SNI).

The RS422 is used to communicate with the power subsystem of the Regatta class machine. The power subsystem connects to the switch Distributed Converter Assembly (DCA-F) via Universal Power Interface Cables (UPICs). UPIC cables carry both power and an embedded RS485 (multi-point serial network), which provides access to power commands, environmental information, and the service processor of the HPS. This is diagrammed in Figure 2-35 on page 54.

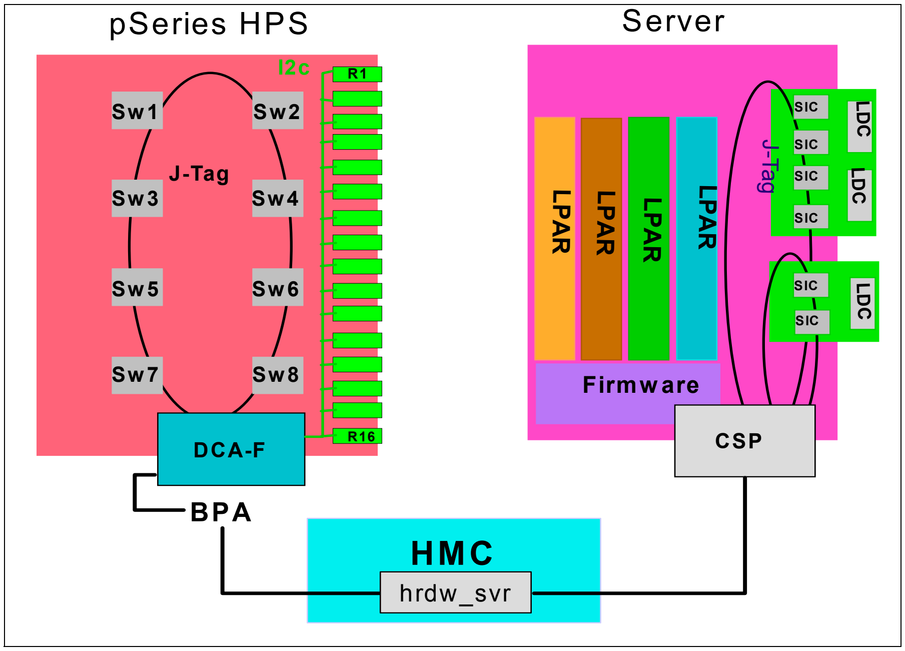


Figure 2-35 JTAG access to HPS components

> **Note:** The fnmd and Hypervisor functions replace the equivalent functions of the fault service worm from PSSP's CSS.

The HMC as fabric manager

Currently, fnmd and hrdw_svr are restricted to running on an HMC. However, that may not be a permanent condition. There has been talk of allowing these packages to be migrated to a CSM master. In such a situation, there would likely exist conditions in which the HMC itself was no longer needed. Until such time, client access for CIMOM,low-level commands (llcmd), and vterm sessions continue to be provided by hdwr_svr through TCP port 9734.

The most important time for the fabric manager is during initialization. Once all of the nodes are initialized and brought onto the switch network, the fabric manager can be brought down. Recovery operations will fail while the fabric manager is down. However, reboots should not leave a node off of the switch.

In other words, once running, the switch can continue should the HMC fail.

> **Important:** Recovery is required for an MP_DOWN or MP_FAIL error. Five such errors logged against the same link will render that link fenced. Should a link become fenced, the only recovery will be to logically power cycle the CEC.

2.6 FNM communication

FNM communication is the API used by the Federation Network Manager daemon (fnmd) to perform all operations required for management of the pSeries HPS. Multiple FNM daemons will be running on HMC. This is normal and is to facilitate the various functions of FNM communication. The various commands are issued through hrdw_svr as shown in Figure 2-36 on page 55.

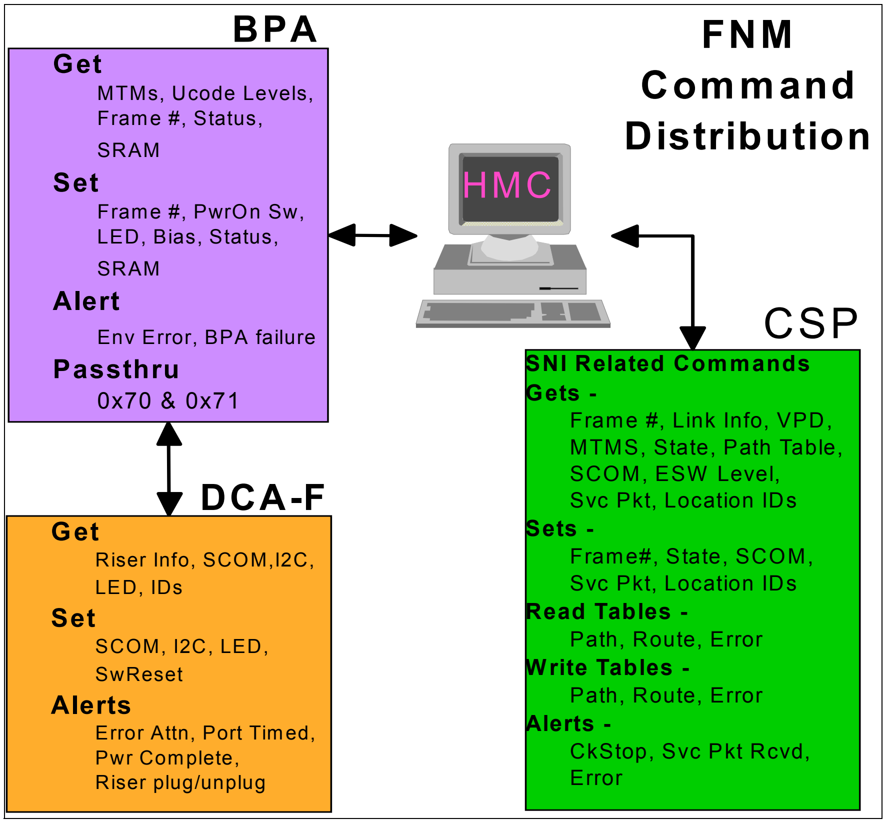

BPA
Get
MTMs, Ucode Levels, Frame #, Status, SRAM
Set
Frame #, PwrOn Sw, LED, Bias, Status, SRAM
Alert
Env Error, BPA failure
Passthru
0x70 & 0x71
HMC
FNM Command Distribution
CSP
SNI Related Commands
Gets -
Frame #, Link Info, VPD, MTMS, State, Path Table, SCOM, ESW Level, Svc Pkt, Location IDs
Sets -
Frame#, State, SCOM, Svc Pkt, Location IDs
Read Tables -
Path, Route, Error
Write Tables -
Path, Route, Error
Alerts -
CkStop, Svc Pkt Rcvd, Error
DCA-F
Get
Riser Info, SCOM,I2C, LED, IDs
Set
SCOM, I2C, LED, SwReset
Alerts
Error Attn, Port Timed, Pwr Complete, Riser plug/unplug

Figure 2-36 FNM command distribution

FNM_Comm operates within four threads: FNM_Init, FNM_Recv, FNM_Rtg, and FNM_Diag, all shown in Figure 2-37 on page 56. There is an external interface as shown. However, very little information is available about its use. All switch functions are performed through these interfaces.

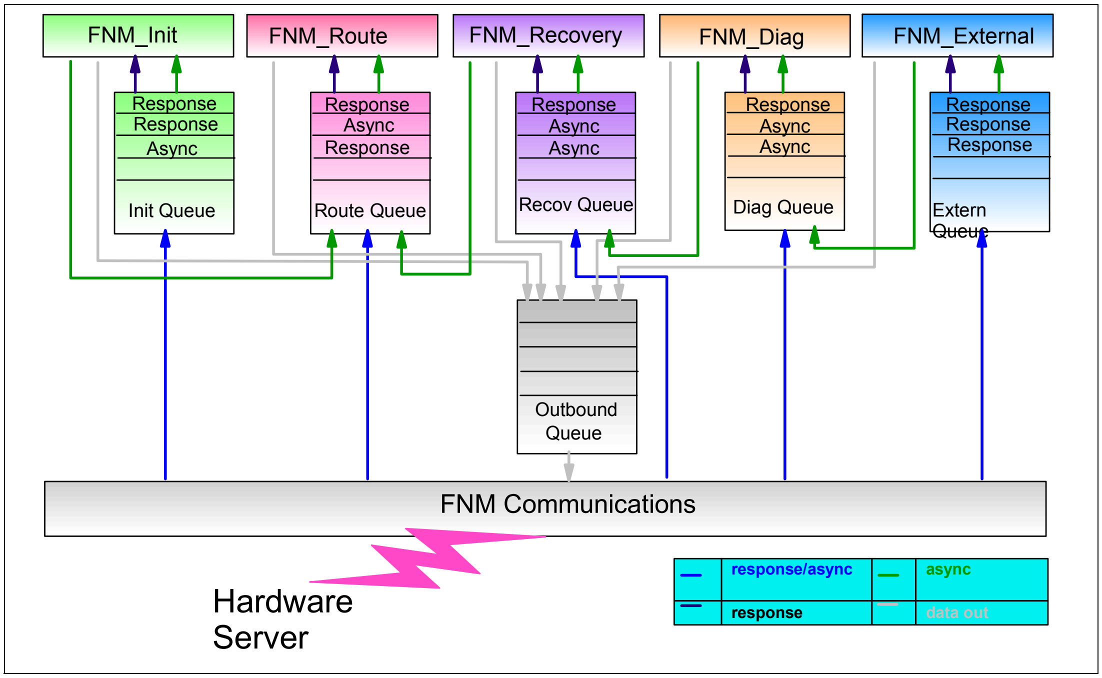


Figure 2-37 Fabric Network Manager functions

2.6.1 Time of Day (TOD)

Time of Day is crucial to the internal synchronization of switch master and backups. The master TOD server is equivalent to the switch primary and is chosen by which CEC has been up the longest. For performance reasons, it may be beneficial to choose a low-load or higher-speed system, since synchronization is maintained within device drivers and firmware.

All other CECs are Time of Day slaves. This is equivalent to a switch primary backup. Time of Day synchronization is maintained within the HPS drivers. No user intervention is required and this will be synchronized on initialization.

2.6.2 Switch data structures

FNM sets up data in flash memory in the CEC as a series of tables. These CSP tables, also known as LIDs, are used by the CSP to bring up the adapters during basic initial self test (BIST) prior to LPAR Ready. This data includes the Path and Route tables, and a set of SCOM instructions in the MP_Regs LID.

Network table

This is a list of the networks that currently exist and relevant information. Presently, this consists of one entry per switch.

MSS list

This keeps track of the current set of TOD masters and backups.

Device table

This is a table of all devices physically attached to the HPS network. Specifically, this contains one entry per switch chip and one entry per SNI.

Link table

This is a table containing every switch chip port in the network. Each switch board contains 64 and each SNI chip contains two. This is used by the endpoint map.

Endpoint map

This maintains the mapping between each Switch Network Interface's endpoint ID and its current switch connection number.

2.6.3 FNM message data types

Each object has an ACK list to keep track of outstanding messages, plus state data pertinent to the task for which the object was created. When the task is complete, the object is removed from the list. The data types used for these tasks are:

- SwitchInit: Switch initialization message traffic and state data
- CSPInit: CEC/SMA initialization message traffic and state data
- SwitchEvent: Created for:
  - Switch Link Up asynchronous event
  - Riser Status Change asynchronous event
  - Setting switch send enables
- SMAEvent: Created if Switch Link Up turns out to be for a node port
- VPDget: VPD collection message traffic

2.6.4 CEC states that CSP makes visible to FNM

During SNM operations, the CEC presents its IPL state via asynchronous notification to fnmd if not directly queried by a call to GetExtendedSystemInfo. The valid states are:

- Pre-IPL

  This indicates that standby power is on, or in other words, the system is physically powered on, but logically powered off.

- SMAs Configured

  This state occurs once CSP has investigated the SMAs enough to allow FNM to issue the GetSmaLinkInfo query.

- Download

  When the “download window” is open, FNM can read and write flash.

- TablesLocked or Post-Download

  Once the “download window” is closed, FNM cannot access flash.

- IPL_Ready

  This state allows FNM to access flash and to read and write adapter registers. This is the point at which partitions can be activated.

2.6.5 FNM message types

FNM and CEC have many message types used to perform initialization. These are broken into groups based on their associated functions as follows:

- FNM status messages
  - FN_NEWSSP: Switch detected by FNM_Comm

  - FN_NEWCSP: CEC detected by FNM_Comm
  - FN_CHANGETOSNM: FNM personality change
  - FN_FRUINFO: FRU info for VPD gathering
  - FN_FNMEXIT: Error, FNM is exiting
  - FN_MSSLISTCHANGE: State change for a member of the MSS list
  - FN_CHANGE_UCODE: Microcode download

- DCA status messages

  - AD_POWERONSEQUENCECOMPLETE: DCA completed PowerOn sequence
  - AD_MASTERDCAACTIVATED: New DCA master selected
  - AD_SWITCHLINKUP: DCA detected a switch link newly timed
  - AD_RISERUNPLUG: DCA detected a riser status change

- DCA response messages

  - DCA_ASYNCACK: DCA response to async
  - DCA_SWITCHPOWERSTATUS: DCA response to query Switch Power Status
  - DCA_RISERINVENTORY: DCA response to riser status query
  - DCA_SWJTAGREAD: DCA response to switch jtag read
  - DCA_SWJTAGWRITE: DCA response to switch jtag write (uniform register values)
  - DCA_SWJTAGMULTI: DCA response to switch jtag write (nonuniform register values)

- BPA response messages

  - BPA_READEEPROM: BPA response to read VPD
  - BPA_GETAVAILABLEFRUS: BPA response to FRU query

- CEC status messages

  - AC_SMACONFIGURED: CSP signal to investigate SMAs
  - AC_DOWNLOADBEGIN: CSP indication that flash access is permitted
  - AC_DOWNLOADEND: CSP indication that flash access is not permitted
  - AC_IPLREADY: CSP indication that CEC has reached LPAR banner
  - AC_PREDOWNLOAD: CSP indication that CEC must re-IPL before being used

- CEC response messages

  - CSP_GETSYSINFO: Response to get CEC state query
  - CSP_SETIPLDELAY: Response to command to set the IPL delay
  - CSP_GETSMALINKINFO: Response to get SMA info query
  - CSP_ENDDOWNLOAD: Response to “`FNM has finished download`”
  - CSP_BEGINDOWNLOAD: Response to “`FNM will use the download window`”
  - CSP_GETSMAVPD: Response to SMA VPD read
  - CSP_GETCECMTMS: Response to CEC MTMS read
  - CSP_ACCESSSCOMREG: Response to SCOM reads and/or writes
  - CSP_OPENREADTABLE: Response to command to open a CSP table for read
  - CSP_OPENWRITETABLE: Response to command to open a CSP table for write
  - CSP_READTBL: Response to command to read a CSP table
  - CSP_WRITETBL: Response to command to write a CSP table
  - CSP_CLOSETBL: Response to command to close a CSP table
  - CSP_GETTRACELOG: Response to command to read the HCA trace log
  - CSP_CLEARNOTE: Response to clear async notification (async ACK)

2.6.6 FNM_Init

FNM_Init is one of the major threads used within FNM communications. Its purpose is to initialize the switch fabric upon command. The FNM_Init performs the functions described in Figure 2-38 on page 59.

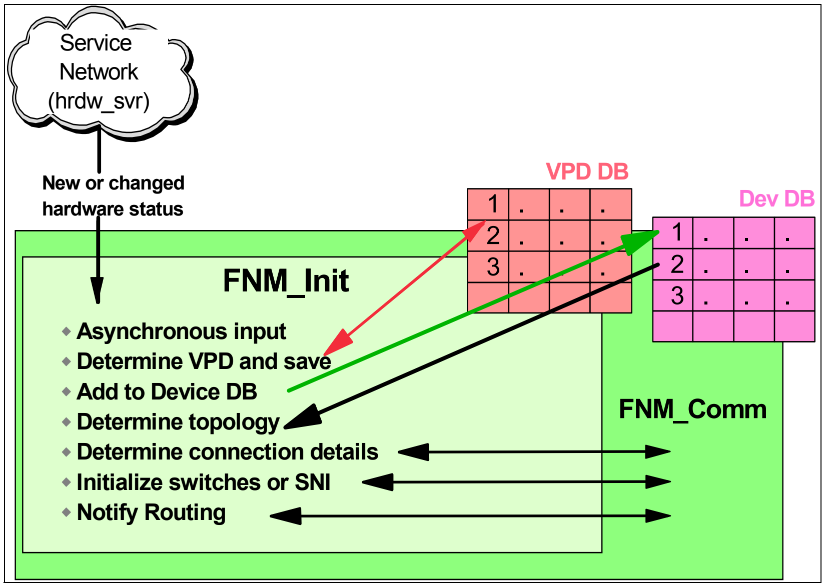


Figure 2-38 FNM_Init logical description

Preconditions for switch initialization

- Link lights should be solid on all ports on all switches and on all SNIs.
- MP_AVAILABLE status means the adapters are ready for communication.
- Time of Day must be synchronized and is done so by the FNM daemon on the HMC during startup.

FNM_Init operations

When the thread is initially started, it is set to wait for asynchronous notification and enters an infinite loop. This loop performs the following actions based on the state specified:

- WAIT_FOR_NOTIFICATION
  - Check for a new message.
  - Process the new message.
  - Update the state if required.
  - Continue the loop.
- DISCOVERY

  This state tells FNM_Init to gather switch connection information in order to determine switch topology. The basic steps are:

  a. Skip to the end if there are still pending switch connection information messages.
  b. Read VPD from hardware to determine connected components.
  c. Write status information to VPD table.
  d. Build the device DB and mark it Ready.
  e. Determine the topology from connection information.
  f. Initialize any switches or SNIs in need.
  g. Notify FNM_Route to generate ideal routes for the topology and perform download to firmware if required.
  h. Make TOD master and backup assignments to adapters and configure SNI registers to match.

  Note: fnmd will die and respawn if TOD cannot be synchronized among the frames.

  i. Change FNM_Init state to PROCESS_SMAs.

- PROCESS_SMAs

  This state is used while FNM_Init gathers information about the Switch Network Interfaces. The basic steps are:

  a. Skip to the end if we are waiting for SMA info.
  b. Recognize the topology. Determine the switch planes. Determine the switch networks.
  c. Assign location ID and endpoint ID to each functional SNI.
  d. Enable switch node port sending.
  e. Update the Adapter Initialization tables (MP_Regs).
  f. Update the switch location IDs.
  g. Change FNM_Init state to OPERATIONAL.

- OPERATIONAL

  When initialization is essentially finished and FNM_Init waits for asynchronous events. These events may be one of the following:

  – Add to or update the MSS list.
  – Enable node port sending.
  – Update adapter Initialization tables.
  – Update adapter microcode.
  – Bring in a late-arriving SNI.

2.6.7 FNM_Route

The route table is equivalent to the annotated topology of an SP Switch. It is propagated from the fnmd on the HMC to the SNIs and switches through Hardware Server. The HPS routing table is stored within the switch and the SNI device databases. Updates to the routes are performed by the Network Manager. When FNM_Route is called, the functional flow is as displayed in Figure 2-39.

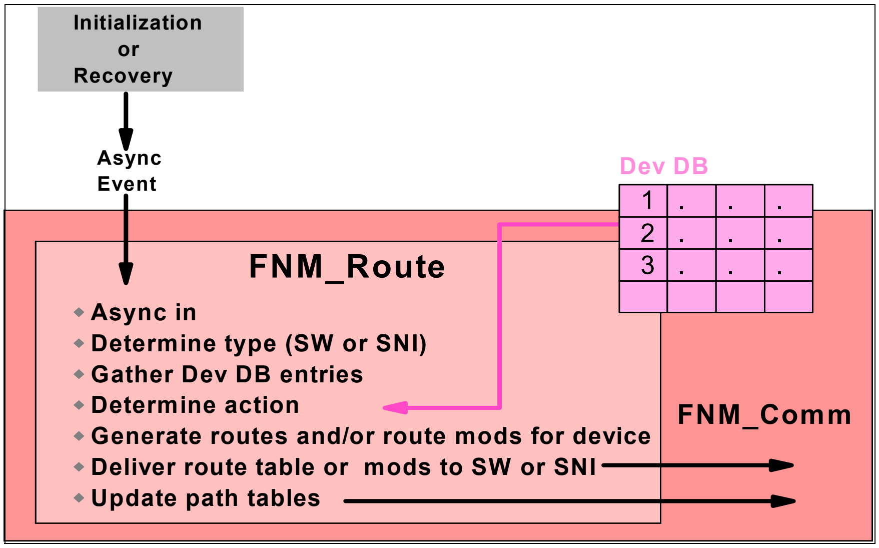


Figure 2-39 HPS routing overview

FNM_Route operations

FNM_Route is important during initialization and recovery. The FNM utilizes the following subroutines in the execution of route table processing:

- FN_GENERATEIDEALROUTES

  Set up routes between all pairs of endpoints in the network and generate multicast table entries for the switch.

  Route generation class methods handle the route table and path table generation. Some of the important methods of this class are described in Table 2-10.

Table 2-10 Route generation classes of importance

Route generation class	Data objects used
RouteGeneration	topology
needNewRoutes	network
generateAdapterRoutes	location ID
modifyAdapterRoutes	location IDs of source and destination
update AdapterRoutes	location ID
generateEndpointRtTbl	route table
generatePathTable	source endpoint ID

- FN_MLTDOWNLOAD

  Download the multicast route tables into the switch. The general flow is as follows:

  a. Issue an *open table* packet to the CSP.
  b. Read each route table.
  c. Read each route to be loaded.
  d. Build the route updates.
  e. Write the updates to the CSP.

- DCA_SWJTAGMULTI

  Notify the switch about the status of the new download. This is performed through the BPA and is functionally similar to FN_MLTDOWNLOAD.

- FN_SMALINKUP

  Used for late SNI initialization or SNI recabling to set up the information to handle the new SNI and add it to mismatched SNIs. The procedural flow is shown in Example 2-3.

Example 2-3 SMA link exception handling

```
list if it does not show up at the expected switch port
generate Adapter Routes
generate Path Table
if the CEC is IPLready
   begin a download window
   queue the request for opening the window
if the download window is open
   log and continue to download
generate and download any deltas that are necessary
```

- FN_SMANEEDSROUTES

  Processes mismatched or unexpected links during route table bringup as shown in Example 2-4.

Example 2-4 SNI path error flow and needs routes

```
Parse the incoming error packet
If the error is due to network fault
   If routes can be repaired
      If alternate routes are available
         generate new routes
Set message flag to disable or re-enable error reporting as appropriate
Requeue the message to recovery
Download any alternate routes that were generated to SNIs.
```

- FN_SMAPATHERR

  Used to process routing faults via the same logic as used in FN_SMANEEDSROUTES.

Routing interfaces

Switch Network Manager uses i_stub interface to upload and download paths and routes. To aid in troubleshooting failed communication between adapters, /opt/hsc/bin/i_stub_FS on the HMC can be used to dump routes and paths, or fnm.SNAP can be used to gather more information.

2.6.8 FNM_Recovery

FNM_Recovery is another of the autonomous functions of FNM communications. On a link failure, FNM recovery is notified. Depending on the type of failure, FNM Recovery may bring the link back into service with a short delay of only five seconds, or it may mark the link permanently down. Once action is taken, any appropriate routing changes are sent to FNM_Route for processing. See Figure 2-40 for the general flow.

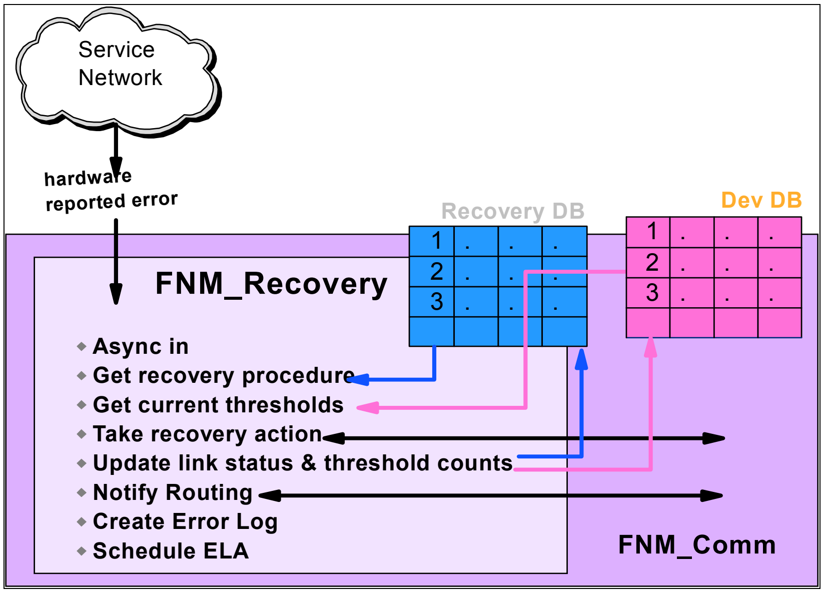


Figure 2-40 FNM Recovery flow

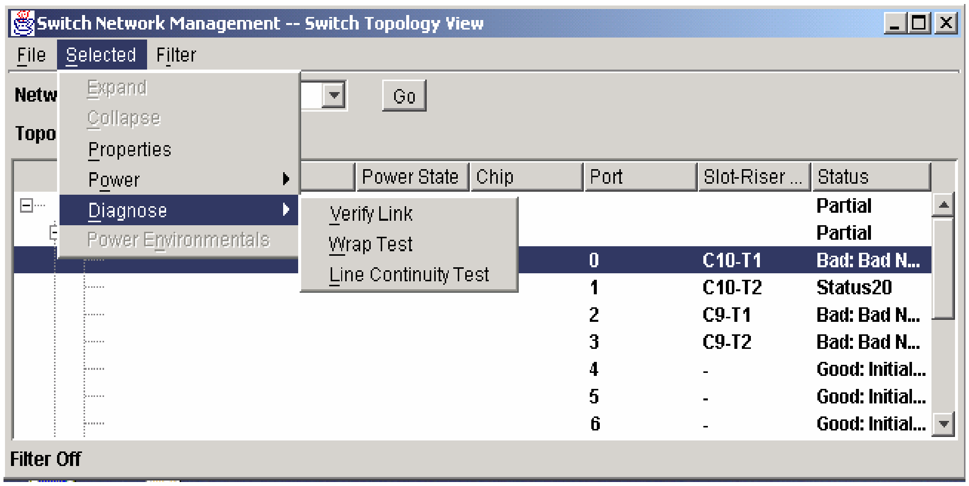
Important: Once a link has registered five MP_DOWN or MP_FAIL events, that link will remain down until the CEC is logically power cycled. Rebooting the LPAR is not sufficient to recover from this.

More details on monitoring for error events are provided in 2.7, “Operating systems” on page 65.

2.6.9 FNM_Diag

FNM diagnostic routines are used to access environmental information, alarms, and chip level statistics. These are primarily called from the HMC command-line utilities residing in /usr/local/hcctool. The primary goals of SNM diagnostics are to isolate faulty components, verify hardware installation, and to collect product information.

Upon errors, SNM diagnostics should always be run regardless of what FRU is listed in serviceable events. Diagnostics should be performed on the SNI first, then on the switch. Diagnostics will report results by switch chip, frame, slot, and adapter numbers. Each of the diagnostic tools will be explained from a command-line standpoint. However, most of them can also be accessed from the SNM GUI topology view, shown in Figure 2-41.



Power State	Chip	Port	Slot-Riser ...	Status
				Partial
				Partial
		0	C10-T1	Bad: Bad N...
		1	C10-T2	Status20
		2	C9-T1	Bad: Bad N...
		3	C9-T2	Bad: Bad N...
		4	-	Good: Initial...
		5	-	Good: Initial...
		6	-	Good: Initial...

Figure 2-41 GUI diagnostics access

Link verify test

The first test, a link verify test, is called with **`verifylink`**. This is a non-intrusive command that may be run concurrently with other network traffic. It will test switch chips, switch risers, and the SNIs. Switch chip registers are checked for problem flags, and riser cards are verified for inconsistent signaling, which indicates improper seating of the card.

Line continuity test

The line continuity test should be used if the link verify test does not help. A line continuity test may be issued for a single link, a set of links, or the entire topology. In addition to the link verify tests, the line continuity test uses service packets to test that each segment of a link can transmit data reliably. The line continuity test interface is provided by **`testlinecont`**.

Important: This test is intrusive and will disrupt network traffic on tested components.

Wrap test

The wrap test requires the FC 3756 Network Diagnostic Tool Kit. This is a user interactive test that focuses on an individual switch link. The user will be required to unplug cables and riser cards and to insert a wrap plug into individual ports. The objective of this test is to isolate faulty components.

hps_check.pl

This script resides in /usr/local/hsctool on the HMC and queries Time of Day and most of the other switch registers. It is a good tool to make sure that FNM_Comm is working well and that all nodes see the same TOD master. An example of its output is shown in Example 2-5. Some of the text has been reduced in order to prevent line wrap.

Example 2-5 hps_check.pl output

```
-----------------------------------------------------------------------------------------------------------
VPORT 1:      fffffe01  CEC_NAME: itso_p690                   CEC_MTMS:  7040-681 022BE2A
VPORT 2:      --------  HMC_REG:  1eff  POLL_FREQ:   45       CEC_STATE: (01) CEC is IPL_READY.
FRAME:     1  CAGE:  0  FNM_REG:  0100  HMC_CONN:    01       OP_PANEL:  LPAR...

SNI Mapping:                                                        Switch Neighbor:
Lpar Name                  Lpar# Sni# => Adapter#  Csp#  Cronus# => Frame Cage Chip Port : Timed?  MPA  TOD
                               1    0           2     6       13        1    3    7    1      YES  YES  SLV
                               1    1           3     7       12        1    3    5    2      YES  YES  SLV
                               2    0           0     2        5        1    3    5    3      YES  YES  SLV
                               2    1           1     3        4        1    3    7    0      YES  YES  MAS

Mapping:                Neighbor:        Summary:       Registers:
PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TIMED     0x24030  MP_AVAIL   0x6050  TOD       0x20000
YY   0   2   5   1    7    1   3   5   3   YES YES SLV  00000000 00000000  10000000 00000000  81fd4f98 35eb72b8
YY   1   3   4   1   12    1   3   7   0   YES YES MAS  00000000 00000000  10000000 00000000  81fd4f98 39649fe8
YY   2   6  13   1   13    1   3   7   1   YES YES SLV  00000000 00000000  10000000 00000000  81fd4f98 3d8fa278
YY   3   7  12   1    6    1   3   5   2   YES YES SLV  00000000 00000000  10000000 00000000  81fd4f98 410af8ce
YN   4   8   6   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   5   9   7   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   6  12  14   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   7  13  15   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  PHYS_ID   0x2B000  NEIGH_ID  0x23020  WHO_MAST  0x21040
YY   0   2   5   1    7    1   3   5   3   YES YES SLV  10070000 00000000  04000007 20001035  10000000 00000000
YY   1   3   4   1   12    1   3   7   0   YES YES MAS  100c0000 00000000  04000001 20001037  10000000 00000000
YY   2   6  13   1   13    1   3   7   1   YES YES SLV  100d0000 00000000  04000003 20001037  10000000 00000000
YY   3   7  12   1    6    1   3   5   2   YES YES SLV  10060000 00000000  04000005 20001035  10000000 00000000
YN   4   8   6   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   5   9   7   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   6  12  14   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   7  13  15   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TOD_GEN   0x21030  TOD_MAST  0x21000  TOD_BACK  0x21010
YY   0   2   5   1    7    1   3   5   3   YES YES SLV  0070e000 00000000  00000000 00000000  00000000 00000000
YY   1   3   4   1   12    1   3   7   0   YES YES MAS  1070e000 00000000  80000000 00000000  00000000 00000000
YY   2   6  13   1   13    1   3   7   1   YES YES SLV  0070e000 00000000  00000000 00000000  00000000 00000000
YY   3   7  12   1    6    1   3   5   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000 00000000
YN   4   8   6   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   5   9   7   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   6  12  14   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------
YN   7  13  15   -    -    -   -   -   -     -   -   -  -------- --------  -------- --------  -------- --------


This snap was generated Thu Nov  6 04:50:49 2003 on hmcitso.itso.ibm.com.
```

fnm_test

This is a low-level utility included with hps_check.pl that connects to Hardware Server. It is used by hps_check.pl and generally should not be called by users. It is a binary file and will not yield much useful information by itself.

/var/hsc/log

The text files in /var/hsc/log on the HMC contain information on # of CECs and switches. For more details, see the manual *eServer Hardware Management Console for pSeries Maintenance Guide*, SA38-0603.

2.7 Operating systems

Only AIX 5.2.0.0-ML02 with APAR IY48488 and later provides support for the pSeries HPS. This may also be referenced as 5.2B PTF 4 (and should contain the fileset bos.mp 5.2.0.14 or later).

NIM

NIM now includes a feature known as Secondary NIM adapter. This allows NIM to configure additional adapters in order to replace the features that PSSP provided. At the time of writing this redbook, NIM did not have the ODM classes available to handle a switch network devices. They can be configured manually. NIM customization will not configure them. This should be resolved in the next PTF.

E-commands

There are no more E-commands and there is no fault service daemon or rc.switch. As such, regular AIX **ifconfig** and **smitty chdev** commands can be used to bring up and down the sni# interfaces. This requires both protocol devices within a link pair to be sequentially numbered on the same subnet. If a second plane is desired, a separate link pair must be assigned.

Linux

There is no support for the HPS on Linux for pSeries at this time.

Booting

During boot, the LPAR virtual LCD should show B00E during the GX test of the SNI. This will take much longer than prior to SNI installation and is perfectly normal.

2.7.1 Devices

Links within an SNI, link devices within AIX, and protocol devices within AIX are all managed by the HPS device drivers.

The switch drivers are now a part of AIX and are included in the filesets shown in Example 2-6.

Example 2-6 SNI driver filesets for V1.1.0.1

```
# installp -ld .
devices.common.IBM.sni.rte 1.1.0.1 Switch Network Interface Runtime
devices.common.IBM.sni.ml 1.1.0.1 Multi Link Interface Runtime
devices.common.IBM.sni.ntbl 1.1.0.1 etwork Table Runtime
devices.chrp.IBM.HPS.rte 1.1.0.1 IBM eServer pSeries High Performance Switch (HPS) Runtime
devices.chrp.IBM.HPS.hpsfe 1.1.0.1 IBM pSeries HPS Functional Exerciser
devices.msg.en_US.common.IBM.sni.rte 1.1.0.1 Switch Network Interface Rte Messages
devices.msg.en_US.common.IBM.sni.ml 1.1.0.1 Multi Link Interface Runtime
devices.msg.en_US.common.IBM.sni.ntbl 1.1.0.1 Network Table Runtime Messages
devices.msg.en_US.chrp.IBM.HPS.rte 1.1.0.1 pSeries HPS runtime Messages
devices.msg.en_US.chrp.IBM.HPS.hpsfe 1.1.0.1 pSeries HPS functional utility
```

Updates to AIX filesets can generally be found at:

https://techsupport.services.ibm.com/server/aix.fdc

Except in cases of hardware failure prior to initial **cfgmgr**, both ports of a link pair will be sequentially numbered starting from sni0 and sni1.

Each link must have its own IP address. It is recommended that SNI links within a pair have matched IP addresses.

For details on configuration of these adapters, see 4.2.7, “Installing and configuring the SNIs in AIX” on page 170.

2.7.2 Common issues and diagnosis

Aside from the Switch Network Manager described in “Switch Network Management GUI submenu” on page 52, the primary tools for HPS diagnostics are the AIX **errpt** and the RSCT class IBM.ServiceEvent. Both of these will show MP_DOWN and MP_FAIL events. In addition, you may use **lssrc -ls nrd** to gain overall statistics of MP_DOWN events per interface.

Resource Monitoring and Control (RMC)

In an HPS environment, the following information should assist you in diagnosing the most common problems with RMC.

ConfigRM related

- **lsrpdomain** shows domain is not active.
- **lsrpnode** shows the node as not online.
- IBM.NetworkInterface has not harvested the adapter information yet.

HATS/HAGS related

- **lssrc -ls cthats** shows adapter as down.
- **lssrc -ls cthags** or **hagsns -s cthags** shows HAGS domain not established.
- **havsvote -ls cthags** shows voting has not finished.

HAGSGLSM related

- **lssrc -ls cthagsglsm** shows membership not formed.

Places to look

- errpt
- RMC trace file: # rpttr -o dtic /var/ct/IW/log/mc/trace
- ConfigRM trace file: # rpttr -o dtic /var/IW/log/mc/IBM.ConfigRM/trace
- HATS log files: /var/ct/<RPD name>/log/cthats
- HAGS and HAGSGLSM log files: /var/ct/<RPD name>/log/cthags
- commands: **lsrpnode, lsrsrc IBM.NetworkInterface**, etc.
- ctsnap

MP_DOWN

MP_DOWN events can be monitored through RMC conditions and responses.

In addition, Service Focal Point, properly configured, will show MP_DOWN and MP_FAIL errors as well as any permanent hardware failure.

For an easy check from a command line, IBM.ServiceEvent can be queried directly as per Example 2-7.

Example 2-7 lsrsrc IBM.ServiceEvent

```
#> lsrsrc IBM.ServiceEvent
resource ##:
        ChangeCounter     = 5698
        HSCId             = ""
        HSCName           = "233DX3M"
        Status            = "Open"
        ErrLogLabel       = "EPOW_RES_CHRP"
        CallHomeCandidate = "No"
        CalledHome        = 0
        CEComments        = {}
        CECommentsNew     = []
        FRUList           = {}
        FRUListAdditional = {}
        FRUListNew        = []
        PtrHscEED         = "
        Nodes             = {}
        ComponentID       = ""
        SpecialHandling   = ""
        NodeNameList      = {"hmcitso.itso.ibm.com"}
```

No Recovery daemon (NRD)

The NRD daemon provides information about SNI status. See Example 2-8.

Example 2-8 lssrc -ls nrd output

```
LPAR# lssrc -ls nrd
Subsystem         Group            PID          Status
 nrd              nrd              196776       active

 Subsystem started: Fri Oct 31 11:59:58 2003

 Number of requests since startup:
 stops ------------------> 0
 cancels ----------------> 0
 refresh ----------------> 0
 status -----------------> 1
 traceson ---------------> 0
 tracesoff --------------> 0
 current trace level -> 2 (all messages, default value)
 SRC socket -------------> 0x00000005
 nrd socket -------------> 0x00000006
 nrd port ---------------> 0x0000dfdf
 status_counters.rejected -----------------> 0x00000000
 status_counters.total.handled -----------> 0x00000000
 status_counters.total.ch_fail_start -----> 0x00000000
 status_counters.total.ch_fail_complete --> 0x00000000
 status_counters.total.ch_other ----------> 0x00000000
 status_counters.total.ch_stop -----------> 0x00000000
 status_counters.total.mp_down -----------> 0x00000000
 status_counters.total.sw_mp_down --------> 0x00000000
 status_counters.total.mp_up -------------> 0x00000000
 status_counters.total.mp_dump_req -------> 0x00000000
 status_counters.total.tce_lmt -----------> 0x00000000
 status_counters.total.zc ----------------> 0x00000000
 status_counters.total.dropped -----------> 0x00000000
 status_counters.total.invalid_req -------> 0x00000000

 Trace file path: /var/adm/sni/nrd.trace
```

AIX error report

Of course, it just wouldn't be complete without an entry in the AIX error report. MP_DOWN errors show up as MP_DOWN in the AIX errpt.

2.7.3 AIX tunables

In order to use the switch, certain virtual memory parameters (VMM) must be changed. On a system with 16 GB RAM, we used the command and parameters shown in Example 2-9.

Example 2-9 vmo command line

```
# vmo -r -o lgpg_size=16777216 -o lgpg_regions=64 -p
```

This change, with the `-p` option, becomes persistent across reboots. NIM additional adapter customization performs this step for you.

Note: You need to reboot for the changes to take effect.

The following parameters of the SNI interface are optional but should be considered for customer tuning environments. For SNI, the ODM attributes specifying bounds on window sizes and memory usage are as follows:

win_poolsize	Total pinned node memory available for all user-space receive FIFOs
win_maxsize	Maximum memory used per window
win_minsize	Minimum memory used per window

You can change them using the **`chgsni`** command. For further details related to the **`chgsni`** command and the recommended values, consult *Switch Network Interface for eServer pSeries HPS Guide and Reference*, SC23-4869.

This is another easter-egg.

Note: You do not need to reboot your machine after making changes to these attributes.

See 3.4.13, "VSD configuration support for the SDD" on page 111 for more details on tuning.

2.7.4 Cluster Systems Manager

To use the SP Switch you need PSSP. To use the HPS, CSM is not needed. CSM is recommended primarily to assist in managing multiple nodes and to assist in building NIM node objects. All of these features can be bypassed. For example, you could manually configure NIM and use RSCT Resource Monitoring and Control.

Also, many of the new filesets for the HPS, including the device drivers and LAPI cannot be installed concurrently with PSSP filesets. Due to limitations of the control workstation in frame management and NIM management, PSSP is not supported for use in an HPS cluster. It is acceptable for PSSP and HPS clusters to use the same HMC, just not the same CEC.

That having been said, we recommend using CSM to manage your HPS cluster. CSM provides a management interface that replaces many of the functions of a PSSP control workstation. Many of these functions are standard UNIX utilities rather than the customized tools from PSSP.

Benefits of CSM

- Install and update machines remotely.

  CSM, now in Version 1.3.2, provides the ability to define your NIM node objects based on your CSM node definitions.
- Remotely power on, off and reboot of nodes in the cluster.
- Continuous monitoring through RMC for predefined or user-defined conditions.

  Automated responses can be run any time a problem occurs and can provide notification or take corrective action.
- Files can be changed in one location and distributed to all the machines or a set of machines in the cluster. This saves time and effort in trying to keep configuration and utilities synchronized between multiple systems.
- Use software maintenance features of CSM to install and update software automatically and remotely without having to log in to each node in the cluster.
- CSM supports AIX and Linux on pSeries servers and Linux on xSeries. The limited integration with IBM Director is planned to grow into a more unified systems management interface.

Note: At this time, CSM does not have `rpower` ability against the HPS itself. The switch can be power cycled from the Switch Network Manager as described in 4.3.1, “Power commands” on page 180.

CSM clusters

CSM can manage machines across different hardware domains, different switch topologies, and across high availability clusters. See Figure 2-42 on page 70 for an example of a cluster.

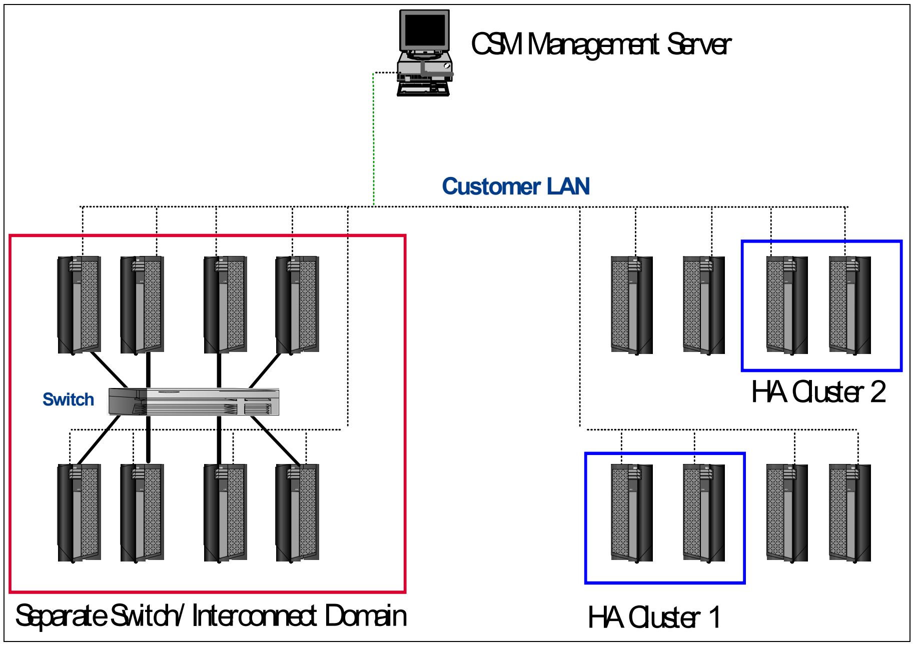


Figure 2-42 CSM management diagram

When configuring the switched systems as part of a Cluster 1600 management domain take into consideration the maximum number of the systems allowed in a CSM cluster. During its initial offering, the pSeries HPS will provide much greater restrictions on Cluster 1600 sizing. Refer to Table 2-11 for more details.

Table 2-11 HPS reference information

HPS reference information	**7040**	**7039**
Servers per Cluster 1600	32	64
HPS Servers per Cluster 1600	16[a]	16[a]
LPARs per Server	16	1
HPS LPARs per server	4	1
LPARs per Cluster 1600	128	
HPS LPARs per Cluster 1600	16	16
HPS Links per Cluster 1600	32	
Servers per HMC	32	32
LPARs per HMC	128	

a. A maximum configuration of 128 servers or logical partitions is available by special order.

For performance reasons, it is recommended that the CSM master be a stand-alone machine rather than an LPAR. If the cluster is all p655s, then it is doubly recommended.

For more details on CSM, see 3.2, “Cluster System Management” on page 89. For administrative and configuration information regarding CSM, see 4.2.6, “Configuring CSM and NIM” on page 165.

2.7.5 Additional applications

The HPS may be desired for IP networking, such as the following:

- TCP and UDP applications
- Shared memory networking such as bulk transfers
- RSCT Resource Monitoring and Control
- LAPI (which has moved to RSCT from PSSP)
- MPI (Message Passing Interface)
- VSD (Virtual Shared Disks, currently using IP over HPS)
- LoadLeveler®
- GPFS
- HACMP TSM (Tivoli Storage Manager)

Information on the relationship of these applications with the HPS is documented in Chapter 3, “Application considerations” on page 73.

2.8 References

Additional useful documentation includes:

- *pSeries 690 Service Guide*, SA38-0589
- *pSeries High Performance Switch Planning, Installation and Service*, GA22- 7951
- *RSCT Group Services Programming Guide and Reference*, SA22-7888
- *IBM Reliable Scalable Cluster Technology for AIX 5L, Administration Guide*, SA22-7889
- *RSCT for AIX 5L: Technical Reference*, SA22-7890
- *RSCT for AIX 5L: Messages*, SA22-7891
- *RS/6000 and @server pSeries Adapter Placement Reference for AIX*, SA38-0538

3

Application considerations

This chapter discusses various applications and how they relate to the IBM eserver pSeries High Performance Switch (HPS).

The applications covered in this chapter are:

- Reliable Scalable Cluster Technology (RSCT)
- Cluster System Management (CSM)
- High Availability Clustering Multi-Processing (HACMP)
- IBM Virtual Shared Disks (VSD)
- IBM General Parallel File System (GPFS)
- Low-level Application Programming Interface (LAPI)
- LoadLeveler
- PE/MPI/ESSL
- Databases

3.1 RSCT Resource Monitoring and Control

RSCT (Reliable Scalable Cluster Technology) is a standard component of AIX that provides the clustering infrastructure for the IBM @server Cluster 1600 suite of products. The RSCT infrastructure contains the basic tools and architecture to build reliable and secure clustered applications.

RSCT provides the foundation for the following applications:

- IBM Virtual Shared Disk (VSD)
- IBM General Parallel File System (GPFS)
- IBM High Availability Cluster Multi-Processing (HACMP)
- IBM LoadLeveler (LL)
- IBM @server IBM @server pSeries High Performance Switch (HPS) network table functions

3.1.1 What is RSCT?

RSCT is a set of software components that work together to provide a comprehensive clustering environment for AIX and Linux. RSCT is a component of AIX 5L derived from the PSSP high-availability infrastructure.

The new IBM High Performance Computing (HPC) clusters provide a different approach from the previous version (PSSP) in the sense that there is no more control workstation (CWS) nor System Data Repository (SDR). Because of this, applications relying on the RSCT clustering infrastructure must take a different approach for storing and managing their configuration data.

RSCT is the infrastructure used by a variety of IBM products to provide clusters with improved system availability, scalability, and ease of use. RSCT is well proven to provide clustering infrastructure for CSM, Distributed Resource Managers, HACMP, GPFS, and VSD, etc.

This chapter provides an overview of the RSCT components. It describes:

- **Resource Management and Control (RMC) subsystem**

  This is the scalable, reliable backbone of RSCT. It runs on a single machine or on each node (operating system image) of a cluster and provides a common abstraction for the resources of the individual system or the cluster of nodes. You can use RMC for single system monitoring or for monitoring nodes in a cluster. RMC provides global access to subsystems and resources throughout the cluster, thus providing a single monitoring and management infrastructure for clusters.

- **RSCT core resource managers**

  A resource manager is a software layer between a resource (a hardware or software entity that provides services to some other component) and RMC. A resource manager maps programmatic abstractions in RMC into the actual calls and commands of a resource.

- **RSCT cluster security services**

  Provides the security infrastructure that enables RSCT components to authenticate the identity of other parties.

- **Topology Services subsystem**

  The Topology Services subsystem is used within an RSCT peer domain to provide other RSCT applications and subsystems with network adapter status, node connectivity information, and a reliable messaging service. The Topology Services subsystem runs as a separate daemon process on each machine (node) in the peer domain.

- **Group Services subsystem**

  The Group Services subsystem is used within an RSCT peer domain to provide other RSCT applications and subsystems with a distributed coordination and synchronization service. The Group Services subsystem runs as a separate daemon process on each machine (node) in the peer domain.

Depending on the cluster application, topology and group services have different names, although in AIX 5.2, the code used is the same:

- **topsvcs, grpsvcs** in HACMP ES clusters
- **hats, hags** in PSSP cluster
- **cthats, cthags** in peer domain clusters (RPD)

3.1.2 RSCT components

The major components of RSCT are as covered in the following sections.

Resource managers (RM)

Provides a core set of resource managers for managing base resources on single systems and across clusters. The core RSCT resource managers are:

- Audit Log resource manager

  Provides a system-wide facility for recording information about the system's operation. It is particularly useful for tracking subsystems running in the background. A command-line interface to the resource manager enables you to list and remove records from an audit log.

- Configuration resource manager

  Provides the ability to create, administer, and monitor an RSCT peer domain. This is essentially a management application implemented as a resource manager. A command-line interface to this resource manager enables you to create a new peer domain, add nodes to the domain, list nodes in the domain, and so on.

- Event Response resource manager

  Provides the ability to take actions in response to conditions occurring in the system. This is essentially a management application implemented as a resource manager. Using its command-line interface, you can define a condition to monitor. This condition is composed of an attribute to be monitored, and an expression that is evaluated periodically. You also define a response for the condition; the response is composed of zero or more actions and is run automatically when the condition occurs.

Cluster Security Services (CtSec)

CtSec is used by RSCT applications and components to perform authentication between the nodes in a cluster within both management and peer domains. Through Cluster Security Services, a cluster software component determines the identity of one of its peers, client applications, or another RSCT subcomponent. Cluster Security Services uses credential-based authentication.

Cluster Technology High Availability Topology Services (CTHATS)

CTHATS is used within an RSCT peer domain to provide other RSCT applications and subsystems with the network adapter status, node connectivity information, and reliable messaging service. CTHATS runs as a separate daemon process on each machine (node) in the peer domain.

Adapter and node connectivity information is gathered by these instances of the subsystem forming a cooperative ring called a “heartbeat” ring. In this ring, each Topology Services daemon process sends a heartbeat message to one of its neighbors and expects to receive a heartbeat from another.

In this system of heartbeat messages, each member monitors one of its neighbors. If the neighbor stops responding, the member that is monitoring it will send a message to a particular Topology Services daemon that has been designated as a Group Leader.

Cluster Technology High Availability Group Services (CTHAGS)

Cluster Technology High Availability Group Services (CTHAGS) provide other RSCT applications and subsystems with a distributed coordination and synchronization service (group membership information). The Group Services subsystem runs as a separate daemon process on each machine (node) in the peer domain.

A group is a named collection of processes. Any process may create a new group, or join an existing group, and is considered a Group Services client. Group Services guarantees that all processes in a group see the same values for the group information. Each member sees all changes to the group information simultaneously.

Figure 3-1 shows the RSCT structure and its components.

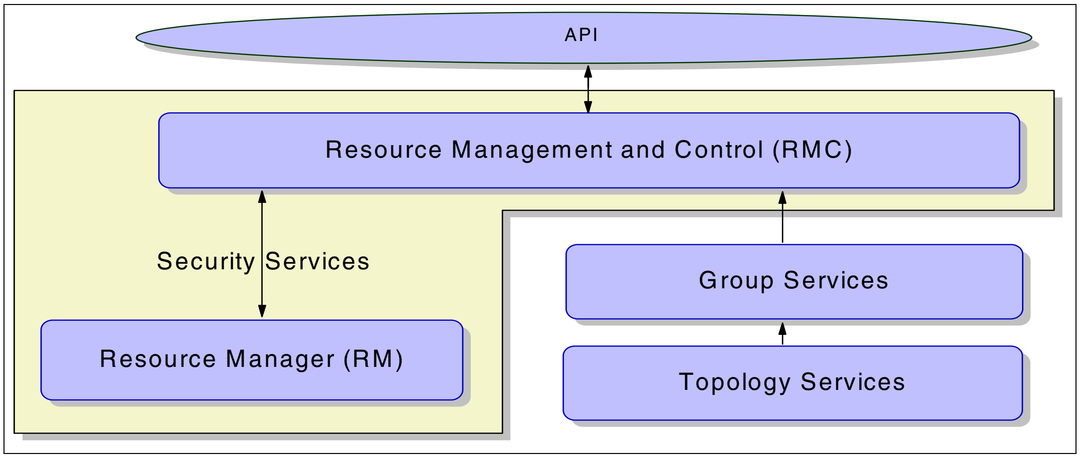


Figure 3-1 RSCT components

HAGS Global Switch Membership

Switch membership is maintained via High Availability Group Services Globalized Switch Membership (HAGSGLSM). HAGSGLSM performs the following functions:

- Registers with IBM.NetworkInterface to monitor the persistent attributes.
- Monitors the switch membership groups (cssXMemberships) for the Globally Consistent OpState.
- Updates Network Availability Matrix (NAM) information via Network Table (NTBL) ioctl() calls.
- Maintains the following CTHAGS groups:
  - HA_GS_CSS{0|1}_MEMBERSHIP_GROUP
  - HA_GS_CSSALL_MEMBERSHIP_GROUP
  - HA_GS_ML0_MEMBERSHIP_GROUP

3.1.3 Resource Management and Control (RMC) subsystem

RMC provides a single monitoring/management infrastructure for both RSCT peer domains (where the infrastructure is used by the configuration resource manager) and management domains (where the infrastructure is used by CSM). RMC can also be used on a single machine, enabling you to monitor/manage the resources of that machine.

RMC provides a generalized framework for managing resources within a single system or a cluster. Its generalized framework is used by cluster management tools to monitor, query, modify, and control cluster resources.

A resource is the fundamental concept of the RMC architecture. It is an instance of a physical or logical entity that provides services to some other component of the system. Examples of resources include lv01 on node 10, Ethernet device en0 on node 14, IP address 9.117.7.21, and so on. A set of resources that have similar characteristics (in terms of services provided, configuration parameters, and so on) is called a resource class.

Resource Monitors for IBM @server pSeries High Performance Switch

RSCT support for HPS includes *switch adapter monitoring.* The resource class IBM.NetworkInterface adds these new resources, as seen in Example 3-1.

These resources correspond to the HPS interfaces ml0 and sn#. In our test environment, we used two SNI links per node (LPAR), which gave us sn0, sn1, and ml0.

Example 3-1 New resources added for HPS monitoring

```
LPAR1 #> lsrsrc IBM.NetworkInterface
>>>>>>>>>>>>>> Omitted Lines <<<<<<<<<<<<<<<<<<<<

resource 5:
        Name            = "ml0"
        DeviceName      = ""
        IPAddress       = "10.10.10.11"
        SubnetMask      = "255.255.255.0"
        Subnet          = "10.10.10.0"
        CommGroup       = ""
        HeartbeatActive = 0
        Aliases         = {}
        NodeNameList    = {"p690_LPAR1.itso.ibm.com"}
resource 6:
        Name            = "sn1"
        DeviceName      = "sni1"
        IPAddress       = "30.30.30.11"
        SubnetMask      = "255.255.255.0"
        Subnet          = "30.30.30.0"
        CommGroup       = "CG8"
        HeartbeatActive = 1
        Aliases         = {}
        NodeNameList    = {"p690_LPAR1.itso.ibm.com"}
        DeviceSubType   = 9696
        LogicalID       = 12
        NetworkID       = 1
resource 7:
        Name            = "sn0"
        DeviceName      = "sni0"
        IPAddress       = "20.20.20.11"
        SubnetMask      = "255.255.255.0"
        Subnet          = "20.20.20.0"
        CommGroup       = "CG3"
        HeartbeatActive = 1
        Aliases         = {}
        NodeNameList    = {"p690_LPAR1.itso.ibm.com"}
        DeviceSubType   = 9696
        LogicalID       = 7
        NetworkID       = 1
```

Notice that three new attributes have been added for these resources:

- LogicalID
- NetworkID
- DeviceSubType

These resources can be monitored by configuring the proper conditions. To configure a resource monitor for the HPS, proceed as follows:

1. Create the condition

   To create the condition you can use the command-line interface or the graphical interface provided by WebSM.

   In our example, we use the graphical interface provided by WebSM.

   a. Select Monitoring

      To create the condition, open the WebSM and double-click the **Monitoring** icon as shown in Figure 3-2.

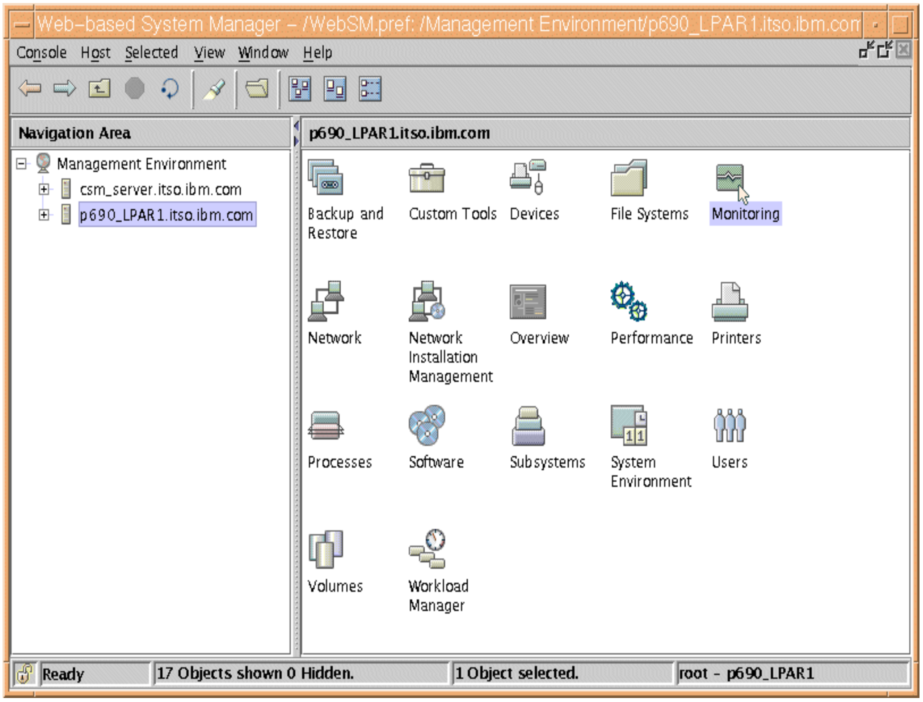


Figure 3-2 Web-Based System Manager - Monitoring

   b. Select Conditions

      Next, double-click the **Conditions** icon, as shown in Figure 3-3 on page 79.

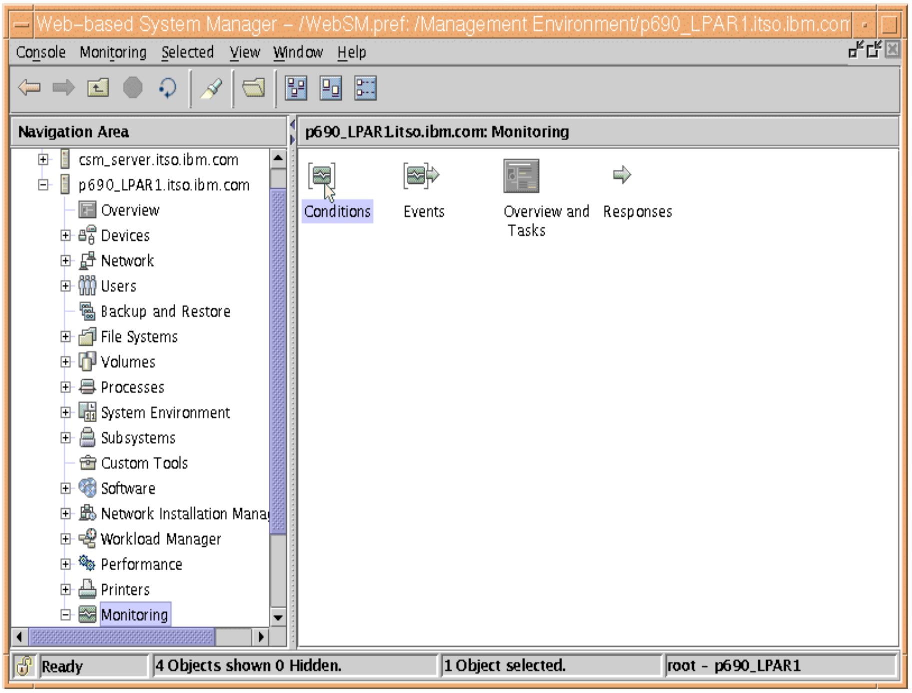
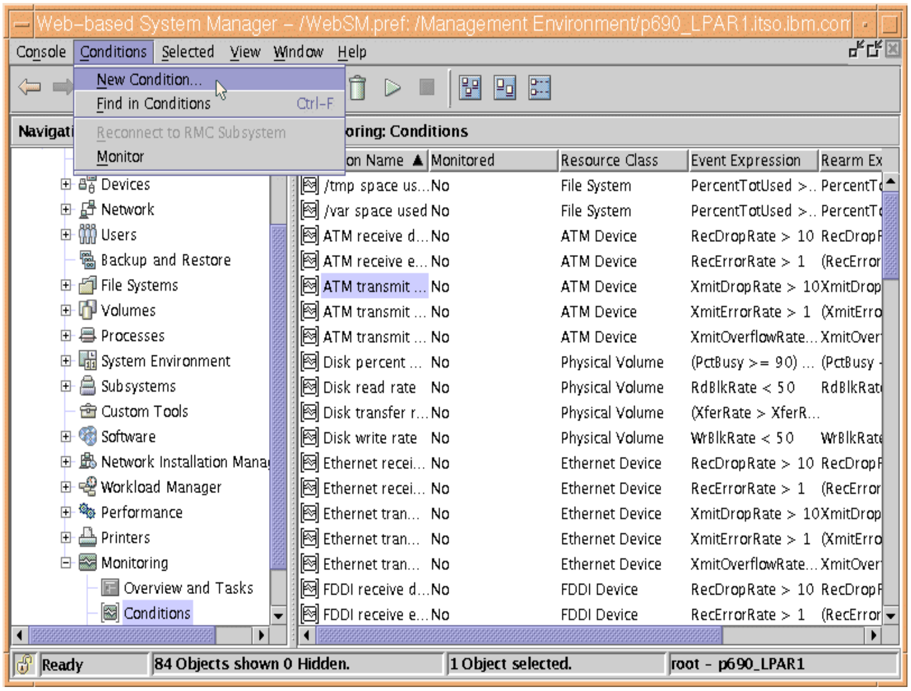


Figure 3-3 Web-Based System Manager - Conditions

c. Add Condition

You should now see a list of conditions. You have to add a condition for the HPS by selecting **Conditions -> New Condition** as seen in Figure 3-4.



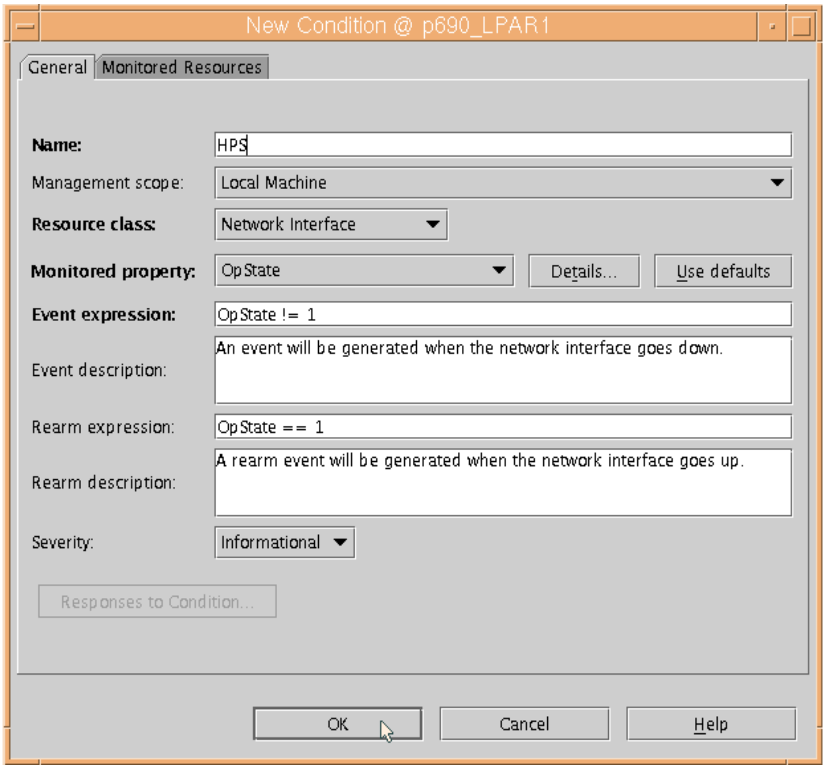
on Name ▲	Monitored	Resource Class	Event Expression	Rearm Ex
/tmp space us...	No	File System	PercentTotUsed >..	PercentT
/var space used	No	File System	PercentTotUsed >..	PercentT
ATM receive d...	No	ATM Device	RecDropRate > 10	RecDropR
ATM receive e...	No	ATM Device	RecErrorRate > 1	(RecError
ATM transmit ...	No	ATM Device	XmitDropRate > 10	XmitDrop
ATM transmit ...	No	ATM Device	XmitErrorRate > 1	(XmitErro
ATM transmit ...	No	ATM Device	XmitOverflowRate...	XmitOver
Disk percent ...	No	Physical Volume	(PctBusy >= 90) ...	(PctBusy
Disk read rate	No	Physical Volume	RdBlkRate < 50	RdBlkRate
Disk transfer r...	No	Physical Volume	(XferRate > XferR...	
Disk write rate	No	Physical Volume	WrBlkRate < 50	WrBlkRate
Ethernet recei...	No	Ethernet Device	RecDropRate > 10	RecDropR
Ethernet recei...	No	Ethernet Device	RecErrorRate > 1	(RecError
Ethernet tran...	No	Ethernet Device	XmitDropRate > 10	XmitDrop
Ethernet tran...	No	Ethernet Device	XmitErrorRate > 1	(XmitErro
Ethernet tran...	No	Ethernet Device	XmitOverflowRate...	XmitOver
FDDI receive d...	No	FDDI Device	RecDropRate > 10	RecDropR
FDDI receive e...	No	FDDI Device	RecErrorRate > 1	(RecError

Figure 3-4 Web-Based System Manager - New Conditions

Note: In this example we had to create the condition. In future versions of RSCT, all conditions for HPS will most likely be built in.

d. Enter values

In the window shown in Figure 3-5, enter the proper values. In this example we entered `HPS` for the condition name and selected **Op State** for Monitored property.



Name: HPS

Management scope: Local Machine

Resource class: Network Interface

Monitored property: OpState

Event expression: OpState != 1

Event description: An event will be generated when the network interface goes down.

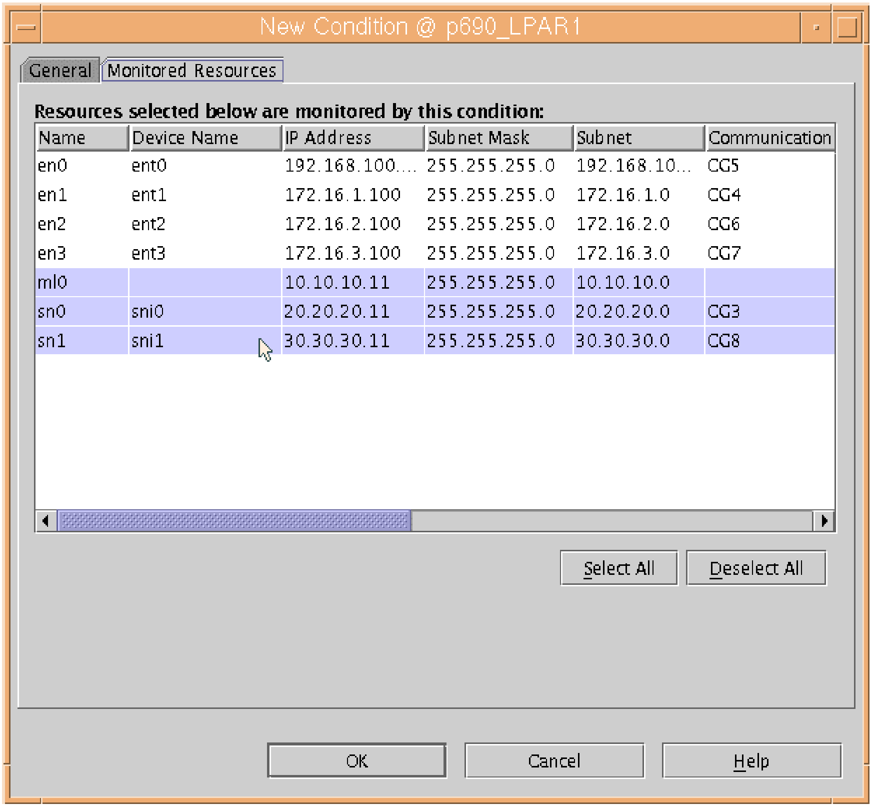
Rearm expression: OpState == 1

Rearm description: A rearm event will be generated when the network interface goes up.

Severity: Informational

Figure 3-5 New Condition - General

On the Monitored Resources tab, you are able to select the interfaces you are going to monitor, as seen on Figure 3-6 on page 81.



Resources selected below are monitored by this condition:

Name	Device Name	IP Address	Subnet Mask	Subnet	Communication
en0	ent0	192.168.100....	255.255.255.0	192.168.10...	CG5
en1	ent1	172.16.1.100	255.255.255.0	172.16.1.0	CG4
en2	ent2	172.16.2.100	255.255.255.0	172.16.2.0	CG6
en3	ent3	172.16.3.100	255.255.255.0	172.16.3.0	CG7
ml0		10.10.10.11	255.255.255.0	10.10.10.0	
sn0	sni0	20.20.20.11	255.255.255.0	20.20.20.0	CG3
sn1	sni1	30.30.30.11	255.255.255.0	30.30.30.0	CG8

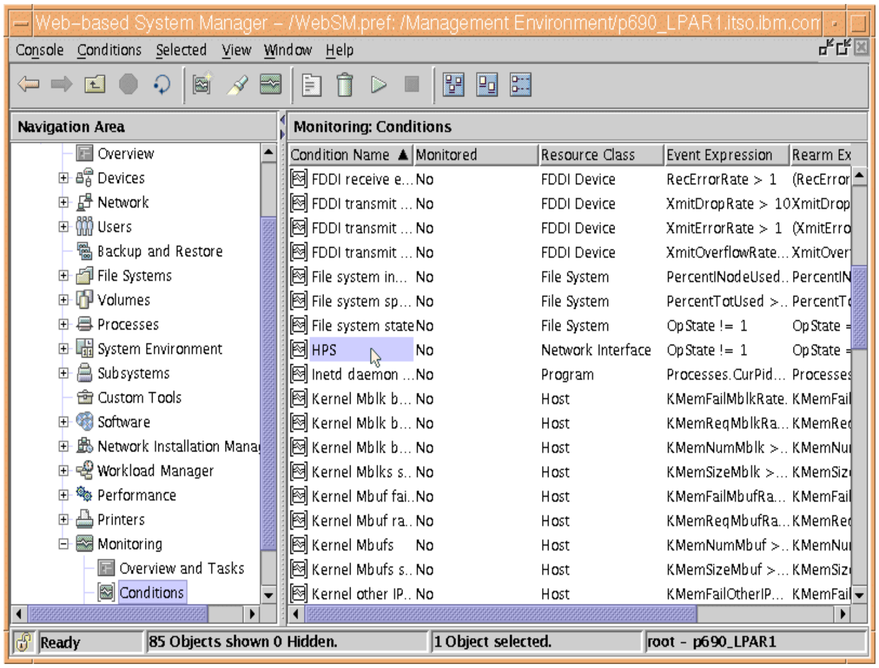
Figure 3-6 New Condition - Monitored Resources

2. Associate a response to the condition.

   Now that you have the condition defined, you have to associate the proper response to that condition. In this example, we are going to configure this monitor to send mail to root every time the condition occurs.

   a. Check the created monitored conditions.

      In Figure 3-7 on page 82, we can see that the condition HPS is created. The value in the second column shows that this condition is not being monitored.



Monitoring: Conditions

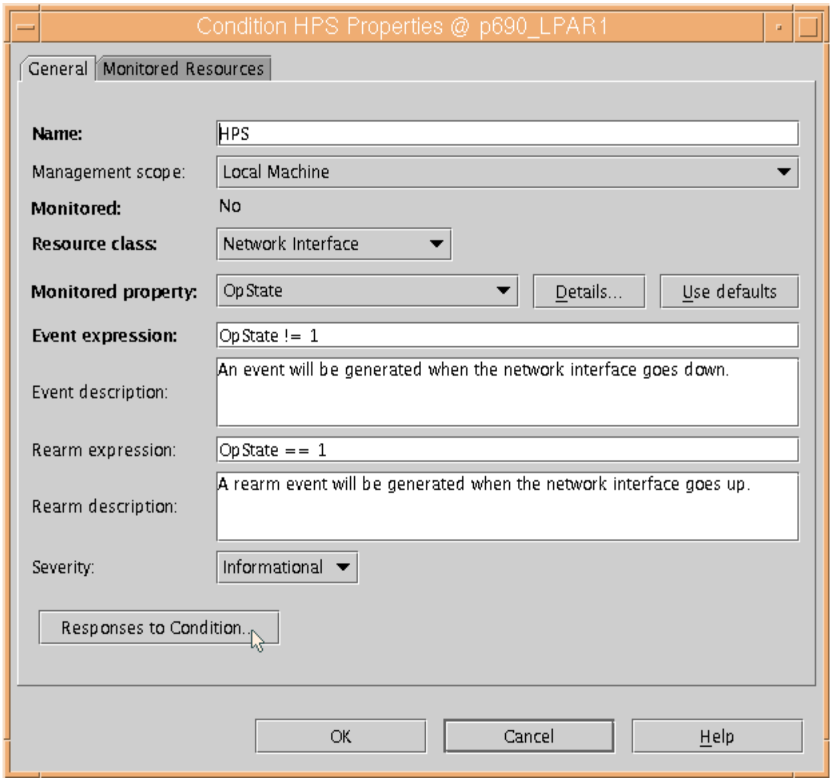
Condition Name ▲	Monitored	Resource Class	Event Expression	Rearm Ex
FDDI receive e...	No	FDDI Device	RecErrorRate > 1	(RecError
FDDI transmit ...	No	FDDI Device	XmitDropRate > 10	XmitDrop
FDDI transmit ...	No	FDDI Device	XmitErrorRate > 1	(XmitErro
FDDI transmit ...	No	FDDI Device	XmitOverflowRate...	XmitOver
File system in...	No	File System	PercentINodeUsed..	PercentIN
File system sp...	No	File System	PercentTotUsed >..	PercentT
File system state	No	File System	OpState != 1	OpState =
HPS	No	Network Interface	OpState != 1	OpState =
Inetd daemon ...	No	Program	Processes.CurPid...	Processes
Kernel Mblk b...	No	Host	KMemFailMblkRate.	KMemFail
Kernel Mblk b...	No	Host	KMemReqMblkRa...	KMemRe
Kernel Mblk b...	No	Host	KMemNumMblk >..	KMemNu
Kernel Mblks s..	No	Host	KMemSizeMblk >...	KMemSiz
Kernel Mbuf fai..	No	Host	KMemFailMbufRa...	KMemFail
Kernel Mbuf ra..	No	Host	KMemReqMbufRa...	KMemRe
Kernel Mbufs	No	Host	KMemNumMbuf >..	KMemNu
Kernel Mbufs s..	No	Host	KMemSizeMbuf >...	KMemSiz
Kernel other IP..	No	Host	KMemFailOtherIP...	KMemFail

Ready | 85 Objects shown 0 Hidden. | 1 Object selected. | root - p690_LPAR1

Figure 3-7 Web-based System Manager - Conditions

b. Assign a response to the condition

To assign a response to the HPS condition, double-click it. Figure 3-8 shows us the Condition Properties window, where we select the **Responses to Condition** button.


Condition HPS Properties @ p690_LPAR1
General
Monitored Resources
Details...
Use defaults
Responses to Condition...
OK
Cancel
Help

Name: HPS

Management scope: Local Machine

Monitored: No

Resource class: Network Interface

Monitored property: OpState

Event expression: OpState != 1

Event description: An event will be generated when the network interface goes down.

Rearm expression: OpState == 1

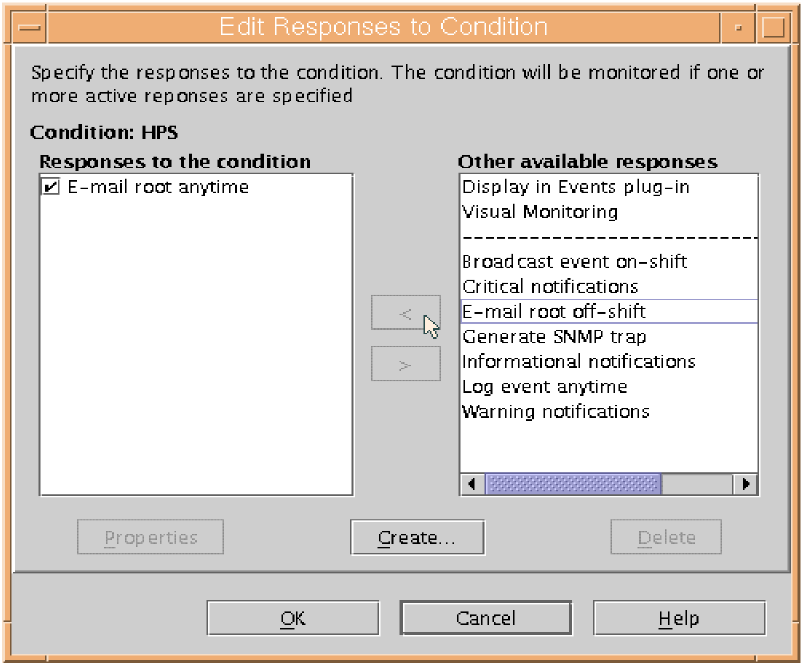
Rearm description: A rearm event will be generated when the network interface goes up.

Severity: Informational

Figure 3-8 Condition Properties - Responses to Condition

The Edit Responses to Condition window is displayed. Select **E-mail root anytime**, and pass it to the Responses to the condition space, as seen in Figure 3-9.

Finally, click the **OK** button. The new HPS Resource monitor is now active.



Edit Responses to Condition
Specify the responses to the condition. The condition will be monitored if one or more active reponses are specified
Condition: HPS
Responses to the condition
☑ E-mail root anytime
Other available responses
Display in Events plug-in
Visual Monitoring
Broadcast event on-shift
Critical notifications
E-mail root off-shift
Generate SNMP trap
Informational notifications
Log event anytime
Warning notifications
<
>
Properties
Create...
Delete
OK
Cancel
Help

Figure 3-9 Edit Responses to Condition

To check that the condition and the associated response have been created, issue the **lscondresp** command, as shown in Example 3-2.

The **lscondresp** command lists information about a condition and its linked responses, if any.

Example 3-2 Condition-Response list

```
LPAR1 #> lscondresp
Displaying condition with response information:
Condition Response              Node                      State
"HPS"     "E-mail root anytime" "p690_LPAR1.itso.ibm.com" "Active"
```

3. Checking occurrences of monitored conditions

   In this example we configured an e-mail response for the monitored condition. Every time the monitored condition occurs (the network interface goes down or goes up), root receives e-mail notification of the occurrence.

   Example 3-3 shows the received e-mail.

Example 3-3 Mail from resource monitor

```
LPAR1 #> mail
Mail [5.2 UCB] [AIX 5.X]  Type ? for help.
"/var/spool/mail/root": 1 messages 1 new
>N  1 root              Mon Nov  3 19:17  26/689  "HPS"
? 1
Message  1:
From root Mon Nov  3 19:17:46 2003
Date: Mon, 3 Nov 2003 19:17:45 -0500
From: root
```

```
To: root
Subject: HPS

=====================================

Monday 11/03/03 19:17:44

Condition Name: HPS
Severity: Informational
Event Type: Event
Expression: OpState != 1

Resource Name: ml0
Resource Class: IBM.NetworkInterface
Data Type: CT_UINT32
Data Value: 2
Node Name: p690_LPAR1.itso.ibm.com
Node NameList: {p690_LPAR1.itso.ibm.com}
Resource Type: 0
=====================================
```

You can configure other responses for this condition or other conditions. For further information about resource monitoring, refer to *IBM Reliable Scalable Cluster Technology for AIX 5L, Administration Guide*, SA22-7889-02.

3.1.4 Security

Security in cluster environments is important to prevent unauthorized access to resources within the cluster.

In addition to the cluster security environment that secures the cluster communication and node membership, you still need to consider the operating system and application security issues.

Some general security requirements in cluster environments are as follows:

- Authentication

  Authentication is the process used to ensure that a client or server is the one identity it claims to be. Cluster applications, such as CSM or resource managers, need to know whether the client is really part of the cluster or is simply attempting to obtain unauthenticated access to resources.

- Authorization

  Authorization is the process used to enable or deny user and node access to resources within the cluster. After the participating nodes have been authenticated successfully, both must ensure the access control to their resources. Basically, this process does a lookup in an access control list (ACL) to retrieve the access based on given criteria.

- Data privacy

  When transmitting data between nodes, the data should be encrypted in a way that only the acceptor can decrypt it. This is called data privacy. It is a many-to-one (n:1) relationship, where every node can send data to another node, but only the acceptor node can decrypt and use that data. If the data has been intercepted by another node, that node will not be able to decrypt it.

- Data integrity

  To ensure that received data is from the expected sender node, data must be encrypted in such a way that the acceptor can be sure that the received data came from the expected node. This is called data integrity. It is a one-to-many (1:n) relationship, where the sender encrypts the data and everybody receiving this data can decrypt it. Because any node that has intercepted the data can decrypt it, this data should not contain sensitive information.

RSCT's cluster security services (CtSec) provides the security infrastructure that enables RSCT components to authenticate and authorize the identity of other parties requiring access to cluster managed resources.

Cluster Security Services (CtSec) uses *credential-based authentication*. This type of authentication is used in client/server relationships and enables:

- A client process to present information that identifies the process in a manner that cannot be imitated to the server.
- The server process to correctly determine the authenticity of the information from the client.

Credential-based authentication uses a third party that both the client and the server trust. In the case of UNIX host-based authentication, the trusted third party is the UNIX operating system. This method of authentication is used between RSCT and its client applications (such as CSM).

The CtSec library consists of several components that are required to provide the current and future security functions.

These are the main features of CtSec:

- Provides common, generic cluster security services
- Requires minimal configuration and management
- Is common to AIX and Linux
- Is installed as part of base AIX via the RSCT rsct.core.sec fileset
- Provides out-of-box cluster security
- Minimizes dependencies on other services

CtSec provides security services to other applications:

- Cluster types:
  - CSM cluster - CSM Management Domains
  - RSCT clusters - RSCT Peer Domains
  - Other clusters - Independent workstations, LoadLeveler, etc.
- Resource managers in:
  - CSM clusters - client/server, Message Security Services (MSS)
  - Peer Domains - peer-to-peer (P2P), MSS
  - HMC-to-LPAR - client/server, MSS
- Other client/server applications (CtSec is optional in this group):
  - LoadLeveler
  - Parallel Operating Environment (POE)
  - Dynamic Probe Class Library (DPCL)

CtSec is implemented in the rsct.core.sec fileset and is installed with base AIX via RSCT. The version for this fileset is 2.3.1.

UNIX host-based authentication (HBA)

The default authentication mechanism provided by cluster security services and utilized by RSCT is UNIX host-based authentication. HBA is comprised of two key components:

- Mechanism Pluggable Module (MPM)

  Each MPM converts the general tasks, received from the Mechanism Abstract Layer (MAL), into necessary tasks the security mechanism uses to satisfy the MAL request. The current version of RSCT provides only one MPM - the unix.mpm. A Kerberos MPM is under development.

- Cluster authentication services daemon - ctcasd
  - Provides and authenticates UNIX identity-based credentials for cluster security services.
  - It is started by the RMC Service whenever the RMC service starts.
  - The first time the ctcasd daemon starts on a particular node, it will create the private key and the public key for that node (Private and Public Key pair - PPK).

HBA is defined in the CtSec configuration file, /usr/sbin/rsct/cfg/ctsec.cf. This file lists the path name of the available MPM.

> **Note:** You should *not* modify the /usr/sbin/rsct/cfg/ctsec.cf file. However, you should be aware that the file exists and is used by cluster security services to locate the MPM.

HBA is based on public key cryptography (asymmetric cryptography). The node uses its private key to encrypt data. The node's public key is provided to other nodes and will be used by them to decipher the data. Similarly, the public key is used by the other nodes to encrypt data that is then deciphered using the node's associated private key. A node's public key is intended to become public knowledge, while the private key remains secret, known only to the node's *root* user.

Figure 3-10 shows the object and file structure for RSCT.

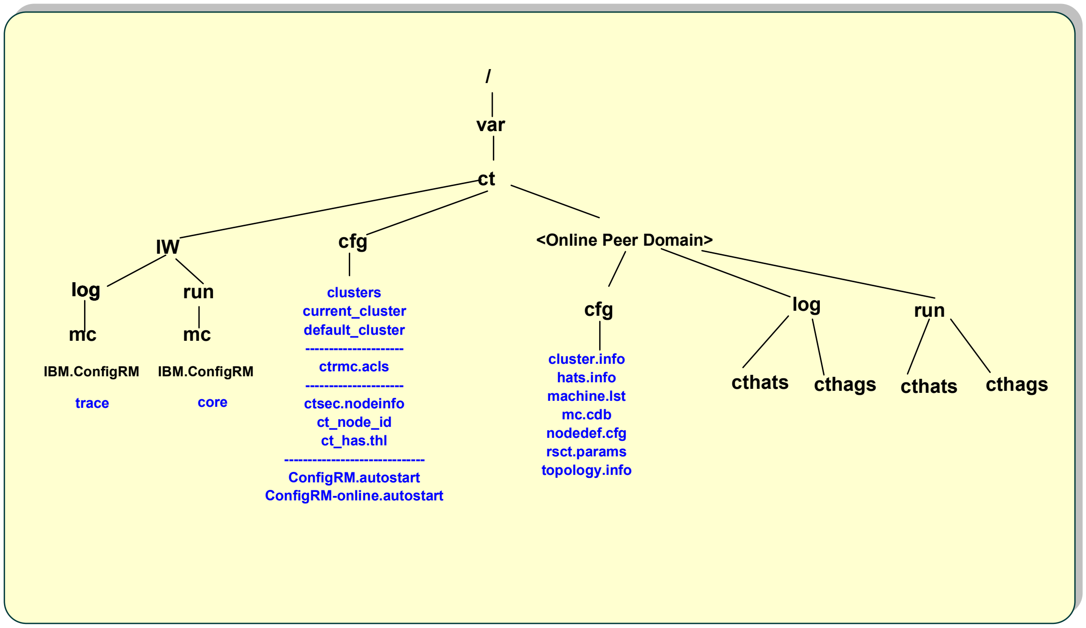


Figure 3-10 RSCT objects and file structure

Important: Host name resolution is extremely important for RSCT. Always be sure to verify that forward and reverse resolution matches in case, spelling, and fullness and it does not contain multiple entries with the same IP address or host name.

In our environment, we configured long host names instead of short ones, because of the problems we encountered using the short ones.

3.1.5 Cluster environment

In AIX 5L, nodes in a cluster can be configured for any of the following cluster domains:

- Stand-alone RMC
- Peer domain RMC
- Management domain RMC

Stand-alone RMC

A stand-alone RMC consists of a single node configuration. The RMC subsystem and core resource managers are used to manage node resources.

Figure 3-11 shows the structure of a single node, stand-alone RMC.

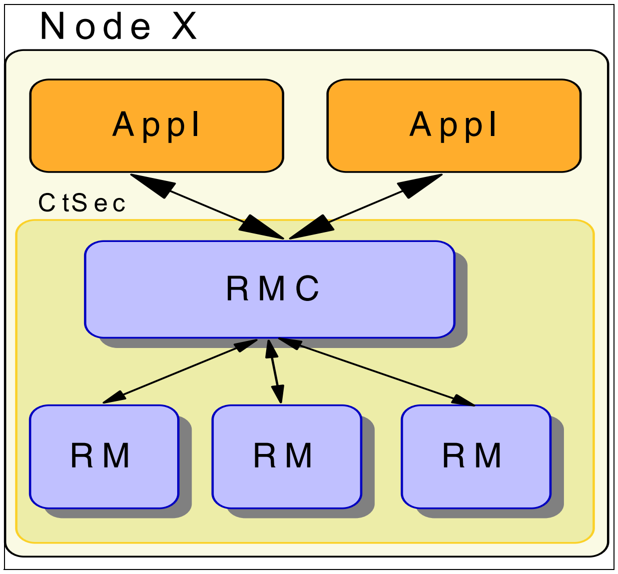


Figure 3-11 Stand-alone RMC

Peer domain RMC

A peer domain RMC consists of a number of nodes with no distinguished or master node. All nodes are aware of all other nodes. Administration commands can be issued from any node in the domain. All nodes have a consistent view of the domain membership.

- The processor architecture and operating system in peer domains are *homogeneous*. They can also be configured to work between LPARs on an IBM @server pSeries 690.
- The RMC subsystem and core resource managers manage cluster resources.
- RSCT cluster security services are used to authenticate other parties.
- The Topology Services subsystem provides node/network failure detection.
- The Group Services subsystem provides cross node/process coordination.

Figure 3-12 shows the structure of a peer domain RMC.

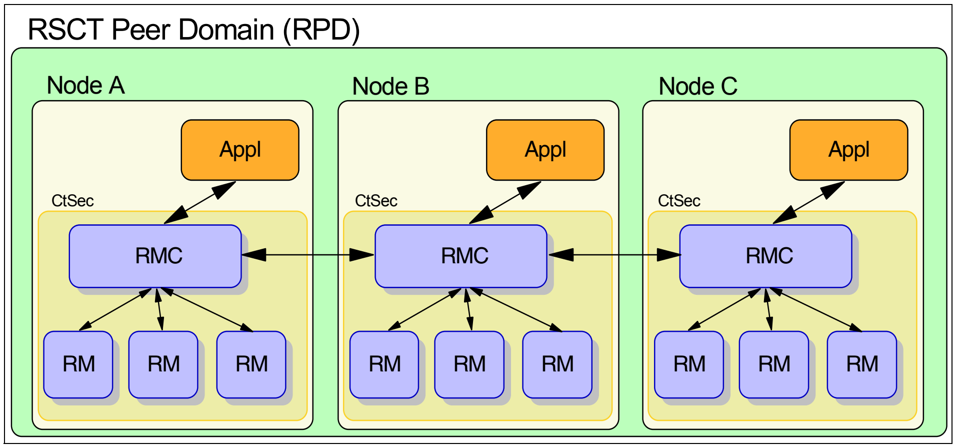


Figure 3-12 Peer domain RMC

For a practical example of an RSCT peer domain configuration, refer to 5.2.4, "Creating the RSCT peer domain" on page 196.

Management domain RMC

A management domain RMSC has a management server that is used to administer a number of managed nodes. Only management servers have knowledge of the whole domain. Managed nodes only know about the server managing them. Managed nodes know nothing of each other.

- Processor architecture and operating system are *heterogeneous*.
- The RMC subsystem and core resource managers are used by CSM to manage cluster resources. CSM also provides an additional resource manager, the domain resource manager.
- RSCT cluster security services authenticate other parties.
- The Topology Services subsystem is *not* needed.
- The Group Services subsystem is *not* needed.

Figure 3-13 on page 89 shows the structure of a management domain RMC.

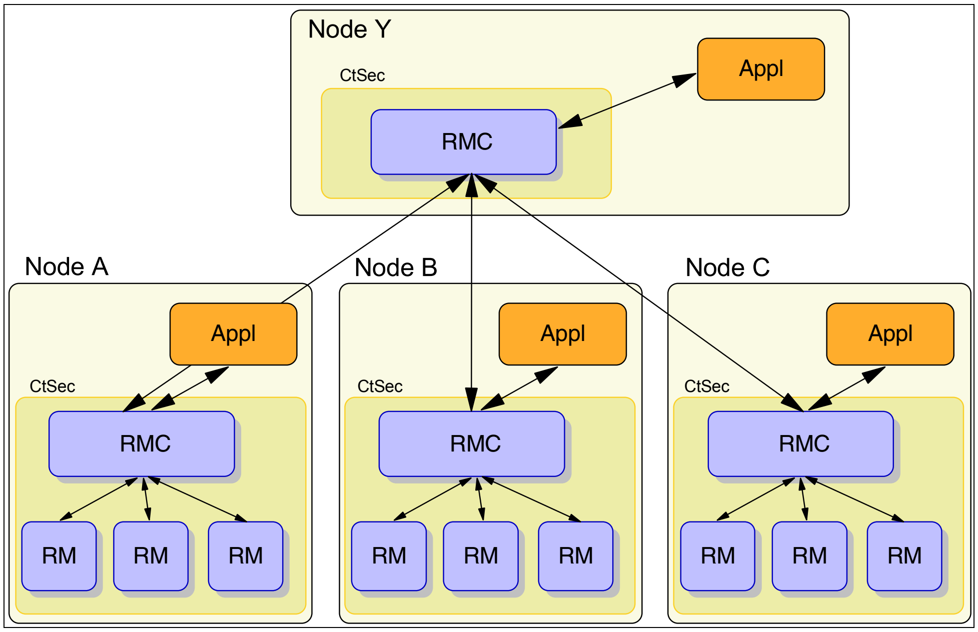


Figure 3-13 Management domain RMC

3.2 Cluster System Management

Cluster System Management (CSM) is an IBM software package that provides management functions for multiple systems. The CSM managed systems can be pSeries running AIX or Linux and xSeries running Linux. CSM provides cluster management tools to install, update, and configure managed nodes.

CSM relies on standard AIX components such as Reliable Scalable Cluster Technology (RSCT), Resource Monitoring and Control (RMC), and Network Installation Manager (NIM). CSM Version 1.3.2 adds support for the High Performance Switch (HPS) as well as other enhancements, described in the following sections.

3.2.1 Supported platforms

Several node types are supported with CSM both with and without a pSeries HPS. Nodes in a Parallel System Support Program (PSSP) managed cluster are not supported with the HPS. PSSP filesets conflict with HPS filesets, specifically the LAPI and CSS filesets.

As such, PSSP nodes cannot use an HPS. Since PSSP requires all or none of the LPARs within a CEC, there can be no mixing of PSSP and HPS LPARs within the same CEC. It is acceptable to have PSSP and HPS CECs on the same HMC.

Note: The following information applies to Cluster System Management (CSM) only.

The following hardware for the cluster nodes is supported in a CSM environment.

Supported pSeries servers for CSM hardware control

The following hardware is supported for the AIX nodes in a CSM cluster:

- IBM Cluster 1600 pSeries 615 (p615)
- IBM pSeries 630 (p630)

- IBM pSeries 650 (p650)
- IBM pSeries 655 (p655)
- IBM pSeries 670 (p670)
- IBM pSeries 690 (p690)
- IBM 9076 SP Nodes (feature numbers 2050, 2051, 2052, 2053, 2056, and 2057)
- IBM 7026 servers (models H80, 6H0/6H1, 6M1, and pSeries 660 or p660)

Supported xSeries server for CSM hardware control

The following hardware is supported for the Linux nodes in a CSM cluster:

- IBM xSeries 330 (x330)
- IBM xSeries 335 (x335)
- IBM xSeries 342 (x342)
- IBM xSeries 345 (x345)
- IBM xSeries 360 (x360)
- IBM xSeries 440 (x440)
- IBM xSeries 445 (x445)
- IntelliStation® 6221
- BladeCenter™: 8677 BladeCenter Management Module, HS20 (8678) Blade Server

Software for AIX nodes

The following software is supported for the AIX nodes in a CSM cluster:

- CSM requires AIX 5.2 with APARs IY34493 and IY34724 installed on the machine configured as the CSM Management Server.
- AIX 5.2 with APARs IY34493 and IY34724 is supported for the nodes in an AIX cluster.
- AIX 5.1 with the 5100-03 Recommended Maintenance package and APARs IY34707 and IY34725 is supported for the nodes in the cluster.

Software for Linux nodes

Table 3.1 shows the current Linux distributions and supported hardware for Linux nodes in an AIX cluster.

Table 3-1 Supported Linux distributions and hardware

Linux Distribution	x330	x335	x342	x345	x360	x440	x445	BladeCenter	IntelliStation Model 6221
Red Hat 7.2	X		X	X					
Red Hat 7.3	X	X	X	X	X			X	X
Red Hat AS 2.1	X	X	X	X	X	X	X	X	
Red Hat 8.0		X		X	X			X	X
SuSE 8.0	X	X	X	X		X			
SuSE 8.1		X		X	X	X		X	
SuSE SLES 7 (7.2)	X	X	X	X		X			
SuSE SLES 8 (8.1)		X		X	X	X	X	X	

For further documentation about CSM and its requirements, refer to:

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

3.2.2 Client install mechanisms

Different install mechanisms are available depending on the kind of node you are installing. For Linux nodes, you can use KickStart if you are using Red Hat or AutoYaST if you are using SuSE distributions.

Network Installation Manager (NIM) for AIX

NIM can be used to install AIX operating software onto the CSM cluster nodes. Through NIM you can install a group of nodes with a common configuration or customize an installation for the specific needs of a given node.

Support for additional adapter configuration has been added.

NIM and CSM have added support for additional Ethernet and HPS adapters. Adapters can be configured during the initial install or after the install.

NIM support for additional adapter configuration

NIM supports additional adapter configuration during **bos_inst** and **cust** operations. To support these operations, NIM uses an adapter stanza file similar to a node definition stanza file. New resources have been added to support additional adapter configuration:

- The adapter_def resource defines the directory where adapter configuration information will reside.
- The **nimadapters** command uses a stanza file to create the configuration files for each node in the adapter_def directory.

CSM support for additional adapter configuration

Some enhancements have been added to the **getadapters** command to support additional adapter configuration:

- Collects additional adapter information from the nodes
- Uses **dsh** on running systems and open firmware interfaces when a node is down
- Can collect install and additional adapter information with one call
- Can be used to create an initial version of the adapter stanza file
- Stanza file can be edited and passed back to **getadapters** and it can also be passed to **nimadapters**

This is a brief description of the procedure to configure additional adapters on nodes:

1. Gather adapter information and create stanza file:

   ```
   getadapters -z mystanzafile -n clstrn01
   ```

   Edit the stanza file to add the required information.

2. Create the NIM adapter_def resource:

   ```
   nim -o define -t adapter_def -a server=master -a \ location=/export/nim/adapter_defs
   my_adapt_res
   ```

3. Create adapter configuration files for each node:

   ```
   nimadapters -d -f mystanzafile my_adapt_res
   ```

4. Configure the adapters (use NIM **bos_inst** or **cust** operation):

   ```
   nim -o cust -a adapter_def=my_adapt_res clstrn01
   ```

KickStart for Red Hat

KickStart enables the automation of Red Hat Linux installation and several customization tasks, such as:

- Language selection
- Network configuration and distribution source selection
- Keyboard selection
- Boot loader installation (for example lilo)
- Disk partition and filesystem creation
- Mouse selection
- X Windows system server configuration
- Time zone selection
- Selection of an (initial) root password
- Which packages to install

KickStart allows the user to script the regular installation procedure by placing all the information that is needed to install the machine into a configuration file. It is also possible to specify a list of shell commands to execute after the installation. Through this mechanism, it is possible to install extra software automatically. KickStart is *not* supported with CSM for AIX. For further information about possible configurations, refer to the following redbooks:

- *Linux HPC Cluster Installation,* SG24-6041
- *An Introduction to CSM 1.3 for AIX 5L,* SG24-6859

AutoYaST for SuSE

In CSM 1.3.2 AutoYaST replaces SIS for SuSE and SLES install on i386 and ppc64 automatic installations. AutoYaST can be used to automatically install one or more SuSE Linux systems. The process for an automatic installation is similar to the KickStart process. The following commands are used to run an automatic installation for SuSE and SLES:

```
installms, systemid, csmconfig -L, definenode, csmsetupyast, installnode
installMethod=autoyast
```

csmsetupyast copies the SuSE or SLES CDs to /csminstall/Linux/SuSE/8.1/arch or /csminstall/Linux/SLES/8.1/arch directory, respectively.

> **Note:** SLES 8.1 and UnitedLinux 1.0 have conflicting directories, so **csmsetupyast** creates symbolic links.

A full install with AutoYaST is not supported on SuSE 8.0 or SLES 7. SuSE 8.1 and SLES 8 are supported on xSeries. SLES 8 is supported on pSeries.

The yastcfg.SLES8.1-arch.xml configuration file is a default configuration file, not a template. The user can provide any valid AutoYaST configuration file. The configuration file provided by the user is modified by the **csmsetupyast** command as necessary. The command **csmsetupyast** creates a customized, node-specific configuration file (/csminstall/csm/1.3.2/autoyast.SLES8.1/<node_ipaddr>-autoyast.xml).

The Perl XML editor (select **SetupBootUtils -> createAutoyastFiles**) is used to edit and parse yastcfg.*.xml files. This tool:

- Adds missing sections and subsections: scripts, users, networking, init, report, general, partitioning, bootloader, mode, clock, language
- Adds root and admin user IDs if they don't already exist
- Adds /dev/sda or /dev/hda (depending on InstallDisk or InstallDiskType attribute) if one doesn't exist
- Modifies options so that the install is completely unattended
- Adds CSM's chrootscript and postscript

It does not modify RPM package selections.

AutoYaST configuration scripts are separate files from the configuration file. The configuration script /opt/csm/install/autoyast.chrootscript.tmpl.arch runs after installation and before reboot. This configuration script runs in a chroot environment. The real installed machine is mounted under /mnt. The configuration script /opt/csm/install/autoyast.postscript.tmpl.arch runs after installation and after reboot. The command **csmsetupyast** customizes the script templates and inserts them into the configuration file.

> **Note:** These files can be modified by the user, but we recommend that you add your own scripts.

This is a brief description of the SuSE xSeries install process using AutoYaST:

1. User runs **installnode**.
2. **rpower** boots node, which broadcasts the MAC address.
3. **dhcpd** on CSM Management Server invokes **pxelinux**, which reads /tftboot/pxelinux.cfg/<NODE_IP_IN_HEX>.
4. **pxelinux** sends the AutoYaST kernel and ramdisk to the node via TFTP.
5. AutoYaST installs the node.
6. AutoYaST **chrootscript** runs (performs the configuration, runs **csmprereboot**, and sets the boot order to the hard disk).
7. **csmprereboot** adds **csmfirstboot** to /etc/inittab.
8. The node reboots to the hard disk.
9. AutoYaST postscripts run (which does more node configuration).
10. **csmfirstboot** runs (runs **makenode**, which installs CSM and sets the management server).
11. The node is installed.

The following files can be checked to watch remote console or logs during the install and to debug problems found during installation:

- /var/log/csm/csmsetupyast.log
- /var/log/csm/installnode.log
- /var/log/csm/install.log (on the nodes)

It is possible to migrate from SIS to AutoYaST. The SIS configuration can be migrated to the AutoYaST format. To migrate the format from SIS to AutoYaST, the **csmsis2yast** command must be run. The **csmsis2yast** command merges a SIS disk partition table and a SIS package list into an AutoYaST configuration file.

AutoYaST can also be used to automatically install SuSE Linux on pSeries nodes. pSeries Linux supports SLES 8. The following attributes are required for **getadapters** and a full install:

- InstallAdapterDuplex
- InstallAdapterSpeed
- InstallAdapterType

For values not set, **getadapters** provides defaults. The MAC address is collected by **getadapters** calling **hmc_nodecond**, which communicates with the open firmware on the node.

This is a brief description of the SuSE pSeries install process using AutoYaST:

1. User runs **installnode**.

2. **installnode** adds an ARP entry on the CSM Management Server for each node to allow TFTP and NFS access.
3. **hmc_nodecond** reboots the node, which communicates with the open firmware to let the node boot from the specific network adapter.
4. The node's firmware loads the kernel from the network via TFTP.
5. AutoYaST installs the node.
6. AutoYaST postcripts runs (performs configuration, runs **csmprereboot**, runs **setbootdisk**).
7. **csmprereboot** adds **csmfirstboot** to /etc/inittab.
8. **setbootdisk** sets the corresponding SCSI hard disk to be the first bootable device in the open firmware.
9. The node does not reboot to the hard disk.
10. **csmfirstboot** runs (runs **makenode**, which installs CSM and sets the management server).
11. The node is installed.
12. NodeFullInstallComplete/removeArpEntries condition/response removes ARP entries from the CSM Management Server.

Note: We recommend choosing the latest patch kernel to install PPC nodes (the kernel version should be 2.24.19-228 or above to solve installation issues). To get the latest kernel version, go to `http://support.suse.de/psdb/`.

The following files can be checked to watch remote console or logs during a Linux install on pSeries nodes and to debug problems found during installation:

- /var/log/csm/csmsetupyast.log
- /var/log/csm/installnode.log
- /var/log/csm/installnode.<node name>.log (hmc_nodecond info during installnode)
- /var/log/csm/getadapters/getadapters.<node name>.log
- /var/log/csm/install.log (on the node)

3.2.3 CSM 1.3.2.1 November 2003

New supported platforms

CSM 1.3.2.1 adds support for:

- IBM @server xSeries 325 (e325) Model 8835

  The operating system supported on e325 servers is SLES 8 Linux.

- IBM @server xSeries BladeServer Model 8832

  The operating system supported on BladeServer is SuSE 8.1 Linux.

CSM 1.3.2.1 has been tested and approved to support up to 512 IBM @server xSeries nodes and up to 128 IBM @server pSeries nodes.

Important: Support for up to 1024 IBM @server xSeries and 1024 IBM @server pSeries is planned.

CSM 1.3.2.1 also adds support for HPS. New functions have been added to the **getadapters** command to support the new switch. The **getadapters** command reports HPS adapters. The adapter type is *sni*.

CSM backup and restore scripts

Scripts for saving and restoring critical Management Server data to and from a specified directory have been added. This scripts are:

- **csmbackup** - Copies and stores vital CSM data to a specified directory
- **csmrestore** - Restores CSM data previously backed up by the **csmbackup** command

The data saved by **csmbackup** is as follows:

- Resources (all, unless individual ones specified by -r):

  Sensor, EventResponse, Condition, Association, ManagedNode, NodeGroup, DmsCtrl
- All Event Response Resource Monitor (ERRM) scripts
- All install configuration templates (KickStart and AutoYaST)
- User-installed customization scripts
- Hardware control password files in /etc/opt/csm/system_config
- Distributed Command Execution Module (DCEM) customization and user scripts
- Listing of cfmroot files
- Listing of OS distribution (distro) directories
- All user-specified files listed in -f file

The files created by **csmbackup** are stored by default in the /var/opt/csm/csmdata directory.

New commands and improvements

The **csmstat** command provides a snapshot of cluster node reachability, power status, and network interface status. The command gathers this information for the specified nodes and displays the output. The default output is displayed by host name.

> **Note:** The **csmstat** command is not supported for nodes on IntelliStation workstations. It is not supported for Linux on pSeries clusters. Displayed LCD values are limited to 16 characters. The **rpower -l** (lowercase L) command displays the complete values.

Example 3-4 shows the output of the **csmstat** command.

Example 3-4 csmstat output

```
{csm_server.itso.ibm.com:root}/-> csmstat
--------------------------------------------------------------------------------
Hostname           HWControlPoint     Status    PowerStatus    Network-Interfaces
--------------------------------------------------------------------------------
p690_LPAR1.itso.~  hmcitso.itso.ibm~  off       on             unknown
p690_LPAR2.itso.~  hmcitso.itso.ibm~  off       on             unknown
```

There have been improvements on the **dsh** command in fanout processing. Now the command does not wait until all nodes in the first group are completed prior to replenishing with additional nodes.

New diagnostic probes

New diagnostic probes are shipped in the csm.diagnostics fileset and installed in the /opt/diagnostics/bin directory.

The following diagnostic probes are provided:

- Management Server Probe - checks for configuration problems after installing the CSM management server:
  - Verifies that all required packages are installed.
  - Verifies that all directories have been created and have correct permissions.
  - Verifies that CFM cronjob is active. If it is not active, it just issues a warning.
  - Calls new ibm.csm.predefined_conditions probe to verify predefined conditions and responses.
  - Calls new ibm.csm.predefined_nodegroups probe to verify predefined node groups.
- Hardware Control Probe - checks that the hardware control points and console servers are configured correctly and available:
  - Invokes new ibm.csm.HWCtrl.rconsole and ibm.csm.HWCtrl.rpower probes.
  - ibm.csm.HWCtrl.rconsole invokes the following new probes:
    - ibm.csm.HWCtrl.rconsole.attributes - verifies that console server attributes in the ManagedNode resource class are valid and self-consistent based on the ConsoleMethod type.
    - ibm.csm.HWCtrl.rconsole.connection - verifies that the console server is available.
  - ibm.csm.HWCtrl.rpower invokes the following new probes:
    - ibm.csm.HWCtrl.rpower.attributes - verifies that power control attributes in the ManagedNode resource class are valid and self-consistent based on the PowerMethod type.
    - ibm.csm.HWCtrl.rpower.connection - verifies that the hardware control point is available.

Node installation mount point

During the node installation process, the management server /csminstall directory is mounted. In previous versions of CSM, the mount point for the /csminstall directory was /mnt. In CSM 1.3.2, the mount point for the /csminstall directory is /var/opt/csm/mnt. This mount point directory is stored in the global variable $::CSMCLIENTMNTDIR in CSMDefs.pm. The following CSM scripts have been updated to use the new global variables:

- **csmfirstboot**
- **csmpreboot**
- **makenode**
- **rmnode**
- **smslocal**
- **smsupdatenode**
- **updatenode**
- **updatenode.client**

The KickStart configuration template files have been updated to reference the new mount point.

Note: If you are running **cmssetupks** with old or previously customized templates, you must retrofit the CSM 1.3.2 changes to prevent node installation failures.

3.3 HACMP

The following sections provide an overview of the benefits of using High Performance Switch in an HACMP environment.

3.3.1 HACMP overview

HACMP is a tool for building UNIX-based mission-critical computing platforms. HACMP software ensures that critical resources, such as applications, are available for processing. HACMP has two major components: high availability (HA) and cluster multi-processing (CMP).

The primary reason for creating HACMP clusters is to provide a highly available environment for mission-critical applications. For example, an HACMP cluster could run a database server program that services client applications. The clients send queries to the server program, which responds to their requests by accessing a database stored on a shared external disk.

In an HACMP cluster, the applications are put under HACMP control to ensure the availability of these applications. HACMP takes measures to ensure that the applications remain available to client processes even if a component in a cluster fails. To ensure availability, in case of a component failure, HACMP moves the application (along with resources that ensure access to the application) to another node in the cluster.

3.3.2 Role of HACMP

HACMP helps you with each of the following:

- The HACMP planning process and documentation include tips and advice on the best practices for installing and maintaining a highly available HACMP cluster.
- Once the cluster is operational, HACMP provides the automated monitoring and recovery for all of the resources on which the application depends.
- HACMP provides a full set of tools for maintaining the cluster while keeping the application available to clients.

HACMP lets you:

- Set up an HACMP environment using online planning worksheets to simplify the initial planning and setup.
- Ensure high availability of applications by eliminating single points of failure in an HACMP environment.
- Leverage high availability features available in AIX.
- Manage how a cluster handles component failures.
- Secure cluster communications.
- Set up fast disk takeover for volume groups managed by the Logical Volume Manager (LVM).
- Manage event processing for an HACMP environment.
- Monitor HACMP components and diagnose problems that may occur.

3.3.3 Physical components of an HACMP cluster

HACMP provides a highly available environment by identifying a set of resources essential to uninterrupted processing, and by defining a protocol that nodes use to collaborate to ensure that these resources are available. HACMP extends the clustering model by defining relationships among cooperating processors, where one processor provides the service offered by a peer should the peer be unable to do so.

An HACMP cluster is made up of the following physical components:

- Nodes
- Shared external disk devices
- Networks
- Network interfaces
- Clients

The HACMP software allows you to combine physical components into a wide range of cluster configurations, providing you with flexibility in building a cluster that meets your processing requirements. Figure 3-14 shows one example of an HACMP cluster. Other HACMP clusters could look very different, depending on the number of processors, the choice of networking and disk technologies, and so on.

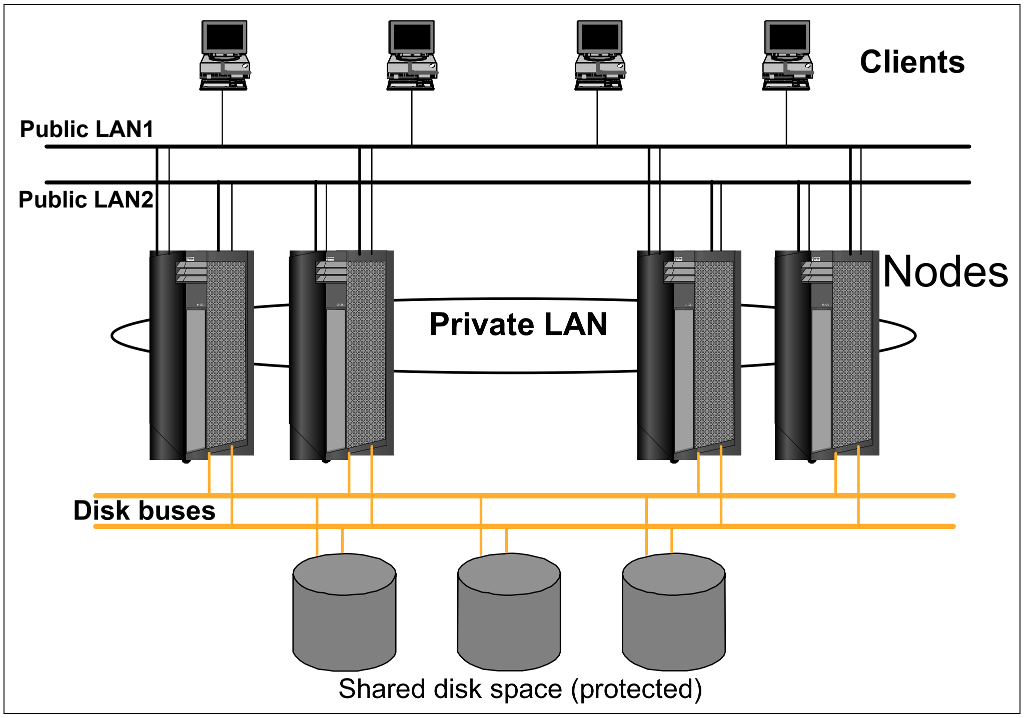


Figure 3-14   Example HACMP cluster scenario

Nodes

Nodes form the core of an HACMP cluster. A node is a processor that runs both AIX and the HACMP software. HACMP supports pSeries uniprocessor and symmetric multiprocessor (SMP) systems (including LPARs) and the Scalable POWERparallel processor (SP) systems as cluster nodes. To the HACMP software, an SMP system looks just like a uniprocessor. SMP systems provide a cost-effective way to increase cluster throughput. Each node in the cluster can be a large SMP machine, extending an HACMP cluster far beyond the limits of a single system and allowing thousands of clients to connect to a single database.

In an HACMP cluster, up to 32 nodes cooperate to provide a set of services or resources to other entities. Clustering these servers to back up critical applications is a cost-effective high availability option. A business can use more of its computing power while ensuring that its critical applications resume running after a short interruption caused by a hardware or software failure.

In an HACMP cluster, each node is identified by a unique name. A node may own a set of resources: disks, volume groups, file systems, networks, network addresses, and applications.

Typically, a node runs a server or a back-end application that accesses data on the shared external disks.

Shared external disk devices

Each node must have access to one or more shared external disk devices. A shared external disk device is a disk physically or logically (GPFS, VSD) connected to multiple nodes. The shared disk stores mission-critical data, typically mirrored or RAID-configured for data redundancy.

A node in an HACMP cluster must also have internal disks that store the operating system and application binaries, but these disks are not shared.

Depending on the type of disk used, HACMP supports two types of access to shared external disk devices: non-concurrent access and concurrent access.

- In non-concurrent access environments, only one connection is active at any given time, and the node with the active connection owns the disk. When a node fails, disk takeover occurs when the node that currently owns the disk leaves the cluster and a surviving node assumes ownership of the shared disk.
- In concurrent access environments, the shared disks are actively connected to more than one node simultaneously. Therefore, when a node fails, disk takeover is not required.

Networks

Cluster nodes communicate with each other over communication networks. If one of the physical network interface cards on a node on a network fails, HACMP preserves the communication to the node by transferring the traffic to another physical network interface card on the same node. If a “connection” to the node fails, HACMP transfers resources to another node to which it has access.

In addition, RSCT sends heartbeats between the nodes over the cluster networks to periodically check on the health of the cluster nodes themselves. If HACMP detects no heartbeats from a node, a node is considered as failed and resources are automatically transferred to another node.

Configuring multiple communication paths between the nodes in the cluster is highly recommended. Having multiple communication networks prevents cluster partitioning, in which the nodes within each partition form their own entity. In a partitioned cluster, it is possible that nodes in each partition could allow simultaneous non-synchronized access to the same data. This can potentially lead to different views of data from different nodes.

As an independent, layered component of AIX, the HACMP software is designed to work with any TCP/IP-based network. Nodes in an HACMP cluster use the network to allow clients to access the cluster nodes, enable cluster nodes to exchange heartbeat messages and, in concurrent access environments, serialize access to data.

The HACMP software has been tested with Ethernet, token-ring, ATM, and other networks.

The HACMP software defines two types of communication networks, characterized by whether these networks use communication interfaces based on the TCP/IP subsystem

(TCP/IP-based) or communication devices based on non-TCP/IP subsystems (device-based).

TCP/IP is a communications subsystem that lets you set up local area and wide area networks. TCP/IP provides facilities that make the computer system an Internet host that can attach to a network and communicate with other Internet hosts.

- TCP/IP-based network. Connects two or more server nodes and optionally allows client access to these cluster nodes, using TCP/IP. Ethernet, token-ring, ATM, HP switch and SP Switch networks are defined as TCP/IP-based networks.
- Device-based network. Provides a point-to-point connection between two cluster nodes for HACMP control messages and heartbeat traffic. Device-based networks do not use the TCP/IP protocol, and therefore continue to provide communications between nodes in the event the TCP/IP subsystem on a server node fails. Target mode SCSI devices, target mode SSA devices, disk heartbeat devices or RS232 point-to-point devices are defined as device-based networks.

IP Address Takeover

If the physical network interface card on one node fails, and if there are no other accessible physical network interface cards on the same network on the same node (and, therefore, swapping IP labels of these Network Interface Cards (NICs) within the same node cannot be performed), HACMP can use the IP Address Takeover (IPAT) operation.

IP Address Takeover is a mechanism for recovering a service IP label by moving it to another NIC on another node, when the initial NIC fails. IPAT is useful because it ensures that an IP label over which services are provided to the client nodes remains available.

HACMP 5.1 supports two methods of performing IPAT:

- IPAT via IP Aliasing (this is the default in HACMP 5.1)
- IPAT via IP Replacement (this is the traditional IPAT method)

Both methods are described in the sections that follow.

IPAT and Service IP Labels

Figure 3-2 summarizes how IPAT manipulates the service IP label.

Table 3-2 Service label IPAT

When IPAT via IP aliases is used	The service IP address/label is aliased onto the same network interface as an existing communications interface. That is, multiple IP addresses/labels are configured on the same network interface at the same time. In this configuration, all IP addresses/labels that you define must be configured on different subnets. This method can save hardware, but requires additional subnets.
When IPAT via IP Replacement is used	The service IP address/label replaces the existing IP address/label on the network interface. That is, only one IP address/label is configured on the same network interface at the same time. In this configuration, two IP addresses/labels on a node can share a subnet, while a backup IP address/label on the node must be on a different subnet. This method can save subnets but requires additional hardware.

IP Address Takeover via IP aliases

You can configure IP Address Takeover on certain types of networks using the IP aliasing network capabilities of AIX. Defining IP aliases to network interfaces allows creation of more than one IP label and address on the same network interface. IPAT via IP aliases utilizes the gratuitous ARP capabilities available on certain types of networks.

In a cluster with a network configured with IPAT via IP aliases, when the resource group containing the IP service label falls over from the primary node to the target node, the initial IP label that is used at boot time is never removed from the NIC on the target node. Service IP labels are added (and removed) as alias addresses on that NIC. Unlike IPAT via IP Replacement, this allows a single NIC to support multiple IP service labels placed on it as IP aliases. Therefore, the number of resource groups a node can host at a time is no longer limited to the number of physical network interfaces of that node.

If the IP configuration mechanism for an HACMP network is via IP aliases, the communication interfaces for that HACMP network must use routes that are different from the one used by the service IP address.

IPAT via IP aliases provides the following advantages over the IPAT via IP Replacement scheme:

- Running IP Address Takeover via IP aliases is faster than running IPAT via IP Replacement, because moving the IP address and the hardware address takes considerably longer than simply moving the IP address.
- IP aliasing allows co-existence of multiple service labels on the same network interface, resulting in fewer physical network interface cards in your cluster.

In HACMP 5.1, IPAT via IP aliases is the default mechanism for keeping a service IP label highly available.

IP Address Takeover via IP Replacement

The IP Address Takeover via IP Replacement facility moves the IP service label of a NIC (along with the IP address associated with it) on one node to a NIC on another node should the NIC on the first node fail. IPAT via IP Replacement ensures that the IP service label that is included as a resource in a resource group in HACMP is accessible through its IP address, no matter onto which physical network interface card this IP service label is currently placed.

If the IP address configuration mechanism is IP Replacement, only one communication interface for that HACMP network can use the route that is associated with the Service IP Address.

In conjunction with IPAT via IP Replacement (previously known as traditional IPAT) you can also configure Hardware Address Takeover (HWAT) to ensure that the mappings in the ARP cache are correct on the target adapter.

Clients

A client is a system that accesses the nodes in a cluster over a local area network. Clients each run a “front end” or client application that queries the server application running on the cluster node.

The HACMP software provides a highly available environment for critical data and applications on cluster nodes. The HACMP software does not make the clients themselves highly available. AIX clients can use the Client Information (Clinfo) services to receive notice of cluster events. Clinfo provides an API that displays cluster status information. The /usr/es/sbin/cluster/clstat utility, a Clinfo client shipped with the HACMP software, provides information about all cluster service interfaces.

3.4 IBM Virtual Shared Disks

This section contains a short description of the IBM Virtual Shared Disks 4.1 product in a pSeries High Performance Switch environment.

In this section we include the following topics:

- VSD overview
- Recovery scenarios
- Restrictions for VSD
- IBM Subsystem Device Driver (SDD) overview
- High Performance Switch considerations and tuning recommendations

3.4.1 Overview of the Virtual Shared Disk components

IBM Virtual Shared Disk is a subsystem that lets application programs that are running on different nodes of an RSCT peer domain access a raw logical volume as if it were local at each of the nodes. Each Virtual Shared Disk corresponds to a logical volume that is actually local at one of the nodes, which is called the server node.

The Virtual Shared Disk subsystem routes I/O requests from the other nodes, called client nodes, to the server node and returns the results to the client nodes. The I/O routing is done by the Virtual Shared Disk device driver that interacts with the AIX Logical Volume Manager (LVM). The device driver is loaded as a kernel extension on each node. Thus, raw logical volumes can be made globally accessible in the VSD nodeset (domain).

The application program interface to a Virtual Shared Disk is the raw device (or device special file). This means application programs must issue requests to a Virtual Shared Disk using the block size specified by the LVM (currently, requests are multiples of 512 bytes on 512-byte block boundaries). Figure 3-15 shows a logical view of the the VSD subsystem.

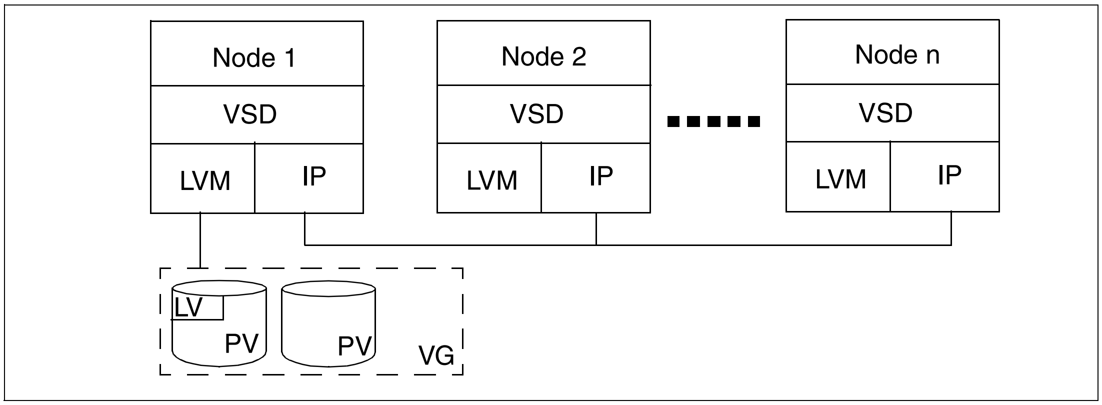


Figure 3-15 Logical view of the VSD subsystem

3.4.2 Product packaging

VSD used to be a component of the PSSP, and only SP customers could take advantage of it. VSD is now shipped as part of the RSCT Release 2.3, included with the AIX 5L operating system. It can also be installed as a PTF to AIX Version 5.2.B (AIX 5.2 maintenance level 2).

VSD packaging is described in Table 3-3 on page 103.

Table 3-3 VSD filesets

Fileset Name	Description
rsct.vsd.vsdd	VSD device driver
rsct.vsd.rvsd	Recoverable VSD
rsct.vsd.cmds	VSD commands
rsct.vsd.vsdrm	VSD resource manager

3.4.3 Configuration repository

In previous releases of VSD, the configuration data for nodes and drives was stored in the SDR database of PSSP. Starting with Version 4.1, VSD comes with RSCT. All configuration data is stored in appropriate RSCT classes.

The VSD product brings four new classes to the RSCT repository:

- IBM.vsdnode
- IBM.vsdtable
- IBM.vsdgvg
- IBM.rvsdrestrictlevel

You can gather the information on these subclasses with the **lsrsrc -ls** command. You tipically will not change entries in these classes manually. They are managed by appropriate VSD commands. These entries can be modified using the standard RSCT command **chrsrc**.

IBM.vsdnode

This class contains node-related information such as the communication interface and IP address. Each node has its own stanza in this class. Example 3-5 is a sample of the class contents.

Example 3-5 Contents of IBM.vsdnode class

```
resource 1:
        VSD_nodenum              = 1
        VSD_adapter              = "ml0"
        VSD_min_buddy_buffer_size = 4096
        VSD_max_buddy_buffer_size = 262144
        VSD_max_buddy_buffers    = 128
        VSD_maxIPmsgsz           = 61440
        RVSD_version             = 0
        CVSD_cluster_name        = ""
        CVSD_node_number         = 0
        cvgs_defined             = 0
        VSD_ipaddr               = "10.10.10.11"
        NodeNameList             = {"p690_LPAR1.itso.ibm.com"}
resource 2:
        VSD_nodenum              = 2
        VSD_adapter              = "ml0"
        VSD_min_buddy_buffer_size = 4096
        VSD_max_buddy_buffer_size = 262144
        VSD_max_buddy_buffers    = 128
        VSD_maxIPmsgsz           = 61440
        RVSD_version             = 0
        CVSD_cluster_name        = ""
        CVSD_node_number         = 0
        cvgs_defined             = 0
        VSD_ipaddr               = "10.10.10.12"
        NodeNameList              = {"p690_LPAR2.itso.ibm.com"}
...
```

IBM.vsdtable

This class contains information about VSD storage objects. Each VSD storage object has its own entry as a separate stanza. See Figure 3-6 for a sample of the class contents.

Example 3-6 Contents of IBM.vsdtable class

```
resource 1:
        VSD_name           = "gpfs0vsd"
        global_group_name  = "gpfs0gvg"
        logical_volume_name = "gpfs0lv"
        minor_number       = 5
        size_in_MB         = 43840
        lv_blk0_pdev       = 3342340
        lv_blk0_pbn        = 4352
resource 2:
...
```

IBM.vsdgvg

This class contains data related to global VSD volume groups. Example 3-7 shows the sample contents of this class.

Example 3-7 Contents of IBM.vsdgvg class

```
resource 1:
        global_group_name = "gpfs0gvg"
        local_group_name  = "gpfs0vg"
        primary_node      = 1
        secondary_node    = 2
        eio_recovery      = 1
        recovery          = 0
        primary_ts        = ""
        secondary_ts      = ""
        server_list       = "0"
        vsd_type          = "VSD"
resource 2:
...
```

IBM.rvsdrestrictlevel

This class contains usually one entry describing the level of VSD software agreed by all nodes in the cluster. Example 3-8 shows the entry for VSD 4.1.

Example 3-8 Contents of IBM.rvsdrestrictlevel class

```
resource 1:
        level = "4010000"
```

3.4.4 Shared external disk access

The Virtual Shared Disk subsystem supports two methods of external disk access: serial (non-concurrent) and concurrent. Figure 3-16 on page 105 presents an example VSD network implementation.

Serial (non-concurrent) access

In a non-concurrent environment, only one node has access to a shared external disk at a given time. A primary server and a backup server are defined.

Concurrent access

Concurrent disk access allows you to use multiple servers to satisfy disk requests by taking advantage of the concurrent disk access environment supplied by AIX. To use this environment, Virtual Shared Disk uses the AIX services of Concurrent LVM (CLVM), which provides the synchronization of LVM and the management of concurrency for system administration services.

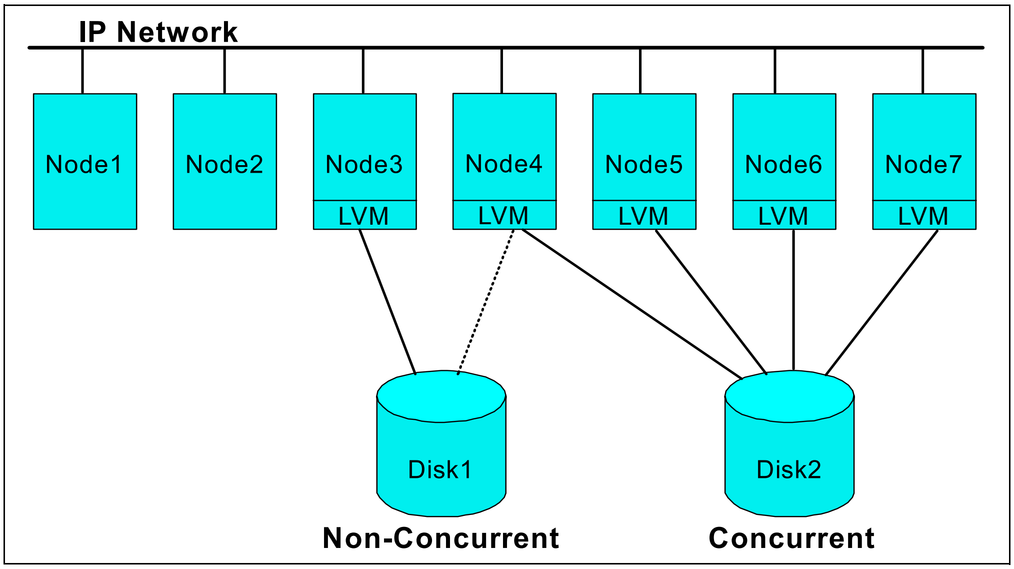


Figure 3-16 A Virtual Shared Disk IP network implementation

3.4.5 Quorum

The RVSD subsystem uses the notion of quorum, which is the majority of the Virtual Shared Disk nodes, to cope with communication failures. If the nodes in an RSCT peer domain are divided by a network failure, so that the nodes in one group cannot communicate with the nodes in the other group, the RVSD subsystem uses the quorum to decide which group continues operating and which group is deactivated.

- Once VSD node quorum is reached on startup, the RVSD subsystem activates all Virtual Shared Disks with servers in the active group.
  - All Virtual Shared Disks on the active nodes that also have a server in the active group will be in the active state.
  - If a Virtual Shared Disk’s primary server node is in the active group, the primary will be the server for that Virtual Shared Disk.
  - If a primary server for a Virtual Shared Disk is not in the active group but the secondary server is, the secondary will be the server for that Virtual Shared Disk.
  - All Virtual Shared Disks without a server in the active group will be in the stopped state.
- If quorum is lost, all the Virtual Shared Disks are put into the stopped state. When quorum is active again, the Virtual Shared Disks are put back into appropriate states based on the list.

Two-node support for VSD

In previous releases of VSD, in order to maintain quorum on a two-node cluster, an SP control workstation was required. VSD now supports “true” two-node support. Two-node systems can only provide node recovery if there are at least two functioning (non-CVSD) shared volume groups between the two nodes. Each node must be a primary server for a volume group and a backup server for a volume group.

3.4.6 IBM Recoverable Virtual Shared Disk overview

The IBM Recoverable Virtual Shared Disk (RVSD) subsystem is an integral component of the IBM Virtual Shared Disk product. It provides recoverability of your Virtual Shared Disks if a node, adapter, or disk failure occurs. The RVSD subsystem manages your Virtual Shared Disks, and, when an error is detected, will automatically switch disk access to an active node. Recovery is transparent to applications and there is no disruption of service except for a slight delay while takeover occurs. Figure 3-17 shows the logical view of the RVSD subsystem.

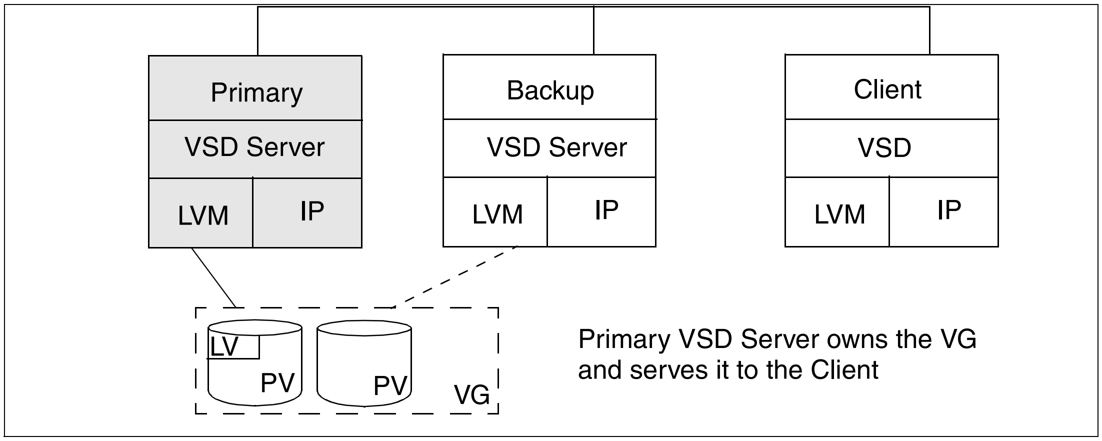


Figure 3-17 RVSD diagram

The RVSD subsystem controls recovery for the Recoverable Virtual Shared Disk component of RSCT. It invokes the recovery scripts whenever there is a change in the group membership. The **ha.vsd** command controls the RVSD subsystem. When a node goes down or a disk adapter or cable fails, the RVSD subsystem notifies all surviving processes in the remaining Virtual Shared Disk nodes so they can begin recovery. If a node fails, recovery involves switching the ownership of a serial-access (non-concurrent) disk to the secondary node. If a disk adapter or cable fails, recovery involves switching the server node for a volume group to the secondary node. When the failed component comes back up, recovery involves switching the disk or volume group ownership back to the primary node.

Communication adapter (**snx, enx**) failures are treated in the same manner as node failures. Recovery for non-concurrent volume groups consists of switching to the secondary server.

Important: When using VSD with the pSeries High Performance Switch, the ml0 adapter should be used to avoid outages due to link failures. For more information regarding High Performance Switch adapters, refer to 2.1.4, “Switch Network Interfaces” on page 26.

In “standard” VSD, the primary node is a node that is physically connected to a set of serial-access Virtual Shared Disks and will always manage them if it is active. The secondary node is a node that is physically connected to a set of serial-access Virtual Shared Disks and will manage them only if the primary node becomes inactive. For concurrent Virtual Shared Disks, there is no concept of primary and secondary nodes. Instead, there is simply a list of servers that can concurrently access the disks. The RVSD subsystem uses the notion of quorum, the majority of the Virtual Shared Disk nodes, to cope with communication failures. If the nodes in an RSCT peer domain are divided by a network failure, so that the nodes in one group cannot communicate with the nodes in the other group, the RVSD subsystem uses the quorum to decide which group continues operating and which group is deactivated.

Table 3-4 shows how the daemons in the nodes in an RSCT peer domain react as inactive nodes come back up and as active nodes fail. The table shows the changes that affect three of the nodes in a system that has more than three nodes.

Table 3-4 Recovery actions when nodes fail

nodes and clients	node is active	node is inactive (recovery scenario)
node 1: primary	► Daemons running on the other nodes in the RSCT peer domain accept node1 into the group. ► All Virtual Shared Disks on node1 become active. ► All clients designate node1 as the manager for the node1 Virtual Shared Disks and designate those Virtual Shared Disks as active.	► All Virtual Shared Disks defined on node1 are put into suspended state on all clients. ► Daemons running on the other nodes in the RSCT peer domain remove node1 from the group. ► Node2, the secondary node, takes over the management of node1’s Virtual Shared Disks. ► All clients designate node2 as the server for node1’s Virtual Shared Disks and put those Virtual Shared Disks into active state.
node 2: secondary to node1	► Daemons running on the other nodes in the RSCT peer domain accept node2 into the group. ► If node1 is active, there is no change to the status of the node1 Virtual Shared Disks. ► If node1 is inactive, node2 takes over the management of the node1 Virtual Shared Disks.	► If node1 is active, there is no change to the status of its Virtual Shared Disks. ► If node1 is inactive, the node1 Virtual Shared Disks are put into stopped state on all clients. They remain in stopped state until node1 or node2 comes back up. ► Daemons running on the other nodes in the RSCT peer domain remove node2 from the group.
node 3: client	► Daemons running on the other nodes in the RSCT peer domain accept node3 into the group. ► All Virtual Shared Disks on active nodes for which node3 is a client are put into active state from node3’s point of view.	► All Virtual Shared Disks defined on node3 are put into stopped state from node3’s point of view. ► Daemons running on the other nodes in the RSCT peer domain remove node3 from the group.

3.4.7 Recovery for concurrent Virtual Shared Disks

Recovery is very similar to the non-concurrent disk example. The differences are:

- When one of the nodes fails, the RVSD subsystem uses hardware fencing to fence the failing node from accessing the disk, and client nodes will simply access the disk using the surviving node.

- The surviving node does not have to vary on the volume group because it is already online to the node.

3.4.8 Disk cable and disk adapter failures

Hardware interruptions at the disk are known as EIO errors. The Recoverable Virtual Shared Disk component can perform recovery by volume group from some kinds of EIO errors, for example, disk cable and disk adapter failures. These failures can affect some of the volume groups defined on a node without affecting other volume groups.

The Recoverable Virtual Shared Disk component switches the server function from the primary node to the secondary node for the failed volume groups on the node, without changing the server for those volume groups that have not failed. This involves failing over the failed volume groups to the newly defined primary server and retrying the I/O request that previously failed.

The attempt to switch the servers only takes place if there has not been an EIO error on the volume group within approximately the last seven minutes. If EIO recovery fails, the related Virtual Shared Disks are placed in the stopped state.

You can tell if an EIO recovery has happened by using the **`vsdatalst -g`** command and looking at the recovery field in the results. If it contains a value other than zero, recovery has taken place at some point.

> **Note:** If sporadic EIO errors are occurring, EIO recovery will keep switching the server for that volume group. This EIO recovery does not apply to concurrent Virtual Shared Disks. For these disks, an EIO error will simply cause the clients to stop accessing the disks through the node that received the EIO error. Occasionally, an I/O request will be sent through the failing node to see if the EIO was only a temporary error.

3.4.9 Communication adapter failures

Communication adapter failure is supported for the pSeries High Performance Switch. When communication adapter recovery is enabled, an adapter failure is promoted to a recoverable Virtual Shared Disk node failure so that non-concurrent volume groups can fail over to the secondary server. For concurrent volume groups, this provides access to the disks through another server.

Only non-concurrent and concurrent volume groups on nodes connected by the pSeries High Performance Switch will recover from adapter failure.

Communication adapter recovery is enabled by default but can be disabled by issuing the **`ha.vsd adapter_recovery off`** command. Use this command when you have supported communication adapters and do not want their failures promoted to node failures.

Issue **`ha.vsd query`** to determine whether adapter recovery is enabled or disabled. The output will be similar to the Figure 3-9, where adapter_recovery can be on or off, and adapter_status can be up, down, or unknown.

Example 3-9 Adapter recovery status

```
LPAR1# ha.vsd query
Subsystem         Group            PID          Status
 rvsd             rvsd             180454       active
 rvsd(vsd): quorum= 1/2, active=1, state=idle, isolation=member,
            NoNodes=2, lastProtocol=fence_protocol,
            adapter_recovery=on, adapter_status=up,
            RefreshProtocol has never been issued from this node,
            Running function level 4.1.0.0.
```

3.4.10 Technical large pages overview

SNI supports technical large pages, which help provide high performance by decreasing the amount of address translation while copying, sending, or receiving large blocks of data. The SNI's IP interface can use technical large pages for backing its send and receive pools. SNI also supports parallel applications that choose to use technical large pages for communications data areas. Such applications can send and receive data directly to and from technical large pages. For more information about using large pages, see the section on large page support in *AIX 5L Version 5.2 Performance Management Guide*.

Application communication

All applications on the machine can use the SNI. The following communication protocols are supported on the SNI:

- Standard TCP/IP communication through AIX sockets or message-passing libraries
- Dedicated user-space access through message-passing libraries

A distinct communication port between an application on the server and the switch is called a window. Each window has its own send FIFO and receive FIFO as well as a set of variables that describes the status of the FIFOs. The variables are used to transfer data to and from the window's FIFOs and the SNI device.

Table 3-5 describes the different types of windows.

Table 3-5 Types of windows

IP	This reserved window is responsible for the IP communication among nodes.
Service	This reserved window manages configuration and monitoring of the SNI.
VSD	This window is reserved for VSD communication if you install VSD on your system, and uses LAPI or KLAPI.
User Space	These windows permit high-speed data communication among user applications.

3.4.11 Restrictions for using Virtual Shared Disks

- The maximum number of Virtual Shared Disks that can be defined in an RSCT peer domain is 10000 (10K).
- Restrictions on AIX logical volumes also affect Virtual Shared Disks.
- HACMP and the Virtual Shared Disk subsystem both use the Concurrent LVM functions. To prevent conflicts, only one product may manage concurrent disks. A Virtual Shared Disk server will not be able to be defined as part of a concurrent volume group cluster if HACMP is installed and is already managing concurrent volume groups. If HACMP is installed but not managing concurrent volume groups, then Concurrent Virtual Shared Disks can be defined.
- The Virtual Shared Disk subsystem is dependent on the Recoverable Virtual Shared Disk (RVSD) subsystem. You cannot run the Virtual Shared Disk subsystem without the RVSD subsystem.

- Concurrent Virtual Shared Disks are supported for SSA (Serial Storage Architecture) disks and disks that support the SCSI Persistent Reserve model implemented by the AIX SCSI device drivers.
  - SSA concurrent shared disks are restricted to two servers.
  - Persistent Reserve concurrent shared disks are restricted to 16 servers.
- Concurrent Virtual Shared Disks are not supported for ESS subsystems with the Subsystem Device Driver (SDD), since it does not comply with the SCSI Persistent Reserve model. Non-concurrent Virtual Shared Disks can be used with SDD instead.
- Concurrent Virtual Shared Disk support has the same limitations as Concurrent LVM; there is no *mirror write consistency* or *bad block relocation*.
- The Virtual Shared Disk subsystem does not support the use of the secure remote command (ssh) for copying files to, and running commands on, remote machines.
- Two-node systems can provide node recovery only if there are at least two functioning (non-concurrent Virtual Shared Disk) shared volume groups between the two nodes. Each node must be a primary server for a volume group and a backup server for a volume group.
- Do not put the root volume group (rootvg) or any other volume group that contains bootable logical volumes on a shared disk.
- Do not use Virtual Shared Disk volume groups with non-Virtual Shared Disk applications. Applications that access data on non-Virtual Shared Disk logical volumes (a JFS file system, for example) will lose access to the logical volume when the Virtual Shared Disk server node changes and can interfere with recovery.
- Do not put a tape drive and a multi-host attached disk on the same SCSI bus.
- The Virtual Shared Disk subsystem requires that you use ml0, the multilink IP address, as the Virtual Shared Disk adapter name, when using the IBM @server pSeries High Performance Switch (HPS). The data is striped across the available adapters. If an adapter fails, communication continues across any remaining available adapters.

3.4.12 Overview of the subsystem device driver (SDD)

The Subsystem Device Driver (SDD) is an IBM Enterprise Storage Server® (ESS) device driver that provides:

- Enhanced data availability
- Automatic path failover and recovery to an alternate path
- Dynamic load balancing of multiple paths
- Concurrent microcode upgrade.

When redundant paths are configured to ESS logical units, and the SDD is installed and configured, the AIX **`lspv`** command shows multiple *hdisks* as well as a new construct called a *vpath*. The hdisks and vpaths represent the same logical unit. You will need to use the **`lsvpcfg`** command to get more information. For example, after issuing **`lspv`**, you see output similar to Example 3-10.

Example 3-10 **`lspv`** *output*

```
hdisk0          0022be2ab1cd11ac                    rootvg          active
hdisk1          0022be2a3d02ead0                    None
hdisk2          0022be2a4cbbafd8                    None
hdisk3          0022be2a470fdfb5                    None
hdisk4          none                                None
hdisk5          none                                None
vpath0          none                                None
```

```
vpath1          0022be2a31fa63ca                  group1lparvg    active
```

After issuing **lsvpcfg**, you see output similar to Example 3-11.

Example 3-11 lsvpcfg output

```
vpath0 (Avail ) 30022513 = hdisk3 (Avail pv )
vpath1 (Avail pv group1lparvg) 30122513 = hdisk4 (Avail ) hdisk5 (Avail )
```

The examples above illustrate some important points:

- `vpath0` consists of a single path (`hdisk3`) and therefore will not provide failover protection. Also, hdisk3 is defined to AIX as a physical volume (`pv` flag) and has a PVID, as you can see from the output of the **lspv** command.
- `vpath2` has two paths (`hdisk4` and `hdisk5`) and has a volume group defined on it. Notice that with the **lspv** command, hdisk4 and hdisk5 look like newly installed disks with no PVIDs. The **lsvpcfg** command had to be used to determine that hdisk4 and hdisk5 make up vpath2, which has a PVID.

3.4.13 VSD configuration support for the SDD

The Virtual Shared Disk subsystem supports Virtual Shared Disks defined in Subsystem Device Driver (SDD) volume groups, which are also referred to as volume groups. To exploit the functions provided by the SDD (including automatic failover), the shared disk volume groups must be created as, or converted to, vpath volume groups.

You can configure Virtual Shared Disks to use the SDD in the following ways:

- Use the **dpovgfix volgrp** or the **hd2vp volgrp** commands to convert an existing hdisk volume group to a *vpath volume group*. These commands remove the PVID from the existing hdisk paths and activate the vpath volume group.
- Use the **createvsd** command. Both hdisks and vpaths cannot be specified on the same invocation.
- Use the SDD **mkvg4vp** command to build volume groups. Next, use the AIX LVM commands to build logical volumes, and then use IBM VSD commands to associate the LVM constructs with IBM VSD constructs.

> **Important:** If you use SDD volume groups (*vpaths*), all nodes that access those volume groups must access them as vpath volume groups. For example, you cannot have one node accessing a *normal* volume group and another node accessing the same volume group as a *vpath* volume group.

3.4.14 HPS considerations when using the IP protocol for data transmissions

This section presents some tuning considerations when using the IP protocol for data transmission over the IBM HPS.

IP message size

If you configure the Virtual Shared Disk nodes to use the pSeries High Performance Switch (ml0), set the maximum *IP message size* (utilized by the Virtual Shared Disk driver) to 61440 (60 KB). Note that the value you assign to *maximum_buddy_buffer_size* also limits the maximum size of the request that the Virtual Shared Disk subsystem sends across the nodes (For more information regarding buddy buffers, refer to “Buddy buffers” on page 113).

For example, if you have:

- A request from a client to write 256 KB of data to a remote Virtual Shared Disk
- A maximum buddy buffer size of 64 KB
- A maximum IP message size of 60 KB

The following transmission sequence occurs:

1. The Virtual Shared Disk subsystem divides the 256 KB of data into four 64 KB requests in four buddy buffers.
2. Each 64 KB block of data becomes one 60 KB packet and one 4 KB packet for transmission to the server via IP.
3. At the server, the eight packets are reassembled into four 64 KB blocks of data, each in a 64 KB buddy buffer.
4. The server then has to perform four 64 KB write operations and return four acknowledgments to the client.

A better scenario for the same write operation would use the maximum buddy buffer size:

- The same 256 KB client request to the remote Virtual Shared Disk
- The maximum buddy buffer size of 256 KB
- The maximum IP message size of 60 KB

Producing the following transmission sequence:

1. The 256 KB request becomes four 60 KB packets and one 16 KB packet for transmission to the server through IP.
2. At the server, the five packets are reassembled into one 256 KB block of data in a single buddy buffer.
3. The server then performs one 256 KB write operation and returns an acknowledgment to the client.

The second scenario is preferable to the first because the I/O operations at the server are minimized.

The switch pool

If you are running the Virtual Shared Disk subsystem over IP, you should set the send and receive pool sizes to at least 16MB.

To check the current sizes of the send and receive pools, type:

```
lsattr -El snix
```

where `snix` identifies the SNI adapter on the node.

The default size for each pool is 524288 bytes (512 KB). We recommend setting this to 16 MB.

To change the sizes of the pools to 32 MB, type:

```
/usr/lib/methods/chgsni -l snix -a spoolsize=33554432
/usr/lib/methods/chgsni -l snix -a rpoolsize=33554432
```

where `snix` identifies the SNI adapter on the node.

Buddy buffers

The buddy buffer is pinned kernel memory. The Virtual Shared Disk server node uses the buddy buffer to temporarily store data for I/O operations originating at a client node. The data in a buddy buffer is purged immediately after the I/O operation completes.

The values associated with the buddy buffer are:

- Minimum buddy buffer size allocated to a single request
- Maximum buddy buffer size allocated to a single request
- Total number of maximum-size buddy buffers that the system will attempt to dynamically allocate

We recommend using the following values:

- Minimum buddy buffer size: 4096 (4 KB)
- Maximum buddy buffer size: 262144 (256 KB)
- Total number of maximum-size buffers:
  - 2000 per Virtual Shared Disk server node
  - One per Virtual Shared Disk client-only node

> **Note:** If your application uses the fastpath option of asynchronous I/O, the maximum buddy buffer size must be greater than or equal to 128 KB. Otherwise, you will receive EMSGSIZE "`Message too long`" errors.

You can either set these values using the **`vsdnode`** command, or update the values using the **`updatevsdnode`** command. We suggest, however, that you configure the recommended amount of buddy buffer space, unless your system is memory-constrained and you want to restrict the amount of buddy buffer space.

When the device driver is configured, the total buddy buffer space is not pinned; instead, approximately one-quarter of the total space requested is pinned when the device driver is configured. This initial amount of spaced pinned is limited to a maximum of 64 MB for a 32-bit kernel, or 128 MB for a 64-bit kernel. After configuration, the device driver attempts to dynamically expand and contract additional buddy buffer space up to the maximum specified, or until AIX can no longer satisfy the memory request. If a buddy buffer cannot be obtained, then the request is queued at the Virtual Shared Disk server until a buddy buffer is available.

Buffer allocation

Your application should make all new allocated buffers on the page boundary. If your I/O buffer is not aligned on a page boundary, the Virtual Shared Disk device drivers will not parallelize I/O requests to underlying Virtual Shared Disks and performance will be degraded.

Maximum I/O request size

The following factors limit the block size that the Virtual Shared Disk subsystem uses to process each I/O request:

- The largest block size the Virtual Shared Disk subsystem will use is the *max_buddy_buffer_size* as specified on the **`vsdnode`** or **`updatevsdnode`** command.
- If the Virtual Shared Disk uses the switch as its adapter, the *max_IP_msg_size* (as specified on the **`vsdnode`** or **`ctlvsd`** command) that could be sent is 65024 bytes (63.5 KB). We suggest the value 61440 (60 KB) for the Virtual Shared Disk device driver. The **`ctlvsd -M`** command can override the default. The *vsd_max_IP_msg_size* should be set to a value that is a multiple of 512 bytes and is less than or equal to 63.5 KB (when the switch is

used) and less than or equal to 24 KB (when the switch is not used). The **statvsd** command displays the current value.

> **Important:** Setting the *max_IP_msg_size* to more than 24 KB when using communication adapters with small MTU (maximum transmission unit) could overflow the adapter driver's internal buffers, causing the IP layer to drop packets. This forces the Virtual Shared Disk device driver to retry, sometimes without success, resulting in a timeout.

The atomicity of an I/O operation is gated by the size of the Virtual Shared Disk request, rather than the size of the application request (if the Virtual Shared Disk request is smaller than the application request the application request would be split down to the size of the Virtual Shared Disk request). The atomicity of an I/O operation is also gated by the maximum buddy buffer size.

3.5 General Parallel File System 2.1

In this section we describe the GPFS product.

3.5.1 GPFS overview

The IBM General Parallel File System (GPFS) allows users shared access to files that may span multiple disk drives on multiple nodes. It offers many of the standard UNIX file system interfaces, allowing most applications to execute without modification or recompiling. UNIX file system utilities are also supported by GPFS. That is, users can continue to use the UNIX commands they have always used for ordinary file operations. The only unique commands are those for administering the GPFS. GPFS provides file system services to parallel and serial applications. Figure 3-18 on page 115 shows a logical view of a GPFS filesystem.

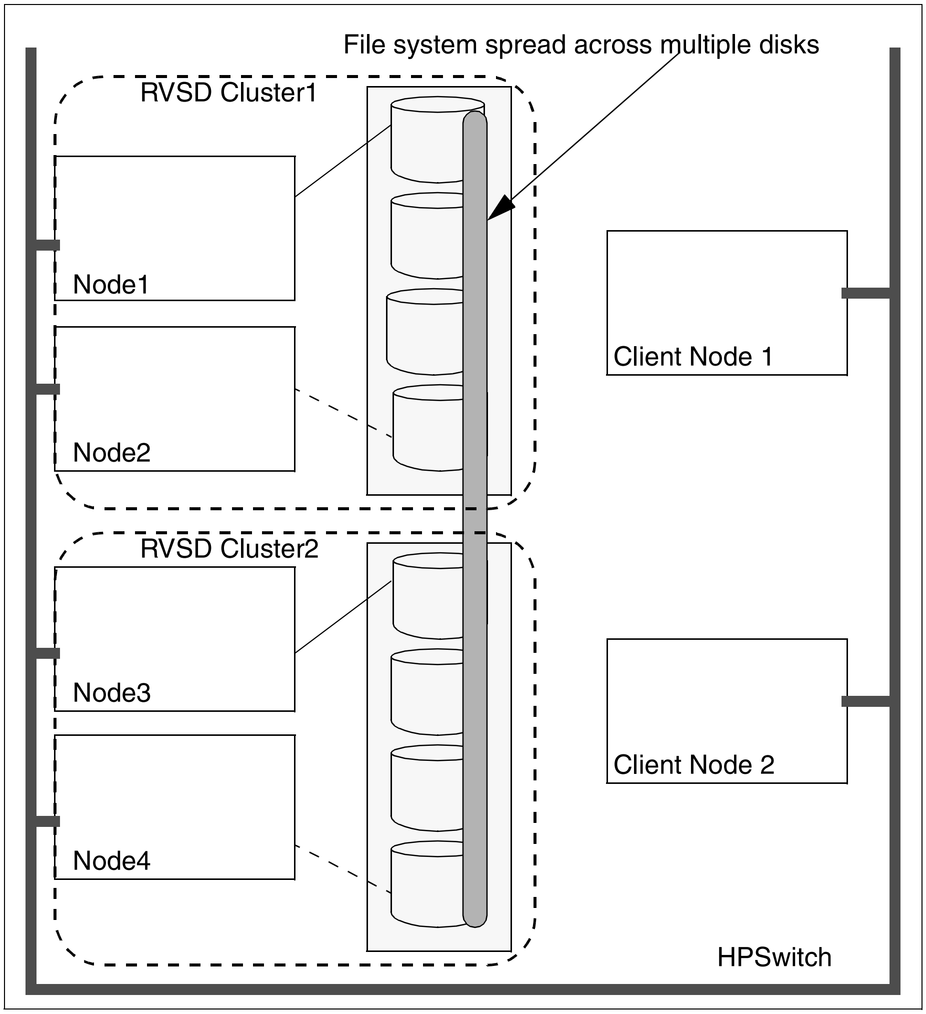


Figure 3-18 GPFS diagram

GPFS allows parallel applications simultaneous access to the same files, or different files, from any node in the GPFS nodeset while managing a high level of control over all file system operations. A nodeset is a group of nodes that all run the same level of GPFS and operate on the same file system. GPFS is particularly appropriate in an environment where the aggregate peak need for data exceeds the capability of a distributed file system server. It is not appropriate for those environments where hot backup is the main requirement or where data is readily partitioned along individual node boundaries.

GPFS 2.1 with APAR IY47306 and the pSeries High Performance Switch, provides support for IBM Virtual Shared Disks in an RSCT peer domain. Virtual Shared Disks may only be used in new GPFS nodesets and filesystems. The use of Virtual Shared Disks in an existing GPFS nodeset or file system in an RSCT peer domain is not supported.

3.5.2 GPFS 2.1 (APAR IY47306) new features

GPFS 2.1 provides several usability enhancements:

- The ability to create a GPFS cluster from an RSCT peer domain.
- Faster failover through the persistent reserve feature.
- Support for the latest IBM @server Cluster 1600 configuration.

- The GPFS for AIX 5L product may be installed in either an AIX cluster environment or a PSSP cluster environment. Consequently, two sets of man pages are now shipped with the product and you must set your MANPATH environment variable accordingly (see the *General Parallel File System for AIX 5L in an RSCT Peer Domain: Concepts, Planning, and Installation Guide*, GA22-7940-00, in the section on installing the GPFS manual pages).
- 64-bit kernel exploitation The GPFS kernel extensions are now shipped in both 32-bit and 64-bit formats.
- Electronic license agreement
- Two new commands for managing the disks (logical volumes) in your GPFS cluster:
  - **mmdellv**
  - **mmlsgpfsdiskv**
- For *atime* and *mtime* values as reported by the **stat**, **fstat**, **gpfs_stat**, and **gpfs_fstat** calls, you may:
  - Suppress updating the value of *atime*. When suppressing the periodic update, these calls will report the time the file was last accessed when the file system was mounted with the `-S` no option or, for a new file, the time the file system was created.
  - Display the exact value for *mtime*. The default is to periodically update the *mtime* value for a file system. If it is more desirable to display exact modification times for a file system, specify the *-E yes* option.
- Commands that have been updated:

  mmcrfs
  mmchfs
  mmlsfsv

- The capability to read from or write to a file with direct I/O. The **mmchattr** command has been updated with the *-D* option for this support.
- The default use designation for nodes in your GPFS nodeset has been changed from manager to client. Commands that have been updated:

  mmconfig
  mmchconfigv

- The terms to *install/uninstall* GPFS quotas have been replaced by the terms *enable/disable* GPFS quota management.
- The GPFS documentation is no longer shipped on the product CD-ROM. You may download, view, search, and print the supporting documentation for the GPFS program product in the following ways:

1. In PDF format, on the World Wide Web at `http://www.ibm.com/servers/eserver/pseries/library/gpfs.html`
2. In hardcopy (paper) format from the IBM Publications Center at `http://www.ibm.com/shop/publications/order`
3. In HTML format at `http://www.publib.boulder.ibm.com/clresctr/windows/public/gpfsbooks.html`

New file system functions existing in GPFS 2.1 are not usable in existing file systems until you explicitly authorize these changes by issuing the **mmchfs -V** command.

3.5.3 Benefits of using GPFS

The benefits of using the GPFS product are discussed in the following sections.

Improved system performance

Using GPFS to store and retrieve your files can improve system performance by:

- Allowing multiple processes or applications on all nodes in the nodeset simultaneous access to the same file using standard file system calls.
- Increasing aggregate bandwidth of your file system by spreading reads and writes across multiple disks.
- Balancing the load evenly across all disks to maximize their combined throughput. One disk is no more active than another.
- Supporting large amounts of data.
- Allowing concurrent reads and writes from multiple nodes. This is a key concept in parallel processing.

Assured file consistency

GPFS uses a sophisticated token management system to provide data consistency while allowing multiple independent paths to the same file by the same name from anywhere in the system. Even when nodes are down or hardware resource demands are high, GPFS can find an available path to file system data.

High recoverability and increased data availability

GPFS is a logging file system that creates separate logs for each node. These logs record the allocation and modification of metadata aiding in fast recovery and the restoration of data consistency in the event of node failure.

GPFS failover support allows you to organize your hardware into a number of failure groups to minimize single points of failure. A failure group is a set of disks that share a common point of failure that could cause them all to become simultaneously unavailable. In order to assure file availability, GPFS maintains each instance of replicated data on disks in different failure groups.

The replication feature of GPFS allows you to determine how many copies of a file to maintain. File system replication assures that the latest updates to critical data are preserved in the event of disk failure. During configuration, you assign a replication factor to indicate the total number of copies you wish to store. Replication allows you to set different levels of protection for each file or one level for an entire file system. Since replication uses additional disk space and requires extra write time, you might want to consider replicating only file systems that are frequently read from but seldom written to. Even if you do not specify replication when creating a file system, GPFS automatically replicates recovery logs in separate failure groups.

Once your file system is created, you can have it automatically mounted whenever the GPFS daemon is started. The automount feature assures that whenever the system and disks are up, the file system will be available.

Enhanced system flexibility

With GPFS, your system resources are not frozen. You can add or delete disks while the file system is mounted. When the time is right and system demand is low, you can rebalance the file system across all currently configured disks. You can also add new nodes without having to stop and restart the GPFS daemon. An exception to this applies when single-node quorum is in effect. After GPFS has been configured for your system, depending on your applications, hardware, and workload, you can reconfigure GPFS to increase throughput.

Simplified administration

GPFS commands save configuration and file system information in one or more files, collectively known as GPFS cluster data. The GPFS administration commands are designed to keep these files synchronized with each other and with the GPFS system files on each node in the nodeset, thereby ensuring accurate configuration data. GPFS administration commands are similar in name and function to UNIX file system commands, with one important difference: the GPFS commands operate on multiple nodes. A single GPFS command performs a file system function across the entire nodeset. Most GPFS administration tasks can be performed from any node running GPFS.

3.5.4 The basic GPFS structure

GPFS is a clustered file system defined over a number of nodes. The overall set of nodes on which GPFS runs is known as a GPFS cluster. Depending on the operating environment, GPFS defines several cluster types, as specified in Table 3-6.

Table 3-6 GPFS cluster types

Cluster type	Environment
SP	The PSSP cluster environment is based on the IBM Parallel System Support Programs (PSSP) program product and the shared disk concept of the IBM Virtual Shared Disk program product.
RPD	The rpd cluster environment is based on Reliable Scalable Cluster Technology (RSCT) peer domain created by the RSCT subsystem of AIX 5L. With an RSCT peer domain, all nodes in the GPFS cluster have the same view of the domain and share the resources within the domain (GPFS cluster type *rpd*).
HACMP	The hacmp cluster environment is based on HACMP cluster created by the HACMP program product (GPFS cluster type *hacmp*).
lc	The loose cluster environment is based on the Linux operating system.

On each node in the cluster, GPFS consists of:

- Administration commands
- A kernel extension
- A multi-threaded daemon

The GPFS kernel extension

The GPFS kernel extension provides the interfaces to the operating system vnode and virtual file system (VFS) interfaces for adding a file system. GPFS kernel extensions exist in both 32-bit and 64-bit forms. Structurally, applications make file system calls to the operating system, which presents them to the GPFS file system kernel extension. In this way, GPFS appears to applications as just another file system. The GPFS kernel extension will either satisfy these requests using resources that are already available in the system, or send a message to the GPFS daemon to complete the request.

The GPFS daemon

The GPFS daemon performs all I/O and buffer management for GPFS. This includes read-ahead for sequential reads and write-behind for all writes not specified as synchronous. All I/O is protected by token management, which ensures that the file system on multiple nodes honors the atomicity and provides data consistency of a file system.

The daemon is a multi-threaded process with some threads dedicated to specific functions. This ensures that services requiring priority attention are not blocked because other threads are busy with routine work. The daemon also communicates with instances of the daemon on other nodes to coordinate configuration changes, recovery and parallel updates of the same data structures. Specific functions that execute on the daemon include:

- Allocation of disk space to new files and newly extended files. This is done in coordination with the file system manager.
- Management of directories including creation of new directories, insertion and removal of entries into existing directories, and searching of directories that require I/O.
- Allocation of appropriate locks to protect the integrity of data and metadata. Locks affecting data that may be accessed from multiple nodes require interaction with the token management function.
- Disk I/O is initiated on threads of the daemon.
- Security and quotas are also managed by the daemon in conjunction with the file system manager.

3.5.5 GPFS RPD cluster

GPFS RPD cluster is a group of nodes with uniform disk access enabling concurrent data sharing. The GPFS cluster is created from an existing RSCT peer domain. There can only be one GPFS cluster per RSCT peer domain. Within that GPFS cluster, you may define multiple GPFS nodesets. However, a node may only belong to one nodeset.

You have two choices for the type of disk attachment you use to enable concurrent data sharing:

- Storage Attached Network (SAN) attachment of each disk to each node in the GPFS nodeset. The attachment can be via Serial Storage Architecture (SSA) or switched Fibre Channel. See Figure 3-19. Disks attached in this manner are formatted into logical volumes for use by GPFS via the `mmcrlv` command.

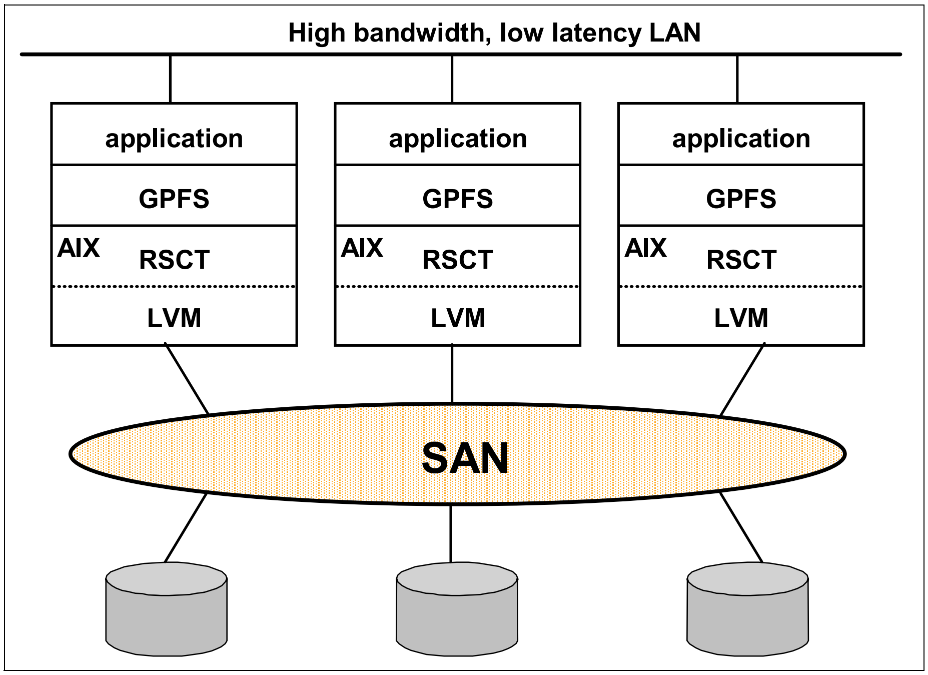


Figure 3-19 A GPFS cluster with SAN-attached disks

- Software simulation of a SAN by the use of the IBM Virtual Shared Disk component of RSCT. See Figure 3-20. Disks attached in this manner are formatted into Virtual Shared Disks for use by GPFS via the `mmcrvsd` command.

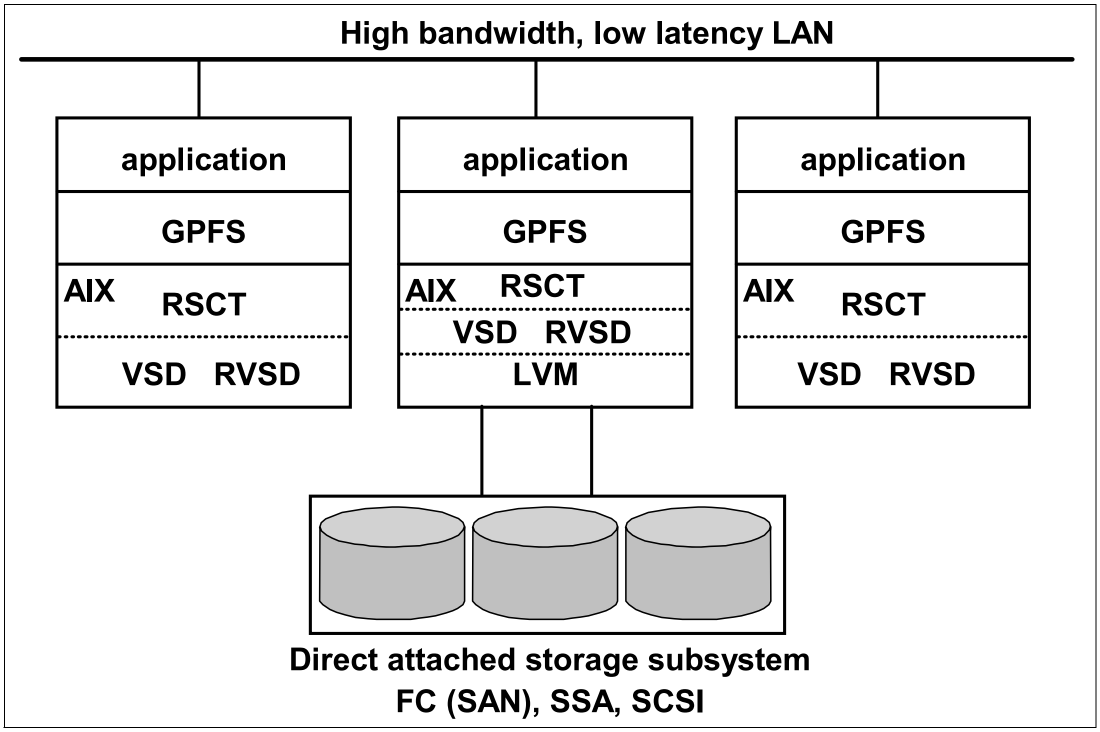


Figure 3-20 A GPFS cluster using the facilities of the VSD and the RVD components of RSCT

The type of disks, either Virtual Shared Disks or logical volumes, within a GPFS nodeset and its file systems, may not be mixed. The first disk type that the `mmcrfs` command encounters is designated as the disk type for the nodeset, the file system being created, and any subsequent file systems for that nodeset.

The size of your GPFS nodeset is constrained by the type of disk attachment.

- For disks SAN-attached to each node in the nodeset:
  - If any of the disks in the file system are SSA disks, your nodeset may consist of up to eight nodes (the size of the nodeset is constrained by the limitations of the SSA adapter).
  - If the disks in the file system are purely switched Fibre Channel, your nodeset may consist of up to 64 nodes (the size of the nodeset is constrained by the limitations of the Group Services software).
- For disks attached to an IBM Virtual Shared disk server using the IBM Virtual Shared Disk and IBM Recoverable Virtual Shared Disk components of RSCT, your nodeset may consist of up to 128 nodes (the size of the nodeset is constrained by the limitations of the Group Services software). The IBM Virtual Shared Disk component allows application programs executing on different nodes to access a logical volume as if it were local at each node.

The IBM Virtual Shared Disk subsystem supports two methods of external disk access:

- A non-concurrent mode in which only one node has access to a shared external disk at a given time. A primary and a backup server are defined.
- A concurrent mode in which multiple servers are defined to access the disk concurrently.

The IBM Recoverable Virtual Shared Disk component allows a secondary or backup server to be defined for a logical volume, providing the fencing capabilities required to preserve data integrity in the event of certain system failures. Therefore, the IBM Recoverable Virtual Shared Disk component is required even in the event that there are no twin-tailed disks.

When a GPFS nodeset is being configured, or nodes are being added to or deleted from the cluster, GPFS obtains the necessary additional configuration data from the resource classes maintained by the RSCT peer domain:

- Node number (PeerNode resource class)
- Adapter type (NetworkInterface resource class)
- IP address (NetworkInterface resource class)

For disks attached to an IBM Virtual Shared Disk server, when a GPFS file system is being configured, or disks are being added to or deleted from the file system, GPFS obtains the necessary configuration data from the resource classes maintained by the RSCT peer domain.

The complete configuration and file system data maintained by GPFS is then stored on the primary, and if specified, the secondary GPFS cluster data server as designated on the **mmcrcluster** or **mmchcluster** command.

The GPFS operating environment in an RPD cluster

GPFS is designed to operate with AIX 5L, which provides:

- The basic operating system services and the routing of file system calls requiring GPFS data.
- The LVM subsystem for direct disk management.
- Persistent reserve for transparent failover of disk access in the event of disk failure.
- The RSCT subsystem, which includes:
  - The Resource Monitoring and Control (RMC) component, establishing the basic cluster environment, monitoring the changes within the domain, and enabling resource sharing within the domain.
  - The Group Services component, coordinating and synchronizing the changes across nodes in the domain, thereby maintaining the consistency in the domain.
  - The Topology Services component, providing network adapter status, node connectivity, and a reliable messaging service.
  - The configuration manager, employing the above subsystems to create, change, and manage the RSCT peer domain.
  - The IBM Virtual Shared Disk component, providing disk driver-level support for GPFS cluster-wide disk accessibility.
  - The IBM Recoverable Virtual Shared Disk component, providing the capability to fence a node from accessing certain disks, which is a prerequisite for successful recovery of that node. It also provides for transparent failover of disk access in the event of the failure of a disk server.

3.5.6 GPFS HACMP cluster

The GPFS in the HACMP cluster has the same constraints as in the RPD cluster described in 3.5.5, “GPFS RPD cluster” on page 119.

After a GPFS nodeset has been configured, or nodes have been added to or deleted from the nodeset, GPFS obtains the necessary additional configuration data from the HACMP Global Object Data Manager (ODM):

- Node number
- Adapter type
- IP address

The complete configuration data maintained by GPFS is then stored on the primary, and if specified, the secondary GPFS cluster data server as designated on the **mmcrcluster** or **mmchcluster** command

3.5.7 Recoverability considerations

GPFS provides you with parameters that enable you to create a highly available file system with fast recoverability from failures. At the file system level, the metadata and data replication parameters are set.

At the disk level when preparing disks for use with your file system, you can specify disk usage and failure group positional parameters to be associated with each disk.

Additionally, GPFS provides several layers of protection against failures of various types, such as node or disk failure.

Node failure

This basic layer of protection covers the failure of file system nodes and is provided by Group Services.

When an inoperative node is detected by Group Services, GPFS fences it out. This prevents any write operations that might interfere with recovery.

If your disks are:

- Served by a Virtual Shared Disk server, GPFS fences it out using the facilities of the IBM Recoverable Virtual Shared Disk component.
- SAN-attached, GPFS fences it out using environment-specific subsystems.

File system recovery from node failure should not be noticeable to applications running on other nodes, except for delays in accessing objects being modified on the failing node.

Recovery involves rebuilding metadata structures, which may have been under modification at the time of the failure. If the failing node is the file system manager for the file system, the delay will be longer and proportional to the activity on the file system at the time of failure, but no administrative intervention will be needed. During node failure situations, if multi-node quorum is in effect, quorum needs to be maintained in order to recover the failing nodes.

If multi-node quorum is not maintained due to node failure, GPFS restarts on all nodes, handles recovery, and attempts to achieve quorum again.

Virtual Shared Disk server and disk failure

The two most common reasons why data becomes unavailable are disk failure and disk server failure with no redundancy.

In the event of a disk failure where GPFS can no longer read or write to the disk, GPFS will discontinue use of the disk and will await its return to an available state. You can guard against loss of data availability from a disk failure by utilizing the GPFS data replication feature or by utilizing hardware data replication such as RAID. Should you choose GPFS data replication, the data on each disk will be duplicated to the second mirror disk in a separate failure group.

In the event of a disk server failure where GPFS can no longer contact the node providing remote access to a disk, GPFS will again discontinue use of the disk.

You can guard against loss of disk server availability by utilizing common disk connectivity at multiple nodes and specifying a backup Virtual Shared Disk server for the common disk. Also, but using a RAID storage device, additional protection may be added, since certain RAID arrays mask the failure of a physical disk device.

An ideal configuration is shown in Figure 3-21, where a RAID device is twin-tailed to two nodes. This protects against server failure as well.

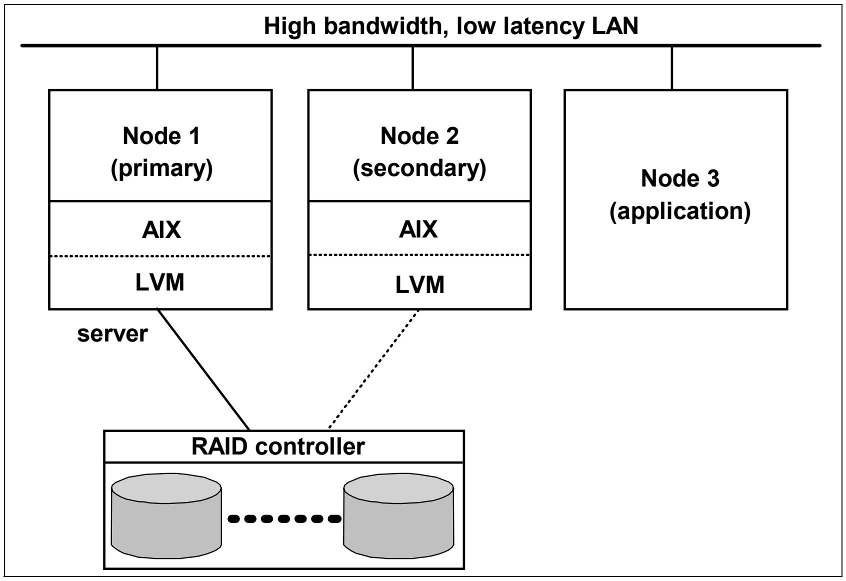


Figure 3-21 Primary node serving RAID device

If node 1, the primary server, fails, its responsibilities are assumed by node 2, the backup server, as shown in Figure 3-22 on page 124.

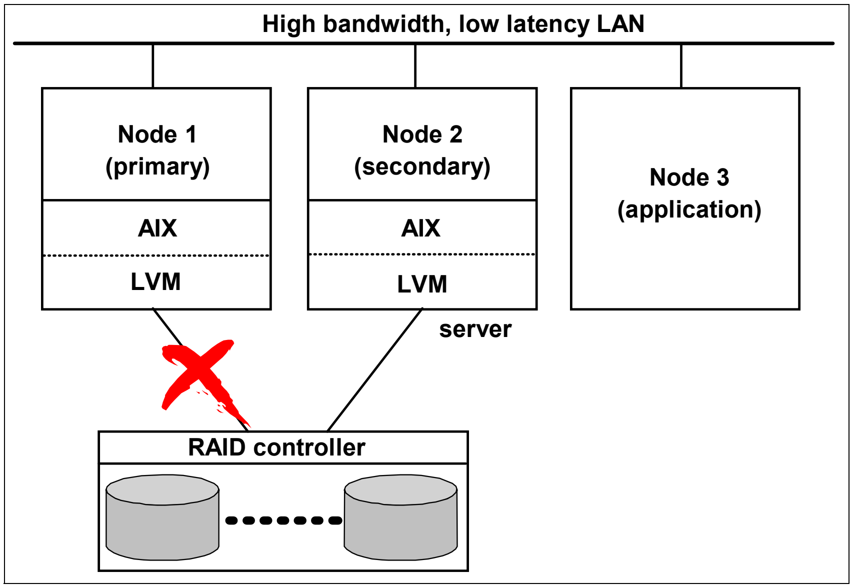


Figure 3-22 Backup node serving RAID device

Another means of data protection is through the use of concurrent Virtual Shared Disks, as shown in Figure 3-23. Concurrent disk access allows you to use multiple servers to satisfy disk requests by taking advantage of the concurrent disk access environment supplied by AIX.

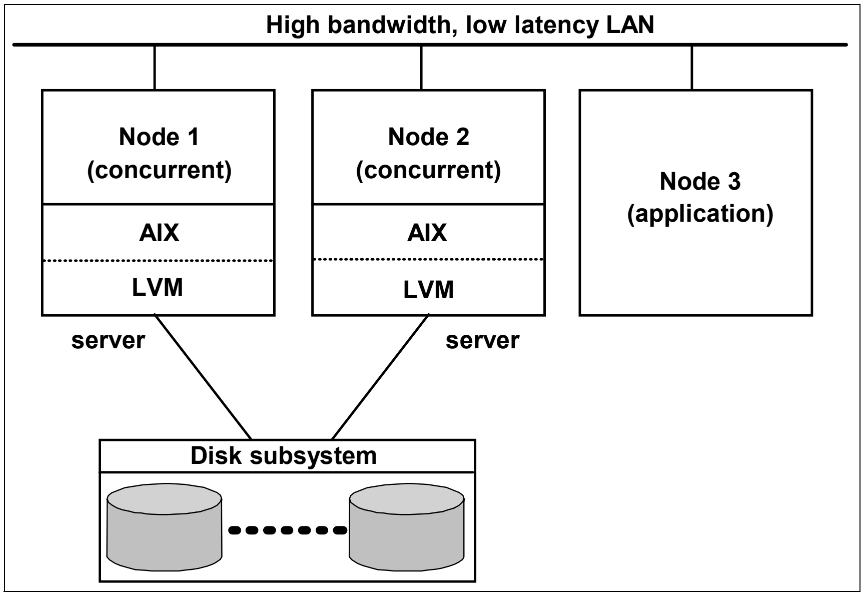


Figure 3-23 Concurrent node serving device

You can also protect your file system against disk failure by mirroring data at the logical volume manager (LVM) level, writing the data twice to two different disks. The addition of twin-tailed disks to such a configuration adds protection against server failure by allowing the IBM Recoverable Virtual Shared Disk program to route requests through a backup server.

SAN-attached disks

If your disks are SAN-attached to each node in the nodeset, one means of data protection is the use of a disk subsystem that supports RAID. RAID masks disk failures by providing hardware support consisting of some combination of parity volumes, mirrored volumes, and hot spares. An examples of a RAID controller is the IBM RAID/Enterprise Storage Subsystem (ESS) controller, which masks disk failures with parity disks. An ideal configuration is shown in Figure 3-24, where a RAID/ESS controller is multi-tailed to each node in the nodeset.

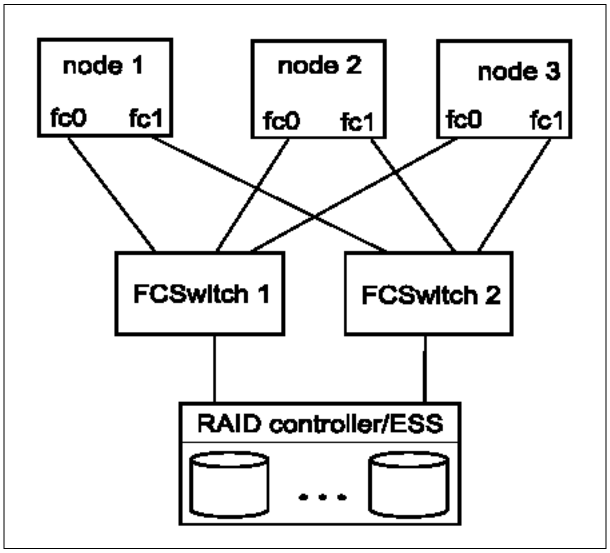


Figure 3-24 RAID/ESS Controller multi-tailed to each node

Node failure when using logical volumes

In order to preserve data integrity in the event of certain system failures, GPFS will fence a node that is down from the file system until it returns to the available state. Depending upon the types of disk you are using, there are three possible ways for the fencing to occur:

- SSA fencing SSA disks

> **Note:** Ensure your SSAR value is set to a different numeric value (excluding zero) by the `chdev` command.

- SCSI-3 persistent reserve
- Disk leasing

GPFS also provides a specific fencing mechanism for disk subsystems that do not support either SSA fencing or SCSI-3 persistent reserve mechanisms.

Single-node quorum is only supported when disk leasing is not in effect. Disk leasing is activated if any disk in any file system in the nodeset is not using SSA fencing or SCSI-3 persistent reserve.

3.5.8 GPFS quorum considerations

GPFS quorum defines the number of nodes in the nodeset that must be functional for GPFS to remain active. If the available nodes fall below the quorum level, GPFS performs recovery in an attempt to achieve quorum again.

Multi-node quorum

If multi-node quorum is used with GPFS, quorum needs to be maintained in order to recover the failing nodes. If multi-node quorum is not maintained due to node failure, all GPFS nodes restart, handle recovery, and attempt to achieve quorum again. Therefore, in a three-node system, failure of one node will allow recovery and continued operation on the two remaining nodes. This is the minimum configuration where continued operation is possible due to the failure of a node. That is, in a two-node system where single-node quorum has not been specified, the failure of one node means both nodes will restart, handle recovery, and attempt to achieve quorum again.

Two-node quorum and single-node quorum

If you configure a two-node cluster, you have the choice of which quorum algorithm to use:

1. Accept the default multi-node quorum (one plus half of the number of nodes in the nodeset). The use of the default multi-node quorum implies the remaining node in a two-node nodeset will no longer have access to GPFS file systems in the event of failure of the peer node.
2. Specifying the use of single-node quorum with either the **`mmconfig -U`** or the **`mmchconfig`** commands. The specification of single-node quorum allows the remaining node in a two-node nodeset to continue accessing GPFS file systems in the event of a failure of the peer node.

That is, in a two-node system where multi-node quorum is in effect, the failure of one node means both nodes will restart GPFS, handle recovery, and attempt to achieve quorum again. Quorum will not be achieved until the failing node is functioning again. If single-node quorum is in effect, the failure of one node results in GPFS fencing the failing node from the disks containing the GPFS file system data. The remaining node will continue processing if the fencing operation was successful. If not, those file systems that could not be completely fenced will be unmounted and attempts to fence the node will continue.

Restrictions when using single-node quorum

- Single-node quorum is not supported when disk leasing is in effect. Disk leasing is a GPFS-specific fencing mechanism for disks that do not support either SSA fencing or persistent reserve. Disk leasing is activated if any disk in any file system in the nodeset is not using SSA fencing or persistent reserve.
- If a disk is using persistent reserve, this needs to be supported at both the disk subsystem level and the AIX device driver level. For GPFS Version 2.1 running AIX 5L, only SSA disks and IBM ESS without the SDD driver support persistent reserve mode.
- Single-node quorum is not supported if SDD is installed.
- When adding nodes to a nodeset using the single-node quorum algorithm, the GPFS daemon must be stopped on all of the nodes. If after adding the nodes, the number of nodes in the nodeset exceeds two, the quorum algorithm is automatically changed to the multi-node quorum algorithm.
- You can use GPFS on top of the Virtual Shared Disks in order to use the single-node quorum with disk leasing, as shown in the scenario described in 5.3, “GPFS on VSD” on page 205.

When deleting nodes from a nodeset:

- A node cannot be deleted from the nodeset without stopping the GPFS daemon on both nodes.
- If the number of nodes remaining in the nodeset falls below three, you must run the **`mmchconfig`** command if you want to change the quorum algorithm to single-node quorum.

GPFS Filesystem Descriptor

There is a structure in GPFS called the Filesystem Descriptor (FSDesc) that is written originally to every disk in the filesystem, but is updated only on a subset of the disks as changes to the filesystem occur, such as adding or deleting disks. The subset of disks is usually a set of three or five disks, depending on how many disks and failure groups are in the filesystem. The disks that constitute this subset of disks can be found by reading any one of the FSDesc copies on any disk. The FSDesc may point to other disks where more up-to-date copies of the FSDesc are located.

To determine the correct filesystem configuration, a quorum of the subset of disks must be online so that the most up-to-date FSDesc can be found. If there are three special disks, then two of the three must be available.

GPFS distributes the copies of FSDesc across the failure groups. If there are only two failure groups, one failure group has two copies and the other failure group has one copy.

In a scenario that causes one entire failure group to disappear all at once, if half of the disks that are unavailable contain the single FSDesc that is part of the quorum, everything stays up. On the other hand, if the downed failure group contains the majority of the quorum, the FSDesc cannot be updated and the filesystem must be force unmounted.

If the disks fail one at a time, the FSDesc is moved to a new subset of disks by updating the other two copies and a new disk copy. However, if two of the three disks fail simultaneously, the FSDesc copies cannot be updated to show the new quorum configuration. In this case, the filesystem must be unmounted to preserve existing data integrity.

> **Tip:** To survive a single ESS failure in a dual ESS configuration, you must have a third failure group on an independent disk outside both ESSs. You can designate this disk to just have filesystem metadata so it does not need as much capacity to balance the other failure groups. This independent disk provides the tiebreaker FSDesc.

3.6 Engineering applications

IBM has long been a leader in the scientific community. These traditions continue with support for the IBM @server IBM @server pSeries High Performance Switch (HPS). In this section we cover some new aspects about engineering applications and their interaction with the IBM @server pSeries HPS. The engineering applications covered in this section are:

- Low-level application programming interface (LAPI)
- LoadLeveler
- Parallel environment
- Message passing interface
- Parallel Engineering and Scientific Subroutine Library

3.6.1 Low-level application programming interface (LAPI)

The low-level application programming interface (LAPI) is a message-passing API that provides a one-sided communication model. In this model, one task initiates a communication operation to a second task. The completion of the communication does not require the second task to take a complementary action. RSCT LAPI provides optimal communication performance over an IBM @server pSeries High Performance Switch (pSeries HPS).

The LAPI library provides basic operations to put data to and get data from one or more virtual addresses of a remote task. LAPI also provides an active message infrastructure. With active messaging, programmers can install a set of handlers that are called and run in the address space of a target task on behalf of the task originating the active message.

Some of LAPI's other general characteristics include:

- Flow control
- Support for large messages
- Support for generic non-contiguous messages
- Non-blocking calls
- Interrupt and polling modes
- Efficient exploitation of switch functions
- Even monitoring support (to simulate blocking calls, for example) for various types of completion events

LAPI has been enhanced to exploit the shared memory of the IBM @server High Performance Switch. As such, it provides access to the improved performance of the new switching architecture.

RSCT LAPI provides a lower-level interface to an IBM @server HPS (as does PSSP LAPI to an SP Switch or an SP Switch 2) than either the message-passing interface (MPI) or the Internet Protocol (IP), so you can choose how much additional communication protocol needs to be added.

For RSCT LAPI users, an efficient bulk transfer mechanism for large data transfers. This mechanism can only be used in conjunction with the pSeries HPS. For further information refer to the *IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 in the section on bulk transfer of messages on page 58.

Table 3-7 shows the new features for LAPI, specific features for IBM @server HPS, and the sources for further information about each topic.

Table 3-7 New features for LAPI

Change or addition	For more information see
For RSCT LAPI users, support for the IBM @server pSeries High Performance Switch (pSeries HPS). To take advantage of the pSeries HPS, you need to have Version 1.3.2 of Cluster Systems Management (CSM) for AIX 5L (product number 5765-F67) installed on your system.	▸ *pSeries High Performance Switch: Planning, Installation, and Service*, GA22-7951-00 ▸ *CSM for AIX 5L: Software Planning and Installation Guide*, SA22-7919-04
The ability to run over the user datagram protocol (UDP) rather than the user space (US) protocol.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - "LAPI communication modes" on page 22
Inline completion handlers, which allow your completion handler to be called from within the thread of execution that completed the data transfer. For applications that rely on completion handlers rather than counters, this can provide a significant performance enhancement.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide* - "Inline completion handlers" on page 57
A generalized mechanism to transfer arbitrary portions of non-contiguous data using data gather/scatter programs (DGSPs).	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - "Using data gather/scatter programs (DGSPs)" on page 35

Change or addition	For more information see
New runtime attributes that let LAPI clients control the communication library and get information about new statistics (PRINT_STATISTICS and QUERY_STATISTICS)	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “LAPI Qenv” on page 134 and “Attributes that return multiple values” on page 215
A new polling foundation that provides more flexible access to invoking the LAPI dispatcher explicitly and lets you poll for messages without polling for specific counters.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “LAPI_Msgpoll” on page 121
Support for multi-threaded programs. LAPI calls can now be made from multiple user threads.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “Thread programming in LAPI” on page 65
For RSCT LAPI users, an efficient bulk transfer mechanism for large data transfers. This mechanism can only be used in conjunction with the IBM @server HPS.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “Bulk transfer of messages” on page 58
A new API call that performs various LAPI utility operations, most notably for user DGSPs.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “LAPI_Util” on page 153
A new LAPI_Xfer transfer type to support DGSP transfers: LAPI_DGSP_XFER (DGSP).	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “lapi_amdgsp_t details” on page 167
Several new data structures.	*IBM Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00 - “New data structures” on page 16

3.6.2 LoadLeveler

LoadLeveler (LL or LoadL) is a job management system that utilizes LAPI and is used by IBM Parallel Environment. LL allows users to run more jobs in less time by matching the jobs’ processing needs with the available resources. LL schedules jobs and provides functions for building, submitting, and processing jobs quickly and efficiently in a dynamic environment.

LoadL interfaces

LoadLeveler has three types of interfaces that enable users to create and submit jobs and allow system administrators to configure the system and control running jobs. These interfaces include the command-line interface, a GUI interface, and an API. All three types of interfaces permit different levels of access to users and administrators.

Command-line interface

The command-line interface gives you access to basic job and administrative functions.

For more information, see *IBM LoadLeveler for AIX 5L Using and Administering,* SA22-7881-01 - Chapter 2, “LoadLeveler command line interface,” on page 192.

A graphical user interface (GUI)

LoadLeveler’s GUI provides system access similar to the command-line interface. Experienced users and administrators may find the command-line interface more efficient than the GUI for job and administrative functions.

For more information, see *IBM LoadLeveler for AIX 5L Using and Administering*, SA22-7881-01 - Chapter 3, “Using the Graphical User Interface,” on page 213.

Application programming interface (API)

The Application Programming Interface allows application programs written by users and administrators to interact with the LoadLeveler environment.

For more information, see *IBM LoadLeveler for AIX 5L Using and Administering*, SA22-7881-01 - Chapter 4, “LoadLeveler API interface,” on page 35.

HPS enablement

LoadLeveler Version 3.2 with APAR IY49652 offers several new features. Primarily, LoadLeveler will make use of the IBM @server pSeries High Performance Switch (HPS). LL includes many new changes to decouple LoadLeveler from PSSP while maintaining backward compatibility.

Support for HPS

LoadLeveler will continue to use the Job Switch Resource Table Services (JSRT) to provide an interface to the SP Switch and the SP Switch adapters. The Network Table Services (NTBL) will provide a new interface for the HPS and HPS adapters. NTBL will not support a function similar to the swtbl_adapter_connnectivity() API function in the JSRT.

If an administrator wants to configure LoadLeveler to communicate using HPS adapters, it is strongly recommended that the ml0 interface be used for a PEER domain containing HPS adapters. The ml0 interface is guaranteed to be present on every operating system image containing an HPS adapter, whereas sn0, sn1,..., snn may not be.

Exploiting RSCT for Dynamic Adapter configuration

The absence of the adapter_stanzas keyword in the LoadLeveler Admin file for an OSI indicates that adapters for the OSI should be configured dynamically. Startd will determine what adapters are present at startup via calls to the RMC API. Startd subscribes to the local RMC for adapter changes. Changes are communicated to the Central Manager as they occur.

Note: Pre-HPS switch adapter characteristics cannot be determined dynamically (css0, css1).

LoadLeveler uses the RMC API to get the adapter connectivity information. LoadLeveler uses the RMC API to extract data from the RSCT Peer Domain to support dynamic adapter configuration. The new LoadLeveler command, **llextRPD**, may be run to extract data from the RSCT Peer Domain needed by LoadLeveler to set up the administration file. The LoadLeveler GSmonitor will optionally use the RMC API when group services invokes the callback.

LoadLeveler has a new task that gets machines and adapters from the peer domain that runs **llextRPD**.

The **llextSDR** command remains unchanged for PSSP domains. The name of the task, *Get Machines and Adapters* from RS/6000 SP, will be changed to *Get Machines and Adapters* from the PSSP Domain.

Job Builder may be invoked prior to the adapter information being known, due to dynamic adapter configuration. The list containing adapter_names and network_types on the Network panel is changed to an editable ComboBox. This allows the user to enter a value that does not appear in the list.

Removal of PSSP dependencies

The LoadLeveler GSmonitor daemon is changed to optionally use the RMC API in the AIX Cluster environment that does not depend on the PSSP's SDR.

The new **llextRPD** command provides the same functionality as the **llextSDR** command without depending on the PSSP's SDR.

LoadLeveler error log entries are generated by directly calling AIX error logging functions, eliminating the dependency on the PSSP ppslog facility. Log entries are more descriptive, and include probable causes and user actions. The number of errlog categories is reduced, and unnecessary errlog entries are eliminated.

Changes and coexistence

The LoadLeveler administrator may set up a LoadLeveler Admin file that contains OSIs that span both PSSP and peer domains. If GSmonitor is to be used for monitoring failing OSIs, `GSMONITOR_RUNS_HERE = TRUE` should be specified in the config file on a node for each domain. A GSmonitor executing in a PSSP or peer domain can monitor only the OSIs contained within that domain.

To identify which set of peers to monitor, also add `GSMONITOR_DOMAIN = [ PSSP | PEER ]` to the config file. This restricts the execution of the GSmonitor daemon to the PSSP or peer (Cluster) domain. The default is `PEER` if GSmonitor is running in an active RSCT peer domain. Otherwise, it will attempt to use the SDR access routines contained in libSDR.a(shr.o).

> **Note:** If the GSMONITOR_DOMAIN is specified as PSSP or PEER, the LoadLeveler GSmonitor daemon will start only if it is in a valid domain corresponding to the specification.

The keywords listed in Table 3-8 vary between SDR and RPD clusters.

Table 3-8 Loadl.admin file changes

Admin file keyword	**Source in SDR (llextSDR)**	**Source in Cluster (llextRPD)**
spacct_excluse_enable	‘spacct_excluse_enable’	Not supported
dce host name	‘dce_host_name’	Not supported
css_type	‘css_type’	Not supported
adapter type	‘Not supported	‘IBM.NetworkInterface->AdapterType’
switch_node_number	‘switch_node_number’	Not supported
logical_id	Not supported	‘IBM.NetworkInterface->AdapterLID’
device driver name	Not supported	‘IBM.NetworkInterface->DeviceName’
network_id	Not supported	‘IBM.NetworkInterface->AdapterNetworkID’

References

For further information, refer to *IBM LoadLeveler for AIX 5L Using and Administering*, SA22-7881-01.

3.6.3 Message passing interface

Message passing interface (MPI) is a paradigm used widely on certain classes of parallel machines, especially those with distributed memory. Although there are many variations, the basic concept of processes communicating through messages is well understood. Over the last ten years, substantial progress has been made in casting significant applications in this paradigm. Each vendor has implemented its own variant.

MPI has been enhanced to support operations on the HPS. Utilization of the SNI link relies on the new low-level API (LAPI) provided by rsct.lapi. MPI is scalable to 512 OS instances. Each p690 OS Instance supports up to eight SNI links and each p655 OS instance supports up to two SNI links.

MPI communications support 16000 independent channels per SNI link to support both dedicated and shared channel connections. These channels are user space accessible through virtual memory addressing.

MPI has been enhanced to use LAPI as the common transport protocol rather than shared memory. This provides switch window sharing between MPI and LAPI, as well as improved MPI memory scalability.

MPI on HPS supports processor offload assistance among up to 16 partitions in a push/pull model. Traffic may use both reliable and unreliable delivery modes. Usage of multiple routes is supported. The traditional SP protocols MPI, TCP/IP, UDP/IP, LAPI, and KLAPI are supported, as well as HPS-specific bulk transfer versions of MPI, IP, and LAPI.

Figure 3-25 shows the Message Passing Software Architecture.

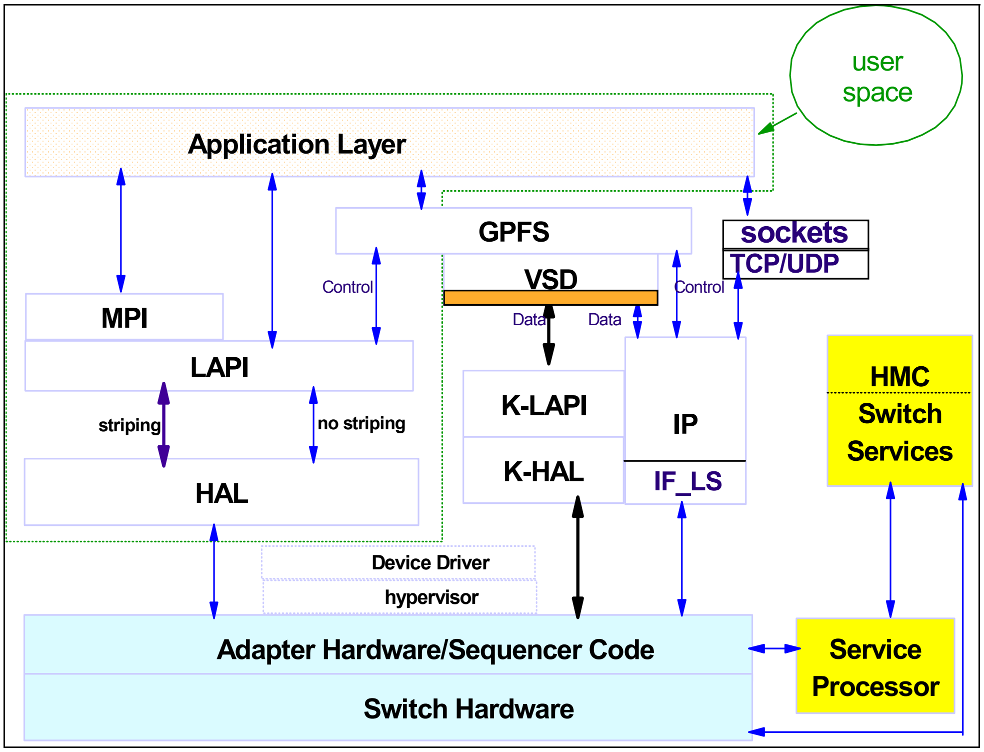


Figure 3-25 Message Passing Software Architecture

3.6.4 Parallel Environment

The IBM Parallel Environment for AIX (PE) software lets you develop, debug, analyze, tune, and execute parallel applications written in Fortran, C, and C++. PE conforms to existing standards such as UNIX and MPI. The PE runs on either an IBM RS/6000 SP (SP) machine or a clustered server.

3.6.5 Parallel Engineering and Scientific Subroutine Library (ESSL)

Parallel Engineering and Scientific Subroutine Library (ESSL) is a scalable mathematical subroutine library that supports parallel processing applications on IBM RS/6000 SP™ Systems (AIX only) and on clusters of IBM @server pSeries and IBM RS/6000 workstations. Parallel ESSL supports the Single Program Multiple Data (SPMD) programming model using the Message Passing Interface (MPI) library. Parallel ESSL provides subroutines in six major areas of mathematical computations.

Available subroutines

Parallel ESSL provides subroutines in the following computational areas:

- Level 2 Parallel Basic Linear Algebra Subprograms (PBLAS)
- Level 3 PBLAS
- Linear Algebraic Equations
- Eigensystem Analysis and Singular Value Analysis
- Fourier Transforms
- Random Number Generation

New features

PE Version 4.1 adds some new features:

- Several collective communications routines for 64–bit programs have been enhanced to use shared memory for better performance.
- PE now supports the new pSeries High Performance Switch (HPS) as well as RS/6000 SP Switches by use of MPI over LAPI.
- PE now can use a co-scheduler for large scale jobs, with fine-grained parallelism.
- Parallel jobs may be check-pointed from within an attached debugger.
- Threaded library support only, with binary compatibility for signal (non-threaded) library applications.
- Electronic Licensing.
- Cluster-based security support.
- Xprofiler now part of AIX.
- Parallel utility subroutine MP_QUERYINTRDELAY, mpc_queryintrdelay is no longer supported. When invoked, it returns a value of zero.
- Parallel utility subroutine MP_SETINTRDELAY, mpc_setintrdelay is no longer supported. When invoked, it returns a value of zero.
- New library to be used when converting MPI trace files to the slog2 file format that is used by the latest version of Jumpshot, an MPI trace viewer program available from Argonne National Laboratory.
- User-written SIGIO handlers are no longer invoked when a packet arrives.

3.6.6 Further information

For further information about PE, MPI and ESSL, refer to the following documentation:

- *IBM Parallel Environment for AIX 5L Hitchhiker's Guide*, SA22-7947-00
- *Parallel Engineering and Scientific Subroutine Library for AIX*, Version 3 Release 1
- *Linux on pSeries, Version 3 Release 1 Guide and Reference,* SA22-7906-01

3.7 Databases

Oracle 9i Real Application Clusters (RAC) is the parallel version of the Oracle 9i database software. It makes it possible to run multiple instances (each on a separate cluster node) for accessing the same database.

Since the different database instances have to access the same database files, a shared storage space is required (which can be accessed from all cluster nodes, concurrently). This can be provided by hardware (for example, via fiber-attached storage) or by an additional software layer that provides shared storage access via the network to other nodes in the same cluster (for example, Virtual Shared Disks).

Database instances running on different cluster nodes require a fast (high bandwidth, low latency) and reliable (highly available) interconnect network for transferring locking information and exchanging database blocks. For example, Oracle 9i RAC needs a fast interconnect network, which should be a high-speed, low-latency, switched network. HPS provides a high-speed, low-latency, switched network. RAC is highly benefited from this network in terms of performance.

With Oracle 9i RAC, GPFS can be used to store the database files (instead of RAW devices), the database executables, and the configuration files (one common repository for all nodes instead of separate copies for each node).

Both VSD and GPFS use the high-performance, low-latency, switched network provided by the IBM @server HPS.

For more information about Oracle9i RAC, see:

- *Oracle9i Real Application Clusters: Concepts, Release 2 (9.2), March 2002*, A96524-01
- *Oracle9i Real Application Clusters: Deployment and Performance, Release 2 (9.2)*, March 2002, A96598-01

And the redbook:

- *Deploying Oracle9i RAC on IBM IBM @server Cluster 1600 with GPFS*, SG24-6954

3.8 Documentation

Manuals:

- *Reliable Scalable Cluster Technology for AIX 5l Administration Guide*, SA22-7889-02
- *Reliable Scalable Cluster Technology for AIX 5L Guide and Reference*, SA22-7889-01
- *Reliable Scalable Cluster Technology for AIX 5L Technical Reference*, SA22-7890-03
- *Reliable Scalable Cluster Technology for AIX 5L Group Services Programming Guide and Reference*, SA22-7888-03
- *Reliable Scalable Cluster Technology for AIX 5L LAPI Programming Guide*, SA22-7936-00

- *Reliable Scalable Cluster Technology for AIX 5L Managing Shared Disks*, SA22-7937-00
- *Cluster Systems Management for AIX 5L Administration Guide*, SA22-7918-04
- *Cluster Systems Management for AIX 5L Software Planning and Installation Guide*, SA22-7919-04
- *Cluster Systems Management for AIX 5L Hardware Control Guide*, SA22-7920-04
- *Cluster Systems Management for AIX 5L Command and Technical Reference*, SA22-7934-02
- *IBM @server pSeries High Performance Switch: Planning, Installation, and Service*, GA22-7951-00
- *IBM LoadLeveler for AIX 5L Using and Administering*, SA22-7881-01
- *IBM Parallel Environment for AIX 5L Hitchhiker's Guide*, SA22-7947-00
- *Parallel Engineering and Scientific Subroutine Library for AIX, Version 3 Release 1, and Linux on pSeries, Version 3 Release 1 Guide and Reference*, SA22-7906-01
- *Oracle9i Real Application Clusters: Concepts, Release 2 (9.2)*, March 2002, A96524-01
- *Oracle9i Real Application Clusters: Deployment and Performance, Release 2 (9.2)*, March 2002, A96598-01
- *AIX 5L Version 5.2 Commands Reference*, SC23-4115-06
- *General Parallel File System for AIX 5L in an RSCT Peer Domain: Concepts, Planning, and Installation Guide Version 2.2*, GA22-7974-00
- *General Parallel File System for AIX 5L in an RSCT peer domain: Administration and Programming Reference*, SA22-7973-00
- *General Parallel File System for AIX 5L in an HACMP Cluster: Concepts, Planning, and Installation Guide Version 2.2*, GA22-7971
- *General Parallel File System for AIX 5L in an HACMP Cluster: Administration and Programming Reference*, SA22-7970

Redbooks:

- *Deploying Oracle 9 RAC on IBM @server Cluster 1600 with GPFS*, SG24-6954
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615
- *Linux HPC Cluster Installation*, SG24-6041
- *An Introduction to CSM 1.3 for AIX 5L*, SG24-6859

Web sites:

- `http://www-1.ibm.com/servers/eserver/clusters/library/`
- `http://www.mpi-forum.org/docs/`
- `http://www.ibm.com/servers/eserver/pseries/library/gpfs.html`

Part 2

Practical implementation examples

In Part 2, we discuss how to install and configure the IBM @server pSeries High Performance Switch (HPS), and provide some useful applications that may benefit from the switch or those that may simplify switch administration.

4

Installation, configuration and administration of pSeries HPS

This chapter provides a basic guide for installation and configuration of the IBM @server pSeries High Performance Switch, and also covers some administration topics. It contains the following sections:

- Planning for the installation of the High Performance Switch
- Installation and configuration of the High Performance Switch
- Administration of the High Performance Switch

4.1 Planning for the High Performance Switch

Planning the HPS installation is an important step in defining your working environment and in determining the actual needs for installing and configuring the pSeries HPS. This section contains two parts:

- Hardware planning: The hardware prerequisites needed for installing the HPS
- Software planning: The software requirements of the HPS

4.1.1 Hardware planning

Hardware planning should consider the following aspects:

- Supported eServer pSeries systems
- Adapters for the local area network
- HMC requirements
- IBM pSeries High Performance Switch

Supported eServer pSeries systems

Due to complex hardware architecture and the high bandwidth provided by the pSeries HPS, there are special hardware requirements for pSeries servers.

The pSeries models that are currently supported in a pSeries HPS environment are:

- eServer pSeries 690 (7040-681), running with CPU type:
  - POWER4 (1.1/1.3 GHz), also known as Regatta H series
  - POWER4+ (1.5/1.7 GHz), also known as Regatta H+ series
- eServer pSeries 655 (7039-651), running with CPU type:
  - POWER4 (1.1/1.3 GHz), also known as Regatta IH series
  - POWER4+ (1.5/1.7 GHz), also known as Regatta IH+ series

The systems connect to the High Performance Switch using a Switch Network Interface card attached to the GX bus slot. Depending on the pSeries system, the following Switch Network Interfaces are supported:

- pSeries 690:
  - IBM 4-Link Switch Network Interface for HPS (FC 6434)
  - IBM 2-Link Switch Network Interface for HPS (FC 6432)
- pSeries 655:
  - 2-Link GX bus mounted card (FC 6420)

The number of switch link connections in a pSeries server depends on the type of server and its hardware configuration. Thus the pSeries 690 can have up to eight links attached to switch, while the pSeries 650 can have only two links attached (see Table 4-1).

Table 4-1 Maximum links per server for connecting to a HPS network

Server	No. of MCMs	Max no. of links	SNI configuration
p690	1	2	2-Link card (FC 6432) on GX Slot 1
p690	2	4	2-Link card (FC 6432) on GX Slot 1 2-Link card (FC 6432) on GX Slot 3
p690	3	6	4-Link card (FC 6434) on GX Slot 1 2-Link card (FC 6432) on GX Slot 3

Server	No. of MCMs	Max no. of links	SNI configuration
p690	4	8	4-Link card (FC 6432) on GX Slot 1 4-Link card (FC 6434) on GX Slot 3
p655	1	2	2-Link card (FC 6420) on GX Slot 1

In a pSeries 690, the number of supported links depends on the MCM configuration, and it may have two, four, six, or eight links attached. See Figure 2-18 on page 31 for a correlation between the MCMs installed and the GX buses available in a pSeries 690 server.

In order to determine the GX slot location in your system for the installation of the switch interface cards, refer to *pSeries High Performance Switch Planning, Installation and Service*, GA22-7951.

Adapters for the local area network

You should plan for at least one network interface card in each LPAR for the LAN connection. For a Cluster 1600, a LAN must connect the following resources:

- The Hardware Management Console
- The partitions defined in the CEC complex
- The CSM server

The LAN adapters are used by the software functions included in the HMC code. They are also be used by the CSM server for management functions of LPARs and by the Network Install Manager when performing network operations against the LPARs.

Trusted network option for multiple HMCs

In a complex configuration, with multiple frames and SNM daemons running on multiple HMCs, a separate network for SNM communication should be provided. This is necessary because communication between SNM daemons running on separate HMCs should not be affected by any other network traffic (for performance and security reasons). Thus, we recommend using at least two network interfaces in each HMC, allocating one network interface for CSM to HMC communication and one network interface for HMC to LPAR traffic.

Tip: This is not a requirement. In a small environment with a couple of HMCs and a few LPARs, both CSM-to-HMC and HMC-to-LPAR traffic may use the same physical and logical network (IP subnet).

HMC requirements

Minimally, the HMC should be a 7315-C01 or equivalent. In supporting this class of machine, the initial HPS offering is limited to 32 switch nodes except by special bid due to HMC performance considerations. This should be lifted by the next major release and may require the next HMC GA level.

For the installation of the pSeries HPS a new serial cabling scheme is needed. There are two types of serial connections from HMC to the rack complex:

- RS232 between the HMC and CEC. This connection is used for CSP (Common Service Processor) hardware management functions. In the case of the switch management functions, this type of interface is used by SNM component for communication with the SNI adapters in the CEC.
- RS422 between HMC and the BPAs in the frame. The RS422 connection is needed for direct control functions over the switch, such as power on/off, status of components inside switch, and alerts.

When planning the connections of the HMC in a pSeries HPS environment, consider the following:

- The necessary number of serial interfaces. The number should be determined by the following rules:
  - Two serial ports are used for the connection between the HMC and each frame containing a switch (RS422).
  - One serial port for communication between the HMC and each CEC (RS232).

  Since the HMC contains only two serial ports, additional serial ports are required. The following adapters are currently provided by IBM for this function:

  - Eight-port asynchronous adapter PCI BUS EIA232/RS422 (FC 2943)
  - 128-port asynchronous controller PCI bus (FC 2944)

  Notes:

  - While the eight-port asynchronous adapter natively supports both RS232 and RS422 modes, the 128-port asynchronous adapter only supports RS232 through 16-port RANs. By the time the HPS is Generally Available (GA), an RS-232 to RS422 convertor should be available as part of the RS-422 cable set as ordered.
  - Up to two eight-port or 128-port asynchronous adapters are allowed per HMC. Mixing eight-port and 128-port adapters in an HMC is not supported.

- LAN adapters for connection with LPARs and the management server.

  Usually a LAN adapter is enough for the traffic between the HMC, the CSM server and the LPARs. In a more complex pSeries HPS environment containing multiple HMCs connected together, a trusted network is required. In this case, a second adapter on HMCs is used only for the traffic between the HMCs, creating the trusted network.

  The currently available IBM Ethernet adapters for HMC are:

  - 10/100 Mbps Ethernet PCI Adapter II (FC 4962)
  - Gigabit Ethernet SX PCI Adapter (FC 2969)
  - 10/100/1000 Base-T Ethernet PCI Adapter (FC 2975)

A redundant HMC can be installed in the pSeries HPS environment as an option. In this case you have to reconsider the hardware resources involved in your configuration. A cabling scheme is shown in Figure 4-1 on page 143.

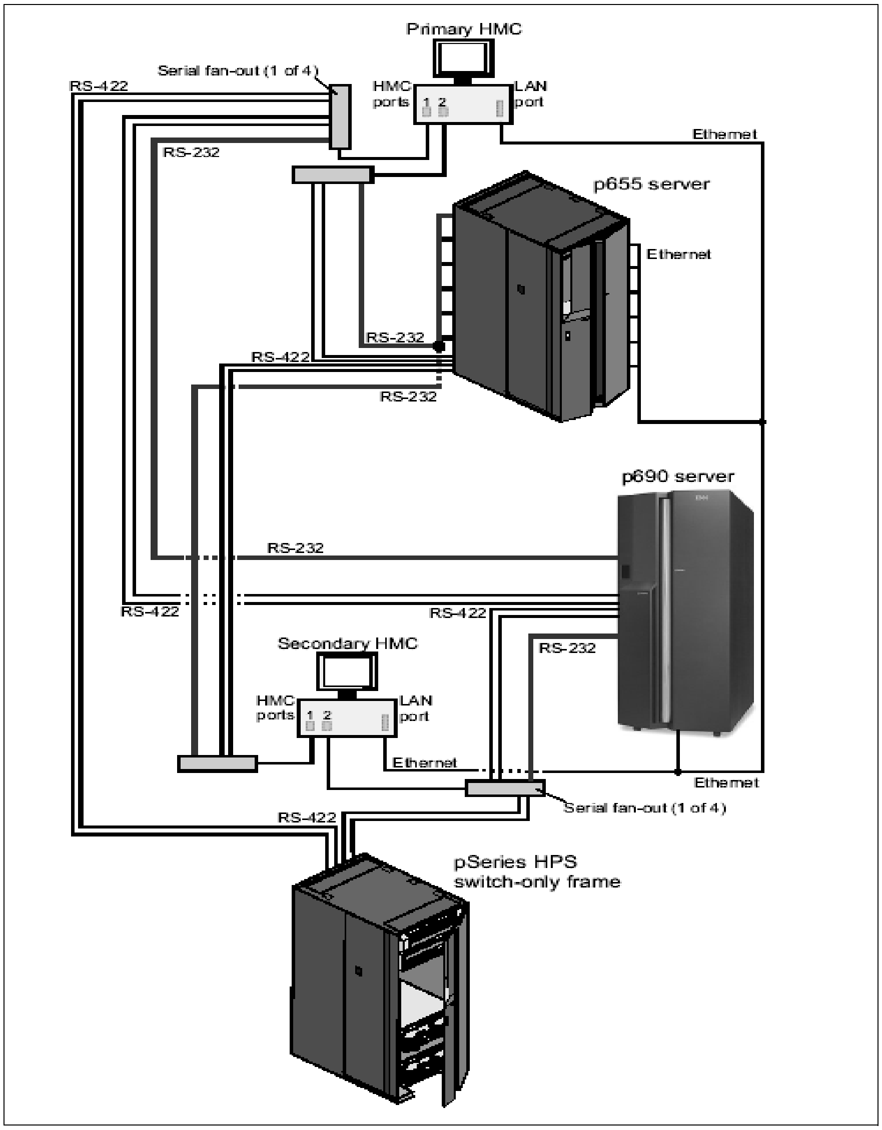


Figure 4-1 A redundant HMC configuration for pSeries HPS

IBM @server pSeries High Performance Switch (HPS)

pSeries HPS Rack

The pSeries HPS is a four-EIA-unit subsystem, so it occupies the same amount of space in the rack as an I/O drawer.

The pSeries HPS can be mounted in a rack according to the following rules:

- One switch in p690 rack (7040-61D)
- One switch in p655 rack (7040-W42)
- Additional switches must be mounted in a dedicated frame (7040-W42).

Important: Any frame containing an HPS must have the EMC skirts installed. In order to accommodate the cable installation in a cluster environment containing multiple frames connected to the HPS network, you must install the frames over a raised floor. Refer to *pSeries High Performance Switch Planning, Installation and Service*, GA22-7951 for further details about the parts needed and the hardware installation procedure.

The pSeries HPS switch connections

There are two types of switch connections:

- Server-to-switch connections. This type of connection involves:
  - SNI Interfaces on the server side: 2-link cards or 4-link cards
  - Copper switch port connection cards (FC6433) on the switch side
  - Copper connections between the SNI interfaces and the switch cards. The cables for server-to-switch connection are:
    - Copper switch cable 1.2 m (FC3161)
    - Copper switch cable 3 m (FC3166)
    - Copper switch cable 10 m (FC3167)

  The connections of an SNI link pair can be spread over the available switch cards, in the same switch assembly, thus providing protection in case one switch raiser card fails.

Note: The link pair provided by a 2-link SNI may be converted to single-link use by installing a special warp plug (FC6437). This feature is not supported for a 4-link SNI card. Should this configuration become available, it will allow two LPARs to be connected to the same switch riser card. When using this configuration, 2-link nodes may not span riser cards and all cabling must be sequential.

- Switch-to-switch connections:
  - A pair of switch raiser cards.

    For switch-to-switch connection there are two options:
    - Copper switch raiser cards (FC6433)
    - Fiber switch raiser cards (FC6436)
  - Communication cables between switch cards:
    - Copper cables: 1.2, 3, or 10 m
    - Fiber cables: 20 m (FC3256) or 40 m (FC3257) cables

Notes:

- The same copper switch cards and cables are used for server-to-switch and switch-to-switch connections, but there are dedicated slots in the switch for each connection type. The fiber cards contains two links = two FC loops (IN, OUT pairs) for a total of four ports on it.
- A switch-to-switch connection must have both links of the switch raiser cards connected.
- For performance reasons, when planning for switch-to-switch connections, take into consideration the number of node links from each switch to the directly attached nodes. For further details, consult *pSeries High Performance Switch Planning, Installation and Service*, GA22-7951.

Firmware

Mainly, for the pSeries HPS environment, you have to check for two types of microcode:

- GFW (system firmware) is the microcode related to the CEC part of the system. The firmware image contains:
  - System power control network programming (SPCN)
  - Service processor programming
  - IPL programming

The minimum level of system firmware required for connecting a system to the pSeries HPS is:

  – eServer pSeries 690: 3H031024
  – eServer pSeries 655: 3J031024

- Frame firmware is the microcode for the frame power subsystem. The minimum level required when installing a pSeries HPS is 259f.

Ensure that the latest firmware and microcode levels are installed on the system. The following firmware and microcode should be brought to the latest level:

- System microcode
- Frame firmware
- Integrated SCSI controller microcode
- Integrated Ethernet microcode

Since the SCSI and Ethernet controllers for the p690 system are actually placed on the I/O drawer, also check the firmware level associated with the I/O drawer 7040-61D.

For the latest versions of firmware for the systems and adapters, check the IBM technical support Web page:

```
https://techsupport.services.ibm.com/server/mdownload2/download.html
```

4.1.2 Software planning

Required software

- AIX 5.2 with PTF level 5200-02 and the SNI device drivers.
- Switch Network Management Software (located on HMC): Provides the pSeries HPS management functions
- Service Focal Point (SFP): Problem management software
- Service Agent: Service Gateway function between IBM and the customer
- Inventory Scout Services: Vital Product Data and Microcode Management software

Additional AIX 5.2 filesets

Reliable Scalable Cluster Technology (RSCT) provides information to other applications on available system resources and switch network topology. Components include:

- High Availability Topology Services (HATS)
- High Availability Group Services (HAGS)
- Configuration resource manager (ConfigRM)
- IBM Virtual Shared Disk (VSD)
- Low-level communications application (LAPI)
- Network Table (NTBL) API

> **Note:** Most of the RSCT components are optional. However, the rsct.core fileset must exist on LPARs for different management functions of the HMC. It is selected by default when installing the AIX 5.2 operating system.

For a summary of software versions required for installing the pSeries HPS, refer to Table 4-2 on page 146.

Table 4-2 Software components required for HPS operation

Product	Comments
AIX 5.2	Minimum PTF level 5200-02, recommended APAR at HPS GA time IY49612
SNI code	Recommended PTF level is 1 (APAR IY49654)
RSCT 2.3.1	Recommended PTF level is 1 (APAR IY49653)
HMC 3.2.5	APAR IY50216, PTF# U496326

Recommended software packages

- Cluster Systems Management for AIX 5L (CSM): CSM provides the management cluster capability for AIX and Linux systems. CSM software is packaged with AIX 5.2.

  When configuring your switch environment as part of a Cluster 1600 managed by CSM, the following versions are required:

  - CSM 1.3.2 or later
  - AIX 5.2 5200-02 or later

- Web-based System Manager: IBM Web-based System Manager GUI is pre-installed on the Hardware Management Console and may be used as the interface for the Switch Network Manager (SNM) GUI application. However, the network manager software may be operated from a command-line interface on the HMC if you do not use the SNM application from the Web-based System Manager GUI interface.
- General Parallel File System (GPFS): GPFS is a scalable cluster file system that provides high-speed parallel file access and provides fault-tolerance including automatic recovery from disk and node failures. When used over a pSeries HPS network, only the VSD can be used. For more details related to GPFS and the HPS, refer to 3.5, “General Parallel File System 2.1” on page 114.
- LoadLeveler for AIX 5L: LoadLeveler is a serial and parallel job-management subsystem. LoadLeveler allocates resources, schedules jobs, and initiates them using Parallel Environment (PE) and its associated MPI software. PE and MPI get the system information they need from the LAPI User Space protocol.
- Parallel Environment (PE): PE calls LoadLeveler for resource assignment information, encodes that into environment variables, starts the user’s executable on the nodes assigned (as tasks of a parallel job), and passes the environment variables to those tasks via any TCP/IP interface. At the user’s task, the PE interface routine decodes the environment variables and uses that information to initialize the MPI library and its LAPI transport interface. Large MPI messages can be subdivided and sent as two or more concurrent submessages in parallel to improve bandwidth for a single task.

> **Note:** Although CSM software is optional for using the pSeries High Performance Switch, we recommend installing the CSM packages to facilitate the management of the LPARs connected to the HPS network.

See Table 4-3 for the versions of software needed in a pSeries HPS environment.

Table 4-3 Recommended software versions at the GA of pSeries HPS

Product	Comments
LAPI v2.3.1	Recommended PTF level is 1 (APAR: IY49650)
LL v3.2	Recommended PTF level is 1 (APAR: IY49652)

Product	Comments
PE v4.1	Recommended PTF level is 1 (APAR: IY49656
VSD v4.1.0	Recommended PTF level is 1 (APAR: IY49651)
GPFS v2.1	Recommended PTF level is 1 (APAR: IY47306)
CSM v1.3.2	
ESSL v4.1	
PESSL v3.1	
DPCL	Check for download package and further details at: `http://oss.software.ibm.com/dpcl`

HMC code

The HMC version required for installing the pSeries HPS is 3.2.5 (with PTF build level 20031020.1) or later. Check for the latest corrective service packages of HMC code on the AIX support Web page:

`https://techsupport.services.ibm.com/server/hmc`

If the HMC code is Release 1 or Release 2, you must upgrade the HMC to Release 3 and then apply the update R3V2.5 using the install corrective service function or perform a new installation of the HMC using Release 3 and then apply the corrective service package. Currently the R3V2.5 of the code is provided as an update, also referenced by APAR IY50216. See further details about installing the PTF in 4.2.1, “HMC installation” on page 151.

The new HMC code provides support for the operation of the pSeries HPS. The components of HMC code required in a pSeries HPS environment are:

- Switch Network Manager (SNM)

  The SNM software brings a new approach to the switch management functions. It provides the core functions for configuration, initialization, monitoring, diagnostic and control the switch network consisting of the switch plains and the SNI links.

  With SNM you don’t require the same level of control for the switch management like in previous versions of switches. Also, take into consideration that, for the pSeries HPS, the E-commands no longer exist.

- Service Focal Point (SFP)

  This software service provides a unified method for managing the serviceable events in an LPAR environment. Using the SFP interface, you can execute maintenance procedures such as examining the error log or performing a part replacement. The SFP is able to provide the Field Replaceable Units (FRUs) identification codes for the pSeries HPS, in case of a failure in the switch network.

- Service Agent

  The Service Agent is the software component provided in the HMC code for monitoring and notification of the hardware failures. The Service Agent receives events resulted from AIX diagnostic routines, Service Processor diagnostics, or SFP, and provides transport functions, such as automatic problem reporting, to IBM.

- Inventory Scout

  This is a set of diagnostic tools that provides two functions:

  - Microcode Discovery Service: Provides a microcode survey over the managed subsystems, showing the subsystem that may need to be upgraded.

- – VPD Capture Service: For transmitting Vital Product Data (VPD) information from the server to the IBM support center.

AIX support for pSeries HPS

AIX 5.2 ML02 is required by IBM @server pSeries High Performance Switch (HPS). The device driver for the SNI is now part of the AIX product. The SNI support code provides:

- IP Sockets
- Support for message-passing application programming interfaces (APIs), including K-LAPI and MPI

Support for HPS on AIX consists of two logical device names: snix and mlt0, where x is the device's minor number.

The logical device *sn#* refers to the logical device name of one of the external links on a 2-link or 4-link SNI. Each SNI has multiple links, and AIX considers each external link a distinct device. For example, the 4-link SNI for HPS will present four devices: sn0, sn1, sn2, and sn3. Each device will be administered independently. Attributes of the SNI can be changed using the **chgsni** command. For more details of the **chgsni** command, refer to *Switch Network Interface for eServer pSeries High Performance Switch Guide and Reference,* SC23-4869.

There is only one multilink interface per operating system image: the logical device named ml0. Only one multilink support interface occurs per operating system instance. The related IP network interface for this device is ml0. The ml0 interface distributes all of its network traffic over the Switch Network Interfaces. The ml0 IP network interface is configured like any other IP network interface, and it functions like any other IP interface. Sending network traffic over the ml0 interface is not necessary, but it can help improve performance by distributing the traffic over the sn# interfaces.

Important: The SNI device drivers require the 64-bit kernel of AIX.

When installing the SNI drivers, refer to Table 4-4 on page 149 for a list of components.

Table 4-4 AIX filesets for Switch Network Interface

Fileset name	Description
devices.common.IBM.sni.rte	Switch Network Interface Runtime
devices.common.IBM.sni.ml	Multi Link Interface Runtime
devices.common.IBM.sni.ntbl	Network Table Runtime
devices.chrp.IBM.HPS.rte	IBM eServer pSeries High Performance Switch (HPS) Runtime
devices.chrp.IBM.HPS.hpsfe	IBM pSeries HPS Functional Exerciser
devices.msg.en_US.common.IBM.sni.rte	Switch Network Interface Runtime Messages
devices.msg.en_US.common.IBM.sni.ml	Multi Link Interface Runtime
devices.msg.en_US.common.IBM.sni.ntbl	Network Table Runtime Messages
devices.msg.en_US.chrp.IBM.HPS.rte	pSeries HPS runtime Messages
devices.msg.en_US.chrp.IBM.HPS.hpsfe	pSeries HPS functional utility

Check for the latest available filesets on IBM technical support Web page:

```
https://techsupport.services.ibm.com/server/aix.fdc
```

Important: In order to be able to configure your AIX environment to work with pSeries HPS, you will need to use your server p690 or p655 in LPAR mode. A partitioning scheme can contain LPARs attached and not attached to the switch.

CSM scaling rules for Cluster 1600 with pSeries HPS

This section lists the scaling rules for Cluster 1600 systems configured with the pSeries HPS and using CSM for cluster management. In this configuration, the cluster may consist of two to 128 AIX operating system images. These operating system images or logical nodes can be:

- A p690 server (M/T 7040) running in full-system partition mode
- A logical partition (LPAR) of a 7040 server (up to four LPARs per CEC)
- A p655 server (M/T 7039) running in full-system partition mode
- A logical partition (LPAR) of a 7039 server (one per CEC)

The Cluster 1600 may be configured with multiple machine types. However, when running CSM, the cluster must meet all of the following limits for AIX operating system images installed:

- No more than 16 servers with 32 links from the set (7040, 7039)
- No more than 16 servers with 32 links from the set (7040)
- No more than 16 servers with 32 links from the set (7039)

When configuring the switched systems as part of a Cluster 1600 management domain, take into consideration the maximum number of the systems allowed in a CSM cluster. Refer to Table 2-11 on page 70 for more details.

Note: For the initial pSeries HPS offering, a Cluster 1600 can contain up to 16 servers or 32 links. A maximum configuration of 128 servers or logical partitions is available by special order.

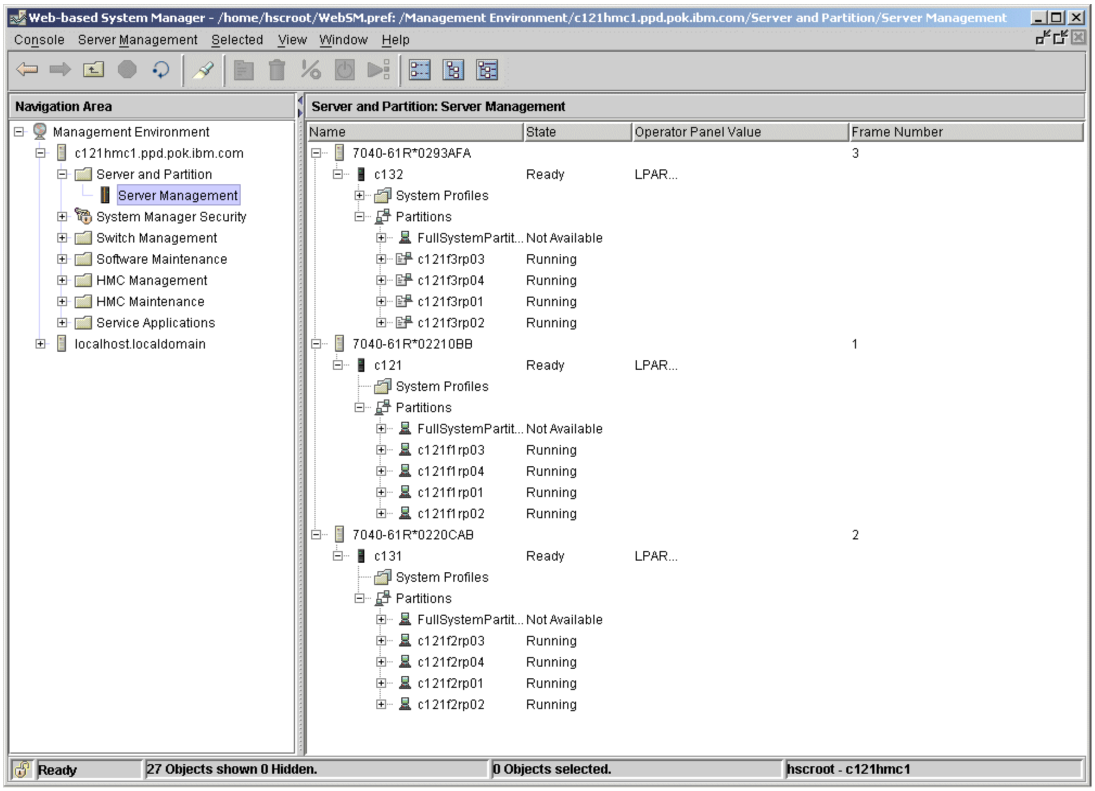
4.1.3 A sample test environment

For the examples presented in this chapter, we used an environment containing:

- Three pSeries 690 systems (7040-681)
- One HMC (7315-C01)
- Two pSeries HPS (7045-SW4) with 16 switch cards (copper switch port connection)

  The switches are mounted in the pSeries 690's rack 7040-61R. There is a two-switch, dual-plane configuration (no interconnection between the switches).
- pSeries 630 (7028-6E4), with CSM Server Version 1.3.2 installed

Each of the p690 system has 32 CPU and contains two 4-link SNI cards. There are four partitions on each CEC, each containing eight CPU and a link pair from the SNI adapters. Our partitions, taken from HMC, are shown in Figure 4-2.



Name	State	Operator Panel Value	Frame Number
7040-61R*0293AFA			3
c132	Ready	LPAR...	
System Profiles			
Partitions			
FullSystemPartit...	Not Available		
c121f3rp03	Running		
c121f3rp04	Running		
c121f3rp01	Running		
c121f3rp02	Running		
7040-61R*02210BB			1
c121	Ready	LPAR...	
System Profiles			
Partitions			
FullSystemPartit...	Not Available		
c121f1rp03	Running		
c121f1rp04	Running		
c121f1rp01	Running		
c121f1rp02	Running		
7040-61R*0220CAB			2
c131	Ready	LPAR...	
System Profiles			
Partitions			
FullSystemPartit...	Not Available		
c121f2rp03	Running		
c121f2rp04	Running		
c121f2rp01	Running		
c121f2rp02	Running		

Figure 4-2 LPARs connected to HPS

4.2 Installation and configuration of pSeries HPS

This section details the installation procedure of pSeries HPS. For most of the examples presented in this section, we used the environment described in 4.1.3, "A sample test environment" on page 150. We also provide general guidelines for different scenarios.

> **Note:** The procedures related to HMC are provided for the hscroot user, unless a different authorization level is required for a specific operation.

4.2.1 HMC installation

This section presents the sequence of operations to be performed to install the HMC software.

Back up the HMC configuration data

This section describes the methods used to save HMC critical data.

If you perform a new pSeries HPS installation with a new LPAR configuration, you can skip this step. Otherwise, before performing the installation of the new HMC code, we recommend that you back up the critical HMC information. There are several operations available for saving the HMC data, which are described in this section.

Save upgrade data

Perform this task if you plan for an upgrade installation of the HMC code immediately before the upgrade operation. The saved information includes:

- System preferences
- Profile information
- Service Agent files
- Inventory Scout Service files

The information can be stored on DVD-RAM or the HMC hard disk in a special partition.

Steps to perform:

1. From the Navigation Area of the HMC System Management, select **Software Maintenance**
2. Select **HMC**
3. Select **Save Upgrade Data**
4. Select either **DVD** or **Hard drive**

> **Tip:** If you see an HSCP0025 error message during the Save Upgrade Data operation, it may be due to insufficient space on /dev/hda2 or in the /var filesystem. Check the /var/hsc/log/SaveUpgradeFiles.log file.

Save profile data

Use this function to save only the LPAR profile data of your server.

Steps to perform:

1. In the **Server Management** window of the HMC System Management environment, right click the system icon of the targeted CEC for this operation.
2. Select **Profile Data**, and then **Backup**.
3. Enter the name of the backup file.

The file is saved in /var/hsc/profiles/<MTMS>/<backup file name>, where MTMS is a string containing the machine type (MT) and the serial number (MS) of the CEC.

Back up critical console data

This task saves important information about the HMC environment, such as:

- User preferences files
- HMC platform configuration files
- User information
- HMC log files

Note: This task requires that DVD-RAM media be inserted in the HMC DVD-RAM drive.

Steps to perform:

1. From the Navigation Area of the HMC System Management, select **Software Maintenance**.
2. Select **HMC**.
3. Select **Backup Critical Console Data**.

Note: If you need to restore your critical console data backup, you will need HMC base media of the same version and release as the currently installed one. Restoring a backup on a different HMC version or release may result in a corrupted HMC installation.

HMC software installation

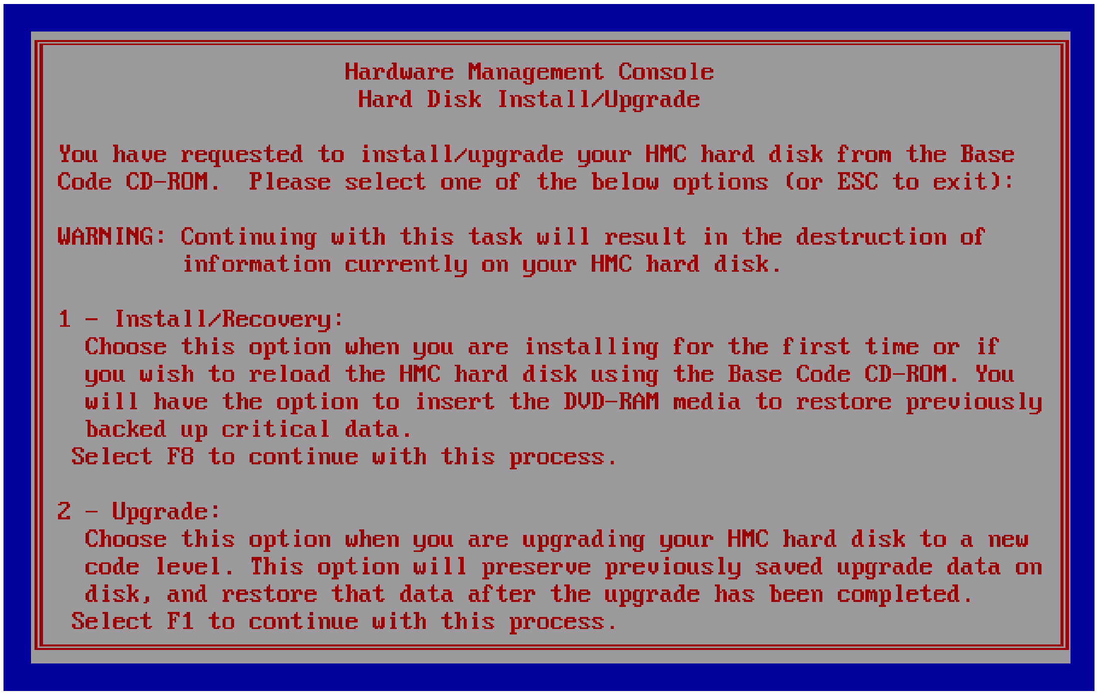
This procedure details the installation steps for a new installation of the HMC.

Note: Before starting this step, be sure that all the CECs are logically powered off (the LCD panel shows OK).

Boot the HMC from the CD-ROM containing the new release of HMC code.

At the Hardware Management Console, Hard Disk Install/Upgrade menu (see Figure 4-3) you can choose:

- **Install/Recovery** for a new installation of the HMC code
- **Upgrade** for an upgrade installation of the new HMC code



```
                    Hardware Management Console
                     Hard Disk Install/Upgrade

You have requested to install/upgrade your HMC hard disk from the Base
Code CD-ROM.  Please select one of the below options (or ESC to exit):

WARNING: Continuing with this task will result in the destruction of
         information currently on your HMC hard disk.

1 - Install/Recovery:
  Choose this option when you are installing for the first time or if
  you wish to reload the HMC hard disk using the Base Code CD-ROM. You
  will have the option to insert the DVD-RAM media to restore previously
  backed up critical data.
 Select F8 to continue with this process.

2 - Upgrade:
  Choose this option when you are upgrading your HMC hard disk to a new
  code level. This option will preserve previously saved upgrade data on
  disk, and restore that data after the upgrade has been completed.
 Select F1 to continue with this process.
```

Figure 4-3 Install/Upgrade HMC panel

If you press F8 (Install/recovery), a warning confirmation will be issued (press F1 to continue). When this process ends, remove the CD from the drive and press Enter.

If you press F1 (Upgrade), a confirmation will be issued (press F1 to continue). When this process finishes, remove the CD from the drive.

If the HMC doesn't boot from the CD, press F1 when the HMC is booting, enter the BIOS Setup Utility, and check if the CD (in fact DVD-RAM drive) is on the startup list.

HMC software configuration

If you performed an update installation of HMC, we recommend that you review this section and make the necessary changes. The following steps apply for a new installation of the HMC code.

Changing mouse, keyboard and time zone configuration

If you are not using a USB mouse and keyboard, you need to change the mouse and keyboard configuration. Follow the on-screen instructions during the first reboot of the HMC, after the installation process ends. Also you can check the date, time and time zone on the HMC. Refer to the operations guide of your HMC for further details.

IP configuration

From the HMC System Manager interface, do the following steps:

1. Select **HMC Maintenance**
2. Select **System Configuration**
3. Select **Customize Network Settings**

   Customize the network settings according to your environment. Some focus points:

IP address	TCP/IP interface 0 address TCP/IP interface 0 network mask Default Gateway
Name Services	DNS Enable DNS Server Search Order Domain Suffix Search Order
Host	Hostname
Devices adapter	Media speed for eth0

   Note: In order to activate these settings, you have to reboot the HMC.

4. Test the network connectivity with another valid host in the network. From the HMC System Manager interface:

   a. Select **HMC Maintenance**
   b. Select **System Configuration**
   c. Select **Test Network Connectivity**

   Provide a host name or an IP address of a valid host on the LAN (for example your CSM server).

Enabling the virtual terminal (optional)

You can enable the virtual terminal option on the HMC to be able to connect to the HMC using the Web-based System Manager Remote Client.

1. From the HMC System Management interface, select **HMC Maintenance**
2. Select **System Configuration**
3. Select **Enable or Disable Remote Virtual Terminal**
4. In the new window, check **Enable remote virtual terminal connections** and click **OK**.

Installing the corrective services

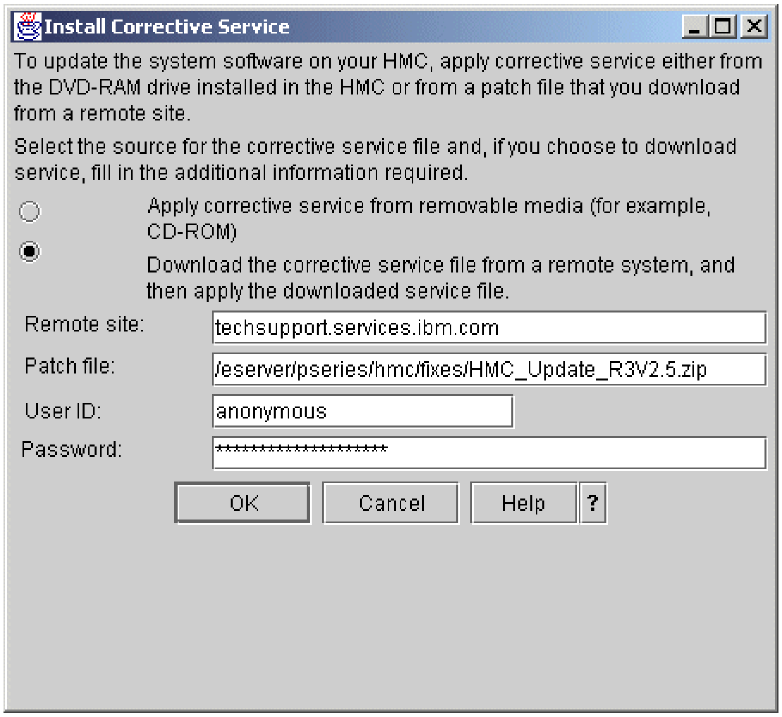
In this step, you will install the available updates for the HMC code. Check for the latest HMC corrective services on the IBM technical support Web page for your version of HMC:

```
https://techsupport.services.ibm.com/server/hmc
```

From the HMC System Management interface, perform the following tasks:

1. Select **HMC Maintenance**.
2. Select **HMC**.
3. Install **Corrective Services**. As the source, you can either choose removable media or to a remote location.

When choosing the remote location option for installing the corrective service, you can download the patch and apply it from a remote location (the file is downloaded in /usr/local/hsc_install.images), as shown in Figure 4-4.



Install Corrective Service

To update the system software on your HMC, apply corrective service either from the DVD-RAM drive installed in the HMC or from a patch file that you download from a remote site.

Select the source for the corrective service file and, if you choose to download service, fill in the additional information required.

○ Apply corrective service from removable media (for example, CD-ROM)

● Download the corrective service file from a remote system, and then apply the downloaded service file.

Remote site: techsupport.services.ibm.com

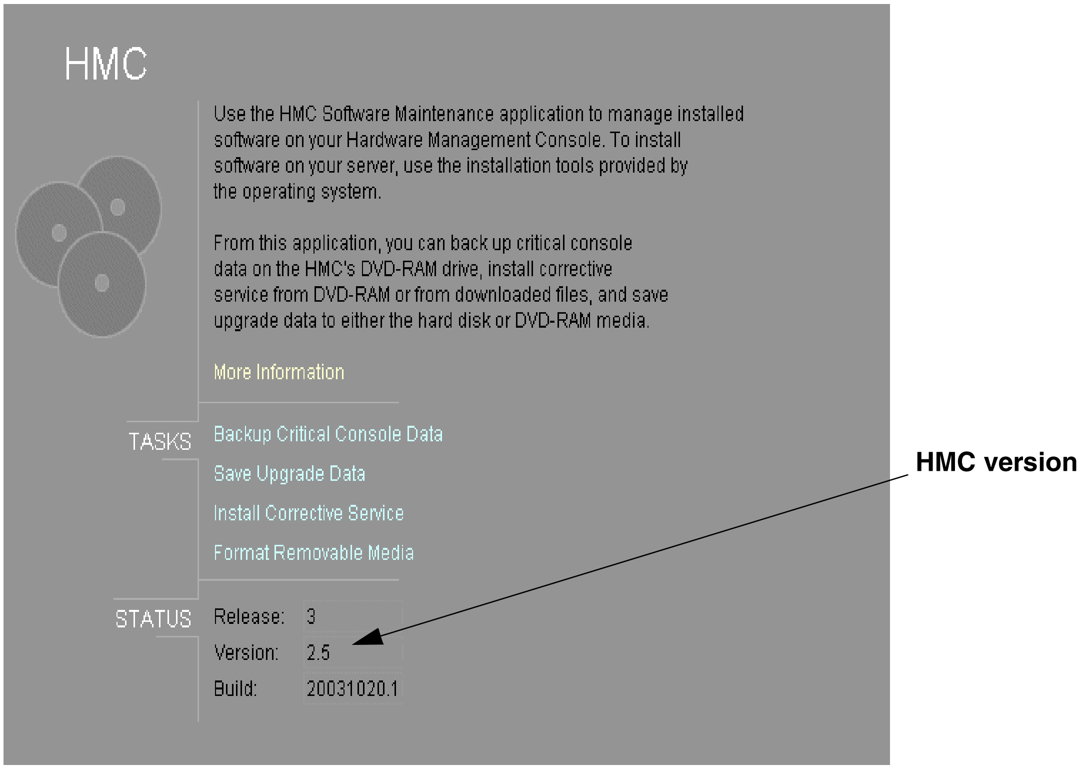
Patch file: /eserver/pseries/hmc/fixes/HMC_Update_R3V2.5.zip

User ID: anonymous

Password: *********************

Figure 4-4 Applying HMC corrective service

4. Check the version and the build date of the HMC Code by selecting **HMC Maintenance -> HMC** (see Figure 4-5 on page 155).


HMC

Use the HMC Software Maintenance application to manage installed software on your Hardware Management Console. To install software on your server, use the installation tools provided by the operating system.

From this application, you can back up critical console data on the HMC's DVD-RAM drive, install corrective service from DVD-RAM or from downloaded files, and save upgrade data to either the hard disk or DVD-RAM media.

Release: 3
Version: 2.5
Build: 20031020.1

Figure 4-5 Checking the version of the HMC

Tip: You can also check the version of your HMC in the file /opt/hsc/data/version.

Serial connectivity configuration

You need to set up your serial adapters for RS232 and RS422 connection between the HMC and the managed frames from your pSeries HPS environment.

From the HMC System Management interface, perform the following steps:

1. Select **HMC Maintenance**.
2. Select **System Configuration**.
3. Select **Configure Serial Adapter**.
4. In the new working window, type 1 to configure the serial adapters. Select how many serial adapters you want to configure and their type. In Example 4-1, we configure one IBM eight-port asynchronous adapter (FC2943).

Example 4-1 Configuring the serial adapter in HMC

```
 1 ) Configure Serial Adapter(s)
 2 ) Configure RS422 ports on an 8-port Serial Adapter
 3 ) Query a Serial Adapter
 0 ) Quit
-> 1

How many boards would you like to install? (1-12)1
Great! we'll install 1 board/s for you.

Board #1. What type of board is this? ('L' for list) (1-16)L
     1: Acceleport Xe ISA
     2: Acceleport Xr ISA
     3: Acceleport Xem ISA
     4: Acceleport Xi ISA
     5: Acceleport C/X ISA
     6: Acceleport Xem PCI
     7: Acceleport Xr PCI
     8: Acceleport C/X PCI
     9: Acceleport Xr(PLX) PCI
     10: Acceleport Xr-422
     11: Acceleport 2r-920 PCI
     12: Acceleport 4r-920 PCI
     13: Acceleport 8r-920 PCI
     14: Acceleport EPC/X PCI
     15: IBM 8-Port Async (PCI)
     16: IBM 128-Port Async (PCI)
Board #1. What type of board is this? ('L' for list) (1-16)15
Great! You've selected to install a IBM 8-Port Async (PCI) board!
Memory addresses will be read from the PCI card itself.
This digiBoard has 8 ports
Do you want to set Altpin on this board?  ('y' or 'n')n
Great! we have ALL the information we need, to get this digiBoard running!!

*** You must reboot the HMC so that the device drivers
*** for the serial adapters can be reloaded

 1 ) Configure Serial Adapter(s)
 2 ) Configure RS422 ports on an 8-port Serial Adapter
 3 ) Query a Serial Adapter
 0 ) Quit
-> 0
```

> **Important:** *All* serial adapters installed in the HMC (up to two of the same type) must be configured at the same time. When adding an additional adapter, the original adapter must also be reconfigured. If you do not reconfigure the original adapter, its definition will be lost.

5. Reboot the system to activate the changes.

   By default the serial ports are configured as RS232 ports. For the 128-port adapter, the RS422 connections to the frame are done by the hardware convertors. For the eight-port adapter, you have to set the RS422 mode for those specific ports connected to the frames. In this case, continue with the following steps:

6. Select **HMC Maintenance**

7. Select **System Configuration**

8. Select **Configure Serial Adapter**. A new window opens. Type `2` in this new window to configure RS422 ports on an eight-port adapter. Select the board, the port, and the RS422 option. Check the planning section in “HMC requirements” on page 141.

   In Example 4-2, we have set up the RS422 mode for the physical port 6 and 7 (port numbering from 0 to 7) of the eight-port adapter, connected to our CEC and frame.

Example 4-2 Setting up the RS422 ports on eight-port serial adapter

```
 1 ) Configure Serial Adapter(s)
 2 ) Configure RS422 ports on an 8-port Serial Adapter
 3 ) Query a Serial Adapter
 0 ) Quit
-> 2
```

```
 Select a board to configure

 1) 8-port board 0

 (x to quit) -> 1

Select the port change (x to exit)
  0  ttyD000  rs232
  1  ttyD001  rs232
  2  ttyD002  rs232
  3  ttyD003  rs232
  4  ttyD004  rs232
  5  ttyD005  rs232
  6  ttyD006  rs232
  7  ttyD007  rs232

 (x to quit) -> 6

Select the port change (x to exit)
  0  ttyD000  rs232
  1  ttyD001  rs232
  2  ttyD002  rs232
  3  ttyD003  rs232
  4  ttyD004  rs232
  5  ttyD005  rs232
  6  ttyD006  rs422
  7  ttyD007  rs232

 (x to quit) -> 7

Select the port change (x to exit)
  0  ttyD000  rs232
  1  ttyD001  rs232
  2  ttyD002  rs232
  3  ttyD003  rs232
  4  ttyD004  rs232
  5  ttyD005  rs232
  6  ttyD006  rs422
  7  ttyD007  rs422

 (x to quit) -> x

 Select a board to configure

 1) 8-port board 0

 (x to quit) -> x

1 ) Configure Serial Adapter(s)
 2 ) Configure RS422 ports on an 8-port Serial Adapter
 3 ) Query a Serial Adapter
 0 ) Quit
-> 0
```

Verifying the HMC serial connections (to CECs and frames)

Verify that the HMC code upgrade is complete by checking for server CECs and frame IDs:

1. Verify CECs:
   a. On the HMC GUI, select **Server and Partition -> Server Management**.
   b. Make sure that all server CECs show up on the Server Management panel.
2. Verify frame IDs:
   a. On the HMC GUI, select **Software Maintenance -> Frame -> Install Corrective Service**.
   b. Make sure that all frame IDs (MTMS) show up as available for service on the corrective service panel.

Installation of Web-based System Manager Remote Client (optional)

You can install a remote client to your workstation to perform the HMC GUI functions.

On your PC, open a browser window and go to `http://HMC_name/remote_client.html`, where `HMC_name` is the host name for your HMC.

On a Windows NT/2000 client, perform the following steps:

1. Save to disk.
2. Execute **`setup.exe`**.

4.2.2 Firmware upgrade

Upgrading the firmware consists of upgrading the General Firmware (GFW) and the frame firmware.

Installing the system firmware

The system firmware, also known as General Firmware (GFW), resides in the primary I/O book. We recommend that you do this upgrade first. The minimum firmware version required for operating with pSeries HPS is:

- pSeries 690: 3H031024
- pSeries 655: 3J031024

There are several ways to update the system firmware, depending on the pSeries type. For downloading the latest firmware code and the instructions for installing it, refer to:

`http://techsupport.services.ibm.com/server/mdownload2`

> **Note:** Prior to AIX installation, p690 firmware may be updated from service processor menus using a floppy disk drive. For 655, the microcode may be updated via NIM. Always check the release version and the install instructions accompanying the firmware.

For both p690 and p655, there is an alternate procedure for upgrading the system firmware, by using an LPAR with AIX installed. This procedure requires the following:

- The LPAR to be used must have service authority
- All other partitions except the one with service authority must be shut down
- The partition with service authority must have the update image on a local file system.

Steps for installing the system firmware:

1. Log in as root to the LPAR with the service authority. All other LPARs must be shut down.

2. Create a directory /tmp/fwupdate if does not exist:

   ```
   mkdir /tmp/fwupdate
   ```

3. Download the proper firmware image from the IBM support site, expand it and store the image file in the /tmp/fwupdate directory.
4. Run the following commands:

   ```
   cd /usr/lpp/diagnostics/bin
   ./update_flash -f /tmp/fwupdate/<image_file_name>
   ```

5. Follow the indicated instructions on the screen. The procedure will shut down the system.

Installing the frame firmware

Important: Before the installation of the frame firmware make sure that the CEC is logical powered off (the LCD panel shows OK).

Installing the frame firmware for pSeries 690 systems

Use the following procedure for upgrading the power subsystem microcode of a p690 system:

1. Disconnect the SPCN cables from the J00B (bottom) port on BPC A and BPC B.
2. Connect the RS-422 cables from the HMC to the J00B (bottom) port on BPC A and BPC B

> **Note:** On the p690 server, the serial cables connecting the HMC to the BPC are normally connected to the J00A (top) port on the BPCs. However, the J00A ports must be enabled for use with the pSeries HPS. During this procedure, you will do the initial code load through the J00B ports. After you complete that process, you will move the RS-422 cables from the J00B ports to the J00A ports. Then, with the serial cables on the J00A ports, you will reinstall the microcode to enable the top ports for use with the pSeries HPS.

3. On the HMC System Management Interface, on the Software Maintenance menu, then perform the following steps:
   a. Select **Frame**.
   b. Select **Install Corrective Service**.
   c. Check the frame code level (next to Frame MTMS in the lower panel).

      If the frame code level is 259f or higher, continue with step 4. If the frame code level is below 259f, do the following:

      i. Return to the Frame panel of the Software Maintenance menu.
      ii. Select **Receive Corrective Service**.
      iii. Place the diskette with the ptcode into the HMC diskette drive.
      iv. Click **OK** to upload from the diskette.
      v. When the upload is complete, return to the Frame panel of the Software Maintenance menu.

   d. Select **Install Corrective Service**.
   e. Highlight the Installed Version and the Frames to be updated fields. Highlight the appropriate corrective service version in the upper part of the Corrective Service panel and then select the frames to be updated in the lower part of the panel.
   f. Select **Install** from the Install Corrective Service panel.

> **Note:** The initial microcode install process may take up to one hour to complete. The second install (to enable the J00A ports) will only take a few minutes.

4. If the code level is correct or the microcode install completed, move the RS422 cables from the J00B (bottom) ports to the J00A (top) ports on the BPCs.

   > **Note:** If there are cables attached to the J00A ports, disconnect the cables from the J00A ports and label the cable ends as not used or if possible, disconnect both ends of the cables and completely remove them from the frame.

5. Reattach the SPCN cables to the J00B ports.
6. Refresh the GUI to update the frame code display.
7. After you have attached the RS-422 cables to the J00A ports and the SPCN cables to the J00B ports, reinstall the microcode to enable the J00A ports for use with the pSeries HPS:
   a. Return to the Software Maintenance menu.
   b. Select **Frame**.
   c. Select **Install Corrective Service**.
   d. Highlight the Installed Version and the Frames to be updated fields. Highlight the appropriate corrective service version in the upper part of the Corrective Service panel and the select the frames to be updated in the lower part of the panel
   e. Select **Install** from the **Install Corrective Service** panel
8. When the code is loaded and complete, power off the system from UEPO switch.

Installing the frame firmware for pSeries 655 systems

Perform the following procedure for loading the power subsystem code on a p655 frame:

1. Connect the RS-422 cables from the HMC to the J00A (top) port on BPC A and BPC B
2. On the HMC System Management Interface, Select **Software Maintenance**, then perform these steps:
   a. Select **Frame**.
   b. Select **Install Corrective Service**.
   c. Check the Frame code level (next to Frame MTMS in the lower panel).

      If the frame code level is 259f or higher, continue with step 3. If the frame code level is below 259f, do the following:
      i. Return to the Frame panel of the Software Maintenance menu.
      ii. Select **Receive Corrective Service**.
      iii. Place the diskette with the ptcode into the HMC diskette drive.
      iv. Click **OK** to upload from the diskette.
      v. When the upload is complete, return to the Frame panel of the Software Maintenance menu.
   d. Select **Install Corrective Service**.
   e. Highlight the Installed Version and the Frames to be updated fields. Highlight the appropriate corrective service version in the upper part of the Corrective Services panel and then select the frames to be updated in the lower part of the panel.
   f. Select **Install** from the Install Corrective Service panel.

3. When the code is loaded and complete, power off the system from the UEPO switch.

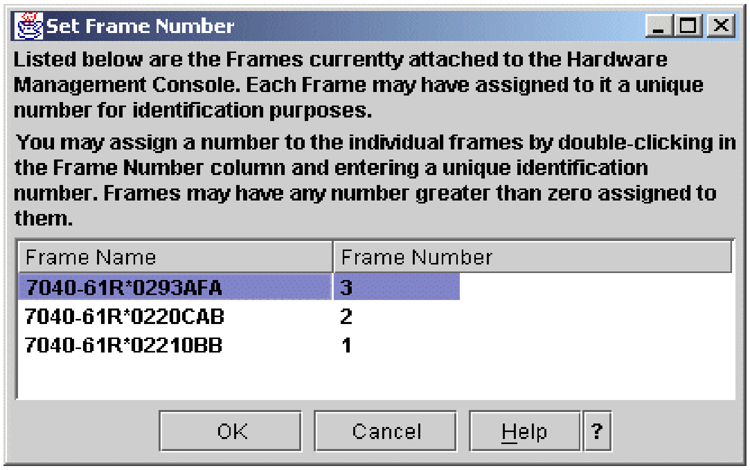
Note: HMC GUI for Updating the frame microcode uses the `frucode` command. It may fail if redundant power subsystems are not installed and active on your frame (for both p690 or p655 systems). In this case, you can use the `instfru` command from the HMC command line to update the frame firmware. Refer to A.2.4, “Upgrading the frame microcode using the instfru command” on page 229 for more details. The frame code must be downloaded on HMC prior to running the `instfru` command.

4.2.3 Frame configuration

After you have performed the frame power code upgrade and you have turned off the UEPO switch, you have to set the frame number to your frame. On the HMC, perform the following steps:

1. Select **Switch Management**.
2. Select **Switch Utilities**.
3. Select **Frame Number Configuration**.

   Set the frame number as shown in Figure 4-6.



Set Frame Number

Listed below are the Frames currently attached to the Hardware Management Console. Each Frame may have assigned to it a unique number for identification purposes.

You may assign a number to the individual frames by double-clicking in the Frame Number column and entering a unique identification number. Frames may have any number greater than zero assigned to them.

Frame Name	Frame Number
7040-61R*0293AFA	3
7040-61R*0220CAB	2
7040-61R*02210BB	1

Figure 4-6 Setting the frame number

Note: You can mix switched and non-switched frames in an environment. HMC can manage both numbered and unnumbered frames, but all the frames connected to the pSeries HPS must have a frame number assigned.

4.2.4 LPAR configuration

In this step, you can create or upgrade your LPARs with the SNI interfaces for connection to the switch.

Creating a new LPAR

Perform the following steps for creating a new LPAR attached to the switch:

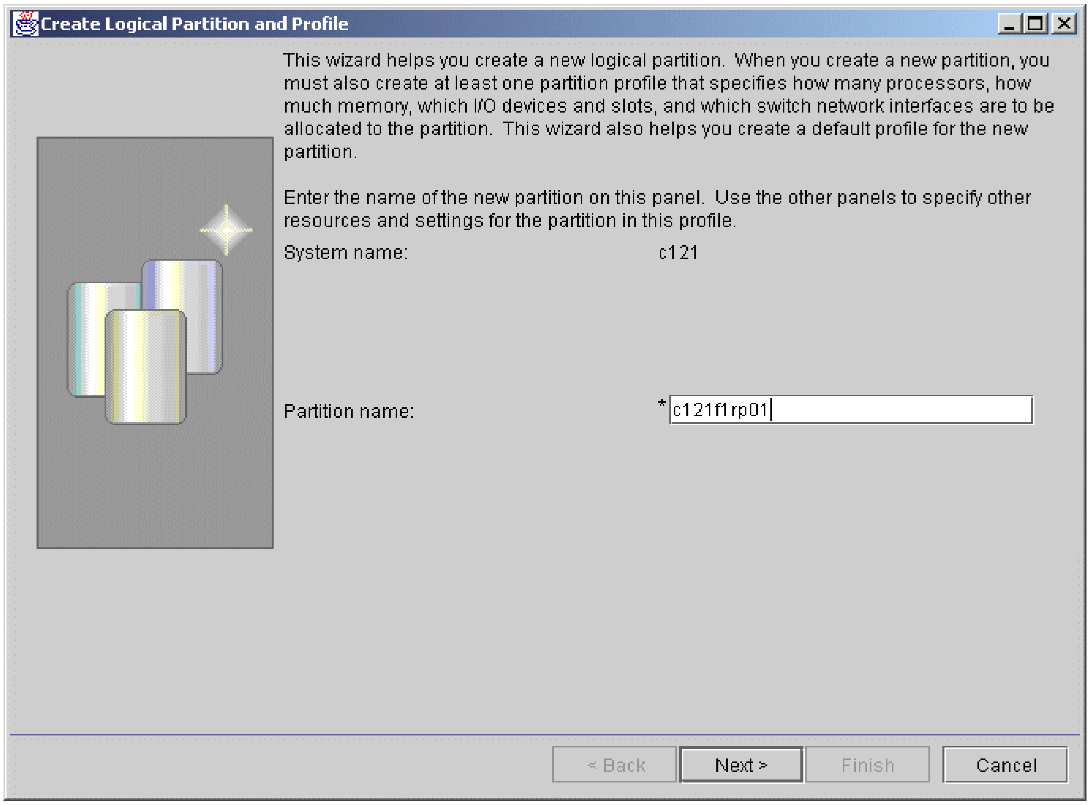
1. Boot the CEC in LPAR mode:
   a. In the Server and Partition window, select the frame you want to start.
   b. Right-click it and select **Power On**.

c. Select **Partition Standby** mode.

Wait for the LCD panel on HMC to display LPAR.

2. Select the frame on which you want to create LPARs.
3. Right-click it and select **Create** -> **Logical Partition**.

A new wizard is started for beginning the configuration of the LPAR (see Figure 4-7). Set a name for the partition and continue with **Next**.



This wizard helps you create a new logical partition. When you create a new partition, you must also create at least one partition profile that specifies how many processors, how much memory, which I/O devices and slots, and which switch network interfaces are to be allocated to the partition. This wizard also helps you create a default profile for the new partition.

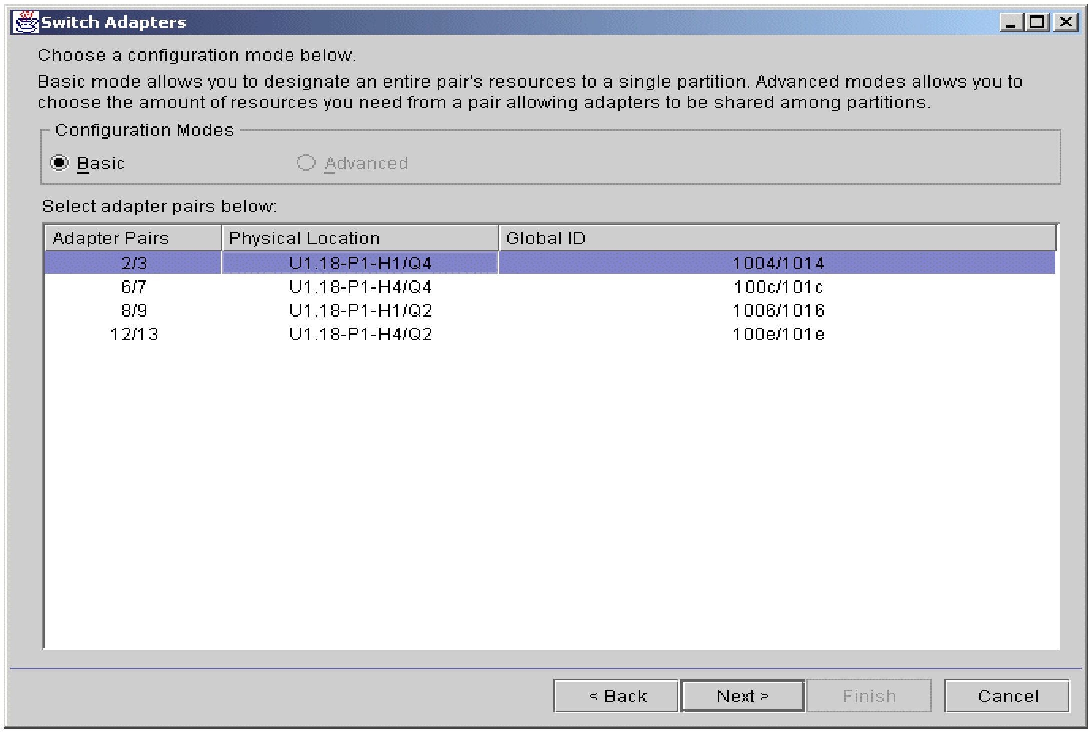
Enter the name of the new partition on this panel. Use the other panels to specify other resources and settings for the partition in this profile.

System name: c121

Partition name: * c121f1rp01

Figure 4-7 Create a new LPAR

4. Select the CPU, Memory and I/O resources for the new partition. After providing the CPU and memory resources, you are requested to select the SNI interfaces for your partition, as shown in Figure 4-8.



Choose a configuration mode below.

Basic mode allows you to designate an entire pair's resources to a single partition. Advanced modes allows you to choose the amount of resources you need from a pair allowing adapters to be shared among partitions.

Configuration Modes

● Basic ○ Advanced

Select adapter pairs below:

Adapter Pairs	Physical Location	Global ID
2/3	U1.18-P1-H1/Q4	1004/1014
6/7	U1.18-P1-H4/Q4	100c/101c
8/9	U1.18-P1-H1/Q2	1006/1016
12/13	U1.18-P1-H4/Q2	100e/101e

Figure 4-8 Assign the SNI interfaces to LPAR

Notes:

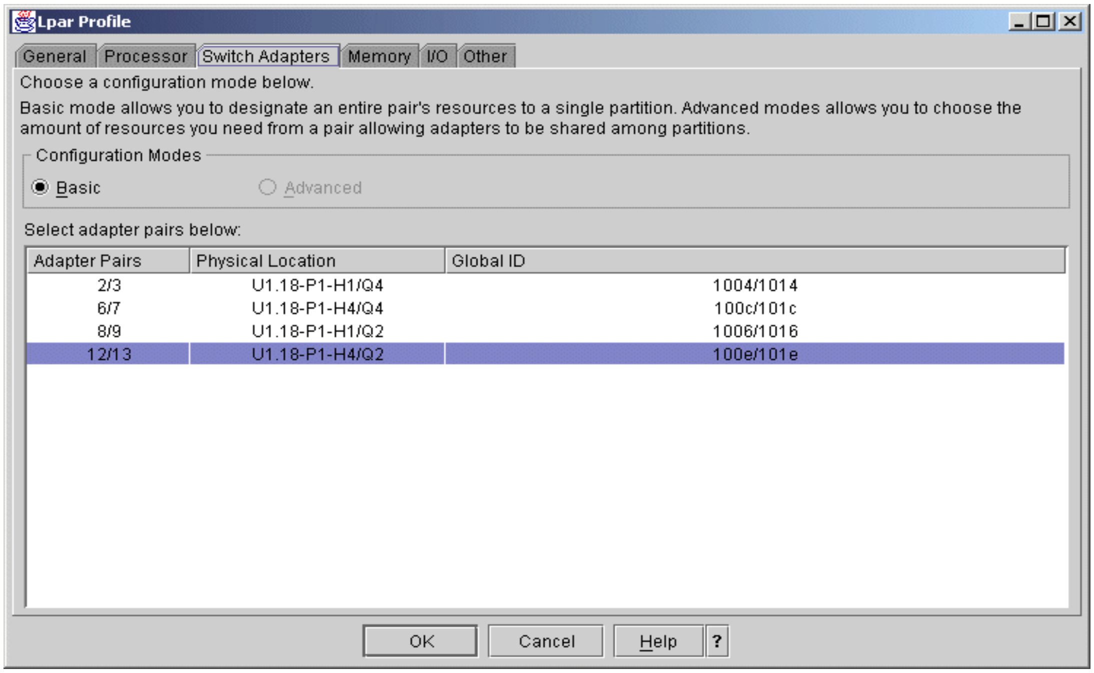
- Q2 and Q4 in Figure 4-8 refer to the two link pairs of a 4-link SNI adapter. The ports of an SNI adapter can be allocated to partitions only in pairs. A link pair cannot be split between partitions.
- The Global ID must not be empty; otherwise you can allocate the link pair to the LPAR, but you won't be able to boot it. This may happen when a four-port SNI card is installed in a p690 system with only two MCMs.

Refer to *pSeries High Performance Switch Planning, Installation and Service*, GA22-7951, for more details related to the physical location of the SNI ports.

Upgrading an existing LPAR

If you are upgrading an existing LPAR environment for connection to the pSeries HPS, then you should change the LPAR properties and add the desired SNI interfaces to the LPAR. Perform the following steps:

1. From the Server and Partition window, expand the CEC icon containing the selected LPAR.
2. Right-click the LPAR name and select **Properties**.
3. Click the **Switch Adapters** tab and select the appropriate SNI interfaces. See Figure 4-9 on page 164.



Choose a configuration mode below.

Basic mode allows you to designate an entire pair's resources to a single partition. Advanced modes allows you to choose the amount of resources you need from a pair allowing adapters to be shared among partitions.

Configuration Modes

● Basic ○ Advanced

Select adapter pairs below:

Adapter Pairs	Physical Location	Global ID
2/3	U1.18-P1-H1/Q4	1004/1014
6/7	U1.18-P1-H4/Q4	100c/101c
8/9	U1.18-P1-H1/Q2	1006/1016
12/13	U1.18-P1-H4/Q2	100e/101e

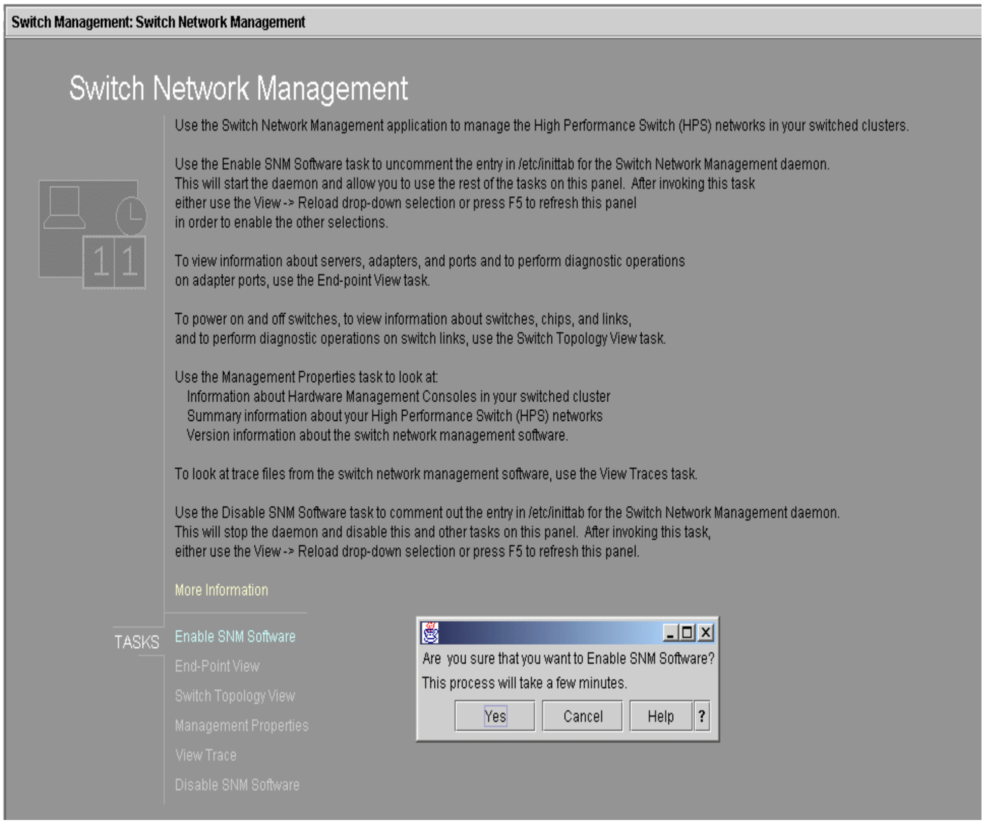
Figure 4-9 Adding the SNI interfaces to an existing LPAR

Note: The SNI interfaces cannot be dynamically allocated. For the changes to become effective you must shut down or reset the partition and wait approximately 30 seconds prior to activation.

4.2.5 Starting the switch

To start the switch, the SNM functions on the HMC have to be enabled by performing the following steps:

1. Power off all the CECs using the HMC. Wait until the messages “`No Connection`” and “`OK`” appear on the HMC console in the Server and Partition window.
2. From the HMC System Management Interface:
   a. Select **Switch Management**.
   b. Select **Enable the SNM Software**. Figure 4-10 on page 165 is displayed.



Use the Switch Network Management application to manage the High Performance Switch (HPS) networks in your switched clusters.

Use the Enable SNM Software task to uncomment the entry in /etc/inittab for the Switch Network Management daemon.
This will start the daemon and allow you to use the rest of the tasks on this panel. After invoking this task
either use the View -> Reload drop-down selection or press F5 to refresh this panel
in order to enable the other selections.

To view information about servers, adapters, and ports and to perform diagnostic operations
on adapter ports, use the End-point View task.

To power on and off switches, to view information about switches, chips, and links,
and to perform diagnostic operations on switch links, use the Switch Topology View task.

Use the Management Properties task to look at:
Information about Hardware Management Consoles in your switched cluster
Summary information about your High Performance Switch (HPS) networks
Version information about the switch network management software.

To look at trace files from the switch network management software, use the View Traces task.

Use the Disable SNM Software task to comment out the entry in /etc/inittab for the Switch Network Management daemon.
This will stop the daemon and disable this and other tasks on this panel. After invoking this task,
either use the View -> Reload drop-down selection or press F5 to refresh this panel.

Are you sure that you want to Enable SNM Software?
This process will take a few minutes.

Figure 4-10 Enable the SNM Software

3. Reboot the HMC.
4. Power on the CEC by activating the Partition Standby Mode to boot the CEC in LPAR mode.
5. Check the HPS is operational:
   a. Log in to a terminal window by pressing Ctrl+Alt+F1. You can switch back to the graphical mode by pressing Ctrl+Alt+F2.
   b. Switch the user to root and run the following command:

      ```
      /usr/local/hsctool/hps_check.pl
      ```

      Check the output of this command to contain valid information (look at “TIME”, “MPA” and “TOD”). See an output example of this script in A.2.5, “Output example of /usr/local/hsctool/hps_check.pl script” on page 229.

4.2.6 Configuring CSM and NIM

This step is performed when the pSeries HPS is part of a Cluster 1600 environment. We recommend using CSM for installing and managing the switched LPARs.

If you make a new installation of HMC, it is a good idea in this step to reboot the CSM Manager.

This section describes the steps involved in defining the LPARs in CSM and installing AIX. This process assumes that you have an installed CSM server, you have previously initialized the CSM master, and you are installing new LPARs. For more details about the CSM server installation and configuration, refer to *IBM Cluster Systems Management for AIX 5L Planning* *and Installation Guide*, SA22-7919. Before defining the CSM nodes, assure that the HMC is running and the CECs are powered on in partition standby mode.

Creating the CSM nodes

> **Important:** Before CSM nodes are created, it is mandatory that the name resolution is properly set up. Verify the /etc/hosts file on both the HMC and the CSM Master. If DNS is used, verify that the /etc/resolv.conf file is identical among managed nodes and CSM Management Server. Direct and reverse host name lookup results should be identical for all systems from all systems. Incorrect name resolution will prevent RSCT from working properly (especially CtSec).

Perform the following steps:

1. Create the access to the HMC. For the CSM server to be able to manage the LPARs, it has to be able to access the HMC managing those LPARs. In Example 4-3, the HMC is c121hmc1.ppd.pok.ibm.com and the user accessing the HMC resources is hscroot.

Example 4-3 A systemid command example

```
# systemid c121hmc1.ppd.pok.ibm.com hscroot
Password:
Verifying, please re-enter password:
systemid:
Entry created.
```

2. Gather hardware info about the LPARs managed by the HMC. In Example 4-4, the **lshwinfo** command is used for collecting information from the HMC c121hmc1 about the managed partitions.

Example 4-4 Collecting the LPAR information on the CSM server

```
# lshwinfo -p hmc -c c121hmc1.ppd.pok.ibm.com -o /tmp/c121hmc1.txt
# cat /tmp/c121hmc1.txt
# Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParID::HWType::HWMode
l::HWSerialNum
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp04::004::7040::681::0220CBB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp03::003::7040::681::0220CBB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp04::004::7040::681::02210CB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp02::002::7040::681::0220CBB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp04::004::7040::681::0293B0A
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp03::003::7040::681::02210CB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp01::001::7040::681::0220CBB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp03::003::7040::681::0293B0A
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp02::002::7040::681::02210CB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp02::002::7040::681::0293B0A
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp01::001::7040::681::02210CB
no_hostname::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp01::001::7040::681::0293B0A
```

3. Edit the hostmap file and set the host name of the LPAR in the first field as in Example 4-5.

Example 4-5 The hostmap file after setting the hostnames

```
Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParID::HWType::HWModel::HWSerialNu
m
c121f2rp04::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp04::004::7040::681::0220CBB
c121f2rp03::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp03::003::7040::681::0220CBB
c121f1rp04::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp04::004::7040::681::02210CB
c121f2rp02::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp02::002::7040::681::0220CBB
c121f3rp04::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp04::004::7040::681::0293B0A
```

```
c121f1rp03::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp03::003::7040::681::02210CB
c121f2rp01::hmc::c121hmc1.ppd.pok.ibm.com::c121f2rp01::001::7040::681::0220CBB
c121f3rp03::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp03::003::7040::681::0293B0A
c121f1rp02::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp02::002::7040::681::02210CB
c121f3rp02::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp02::002::7040::681::0293B0A
c121f1rp01::hmc::c121hmc1.ppd.pok.ibm.com::c121f1rp01::001::7040::681::02210CB
c121f3rp01::hmc::c121hmc1.ppd.pok.ibm.com::c121f3rp01::001::7040::681::0293B0A
```

4. Define the nodes in CSM. The collected LPARs are added as endpoint nodes to the CSM server. Example 4-6 uses the LPARs definitions from Example 4-5 on page 166.

Example 4-6 Defining LPARs as CSM managed nodes

```
# definenode -M /tmp/c121hmc1.txt
Defining CSM Nodes:
Defining Node "c121f2rp04.ppd.pok.ibm.com"("9.114.66.76")
Defining Node "c121f2rp03.ppd.pok.ibm.com"("9.114.66.75")
Defining Node "c121f1rp04.ppd.pok.ibm.com"("9.114.66.68")
Defining Node "c121f2rp02.ppd.pok.ibm.com"("9.114.66.74")
Defining Node "c121f3rp04.ppd.pok.ibm.com"("9.114.66.84")
Defining Node "c121f1rp03.ppd.pok.ibm.com"("9.114.66.67")
Defining Node "c121f2rp01.ppd.pok.ibm.com"("9.114.66.73")
Defining Node "c121f3rp03.ppd.pok.ibm.com"("9.114.66.83")
Defining Node "c121f1rp02.ppd.pok.ibm.com"("9.114.66.66")
Defining Node "c121f3rp02.ppd.pok.ibm.com"("9.114.66.82")
Defining Node "c121f1rp01.ppd.pok.ibm.com"("9.114.66.65")
Defining Node "c121f3rp01.ppd.pok.ibm.com"("9.114.66.81")

# lsnode
p690_LPAR1.itso.ibm.com
p690_LPAR2.itso.ibm.com
c121f2rp04.ppd.pok.ibm.com
c121f2rp03.ppd.pok.ibm.com
c121f1rp04.ppd.pok.ibm.com
c121f2rp02.ppd.pok.ibm.com
c121f3rp04.ppd.pok.ibm.com
c121f1rp03.ppd.pok.ibm.com
c121f2rp01.ppd.pok.ibm.com
c121f3rp03.ppd.pok.ibm.com
c121f1rp02.ppd.pok.ibm.com
c121f3rp02.ppd.pok.ibm.com
c121f1rp01.ppd.pok.ibm.com
c121f3rp01.ppd.pok.ibm.com
```

5. Create node groups. In Example 4-7 we create three node groups associated with the CEC frames of the nodes.

Example 4-7 Creating node groups

```
# nodegrp -n c121f1rp01.ppd.pok.ibm.com,c121f1rp02.ppd.pok.ibm.com,\
> c121f1rp03.ppd.pok.ibm.com,c121f1rp04.ppd.pok.ibm.com frame1_grp
# nodegrp -n c121f2rp01.ppd.pok.ibm.com,c121f2rp02.ppd.pok.ibm.com,\
> c121f2rp03.ppd.pok.ibm.com,c121f2rp04.ppd.pok.ibm.com frame2_grp
# nodegrp -n c121f3rp01.ppd.pok.ibm.com,c121f3rp02.ppd.pok.ibm.com,\
> c121f3rp03.ppd.pok.ibm.com,c121f3rp04.ppd.pok.ibm.com frame3_grp

# nodegrp |grep frame
frame1_grp
frame2_grp
frame3_grp
```

Installing AIX on the LPARs

1. Run the **rpower** command on the management server. This command is used for querying the status of the nodes as shown in Example 4-8.

Example 4-8 rpower command example

```
# rpower -N frame1_grp,frame2_grp,frame3_grp query
c121f2rp04.ppd.pok.ibm.com off
c121f2rp03.ppd.pok.ibm.com off
c121f1rp04.ppd.pok.ibm.com off
c121f2rp02.ppd.pok.ibm.com off
c121f3rp04.ppd.pok.ibm.com off
c121f1rp03.ppd.pok.ibm.com off
c121f2rp01.ppd.pok.ibm.com off
c121f3rp03.ppd.pok.ibm.com off
c121f1rp02.ppd.pok.ibm.com off
c121f3rp02.ppd.pok.ibm.com off
c121f1rp01.ppd.pok.ibm.com off
c121f3rp01.ppd.pok.ibm.com off
```

2. Get adapter MAC addresses of the LPAR adapters used for the installation. Example 4-9 shows a method for getting the MAC address of a node using the **getadapters** command and storing it in CSM. You can choose to add the MAC address of your node manually if it is appropriate.

Example 4-9 Gather adapter information in CSM

```
# getadapters -w -t ent -D -s 100 -d full -n c121f1rp01.ppd.pok.ibm.com

# Name::Adapter Type::MAC Address::Location Code::Adapter Speed::Adapter Duplex:
:Install Server::Adapter Gateway::Ping Status
Acquiring adapter information using dsh.
Can not use dsh - No nodes in Managed or MinManaged mode.
Acquiring adapter information from Open Firmware for node c121f1rp01.ppd.pok.ibm.com.
c121f1rp01.ppd.pok.ibm.com::ent::0002556A5352::U1.9-P1-I3/E1::100::full::csm_server.ppd.pok
.ibm.com::0.0.0.0::ok

#lsnode -l c121f1rp01.ppd.pok.ibm.com|grep InstallAdapterMacaddr
 ChangedAttributes =
{InstallAdapterMacaddr,InstallAdapterType,InstallAdapterSpeed,InstallAdapterDuplex}
 InstallAdapterMacaddr = 0002556A5352
```

In Example 4-9, the installation adapter is set to 100_full_duplex mode. You should choose the settings according to your network environment. You can check the adapter attributes in the CSM node definition using the **lsnode** command and make the necessary changes if needed, using the **chnode** command.

> **Note:** In CSM 1.3.2.1, the **getadapter** command first tries to establish a remote shell connection to the node, only if that node has the attribute Mode set to Managed or MinManaged. If the remote shell connection fails, the **getadapter** command tries to use the hardware control method, which may result in rebooting your LPAR at the open firmware prompt (assuming hardware control is available).

3. Configure the NIM environment:

   a. Create the NIM resources

– The lpp_source and spot resources

The lpp_source should contain the AIX 5.2 package and the patches for ML02. Also, apply the SNI filesets to the lpp_source and create the SPOT as shown in Example 4-10.

Example 4-10 Creating the lpp_source and SPOT

```
Create the lppsource_52ML2 resource
# nim -o define -t lpp_source -a source=/dev/cd0 -a server=master \
> -a location=/csminstall/AIX/aix520/lppsource lppsource_aix52ML2

Update the lppsource with the ML02 package from a local directory: /tmp/AIXML02
# nim -o update -a packages=all -a source=/tmp/AIXML02 lppsource_aix52ML2

Update the lppsource with the SNI file sets from directory: /tmp/SNI
# nim -o update -a packages=all -a source=/tmp/SNI lppsource_aix52ML2

Create the SPOT
# nim -o define -t spot -a source=lppsource_aix52ML2 -a server=master \
> -a location=/csminstall/export/spot52_002/ spot52_ML2
```

Example 4-10 assumes that the AIX package is loaded from CD-ROM media. The patches and the SNI filesets are downloaded to local directories. For a list of SNI filesets, refer to Table 4-4 on page 149.

– Customize a bosinst_data resource

You can use the template file /usr/lpp/bosinst/bosinst.template and customize it. For an example of bosinst.data file we used in our scenario, see Example A-1 on page 222. For creating the bosinst_data resource, see Example 4-11.

Example 4-11 Create a bosinst_data resource

```
# nim -o define -t bosinst_data -a server=master \
> -a location=/csminstall/bosinst.data bosinst_data_52
```

– Define a resolv_conf resource

You can define a resolv_conf resource in NIM for setting the name resolution configuration on the nodes. Example 4-12 creates a resource named resolv_conf_52, using the file /csminstall/resolv.conf.

Example 4-12 Define a resolv_conf resource

```
# nim -o define -t resolv_conf -a server=master -a location=/csminstall/resolv.conf \
> resolv_conf_52
```

– Define the default route in NIM

You can define the default route in the network NIM resource. See Example 4-13.

Example 4-13 Define a default route in the nim_network resource

```
# nim -o change -a routing1="default 9.12.6.92" nim_network
```

– At this point you can choose to define the configuration for the HPS adapters in NIM resource, using the secondary adapter definition feature. This is detailed in “Configuring the SNI adapters using NIM” on page 171.

b. Create NIM node environment from CSM. At this step, use the **csm2nimgrps** and **csm2nimnodes** scripts to create the NIM nodes and groups from the CSM definitions. In

Example 4-14, we used the nodes and groups previously created in Example 4-6 on page 167 and Example 4-7 on page 167.

Example 4-14 Create the NIM nodes and node groups

```
# csm2nimnodes -N frame1_grp,frame2_grp,frame3_grp type=standalone platform=chrp \
> netboot_kernel=mp network_name=nim_network cable_type="N/A"
# csm2nimgrps -N frame1_grp,frame2_grp,frame3_grp

Check that the NIM definitions were created:
# lsnim
```

> **Note:** The csm2nimnodes fails if the CSM nodes do not have the network attributes defined in CSM. Check the network attributes of the node using **lsnode** command.

c. Setting up the CSM client software to be installed on the nodes. See Example 4-15.

Example 4-15 Setting the CSM client to be installed on the nodes

```
# csmsetupnim -N frame1_grp,frame2_grp,frame3_grp
```

Use **lsnim** to check that a new resource, csmprereboot_script, was created.

d. Set up nodes to be installed from the network. See Example 4-16.

Example 4-16 Set up nodes to be installed from the network

```
# nim -o allocate -a spot=spot52_ML2 -a lpp_source=lppsource_52ML2 \
> -a bosinst_data=bosinst_data_52 -a resolv_conf=resolv_conf_52 frame1_grp

# nim -o allocate -a spot=spot52_ML2 -a lpp_source=lppsource_52ML2 \
> -a bosinst_data=bosinst_data_52 -a resolv_conf=resolv_conf_52 -a group=frame2_grp

# nim -o allocate -a spot=spot52_ML2 -a lpp_source=lppsource_52ML2 \
> -a bosinst_data=bosinst_data_52 -a resolv_conf=resolv_conf_52 -a group=frame3_grp

# nim -o bos_inst -a source=rte -a boot_client=no -a accept_licenses=yes frame1_grp
# nim -o bos_inst -a source=rte -a boot_client=no -a accept_licenses=yes frame2_grp
# nim -o bos_inst -a source=rte -a boot_client=no -a accept_licenses=yes frame3_grp
```

e. Network boot the target nodes:

Issue a **netboot** command for booting the nodes from the network. Example 4-17 shows how to perform a network boot for a group of nodes.

Example 4-17 Network boot a group of nodes

```
# netboot -N frame1_grp
```

From a graphical console, you can open a terminal window for monitoring the installation process on the node. Issue: **rconsole -n node_name**

4.2.7 Installing and configuring the SNIs in AIX

This section details the steps involved in the installation and configuration of the SNI adapters. Use this procedure when you add the SNI adapters to an LPAR with AIX installed. There are two procedures detailed: the first uses NIM, and the second uses the AIX standard methods for installing and configuring the SNI adapters.

Configuring the SNI adapters using NIM

To configure the SNI adapters using NIM, you must update the NIM resources and define the SNI adapter's IP configuration to NIM.

Updating the NIM resources

The following procedure assumes that the LPARs are defined in the CSM server and the NIM object already created, as covered in 4.2.6, "Configuring CSM and NIM" on page 165.

1. Update the NIM Server lpp_source and spot resources:
   a. Place the required PTFs and the HPS filesets in the lpp_source directory using the update operation in NIM. For a list of SNI filesets to copy, see Table 4-4 on page 149.
   b. Update the spot image to work with the new APARs and HPS filesets. See Example 4-18.

Example 4-18 Update the lpp_source and spot resources

```
# nim -o update -a packages=all -a source=/tmp/SNI lppsource_aix52ML2
# nim -o cust -a lpp_source=lppsource_52ML2 -a installp_flags=-aXY -a fixes=update_all \
> spot52_ML2
```

Defining the SNI's adapter IP configuration in NIM

2. Create a stanza file.

   Create a stanza file, which includes the configuration information for the SNI adapters. Each stanza begins with the name of the node and is followed by a series of lines in an `attribute=value` format. The stanza file will contain a stanza for each SNI device being configured. For more details about the SNI attributes in the stanza file, refer to *Switch Network Interface for eServer pSeries High Performance Switch Guide and Reference,* SC23-4869.

   You can gather the adapter information using the **`getadapters`** command, as shown in Example 4-19.

Example 4-19 Gathering the adapter information from a node

```
# getadapters -z /tmp/adapters.c121f1rp01 -n c121f1rp01
```

Note: After the adapter information is gathered in the file, you have to add the IP address and netmask values corresponding to the SNI interfaces. In order to configure the ml0 interface, you have to create your own stanza in the definition file, as shown in Example 4-20.

Example 4-20 Example of definition file for sn# and ml0 interfaces

```
###CSM_ADAPTERS_STANZA_FILE###
c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=sn
     netaddr=20.20.20.11
     location=U1.18-P1-H1/Q3
     subnet_mask=255.255.255.0

c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=sn
     netaddr=30.30.30.11
     location=U1.18-P1-H1/Q4
     subnet_mask=255.255.255.0
```

```
c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=ml
     interface_name=ml0
     netaddr=10.10.10.11
     subnet_mask=255.255.255.0
```

3. Create a NIM adapter_def resource as shown in Example 4-21.

Example 4-21 Define the adapter_def resource in NIM

```
# nim -o define -t adapter_def -a server=master -a location=/csminstall/hps_def_adapters \
> hps_adapter_def
# lsnim -l hps_adapter_def
   class       = resources
   type        = adapter_def
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /csminstall/hps_def_adapters
   alloc_count = 0
   server      = master
```

> **Note:** The location attribute in Example 4-21 points a directory. The adapter definition files generated by the **nimadapters** command are stored in this directory.

4. Run the **nimadapters** command to import the stanza file in NIM, as shown in Example 4-22.

Example 4-22 Add the adapter definitions to the adapter_def resource

```
nimadapters -d -f /tmp/adapters.c121f1rp01 hps_adapter_def

Summary

    3  Machines will be added to the NIM environment.
```

This command parses the specified stanza file and creates the adapter definition files based on the host names existing in the input stanza file. The new definition files are added in the directory specified when defining the adapter_def resource. The files are named with the host names encountered in the stanza file. You can see an example of the adapter definition in Example 4-23, based on Example 4-20 on page 171 and Example 4-21.

Example 4-23 NIM definitions for the secondary adapter

```
# ls -l /csminstall/hps_adapter_def
c121f1rp01:
    hostname=c121f1rp01.ppd.pok.ibm.com
    machine_type=secondary
    network_type=sn
    hostaddr=9.114.66.65
    location=U1.18-P1-H1/Q3
    netaddr=20.20.20.11
    subnet_mask=255.255.255.0
c121f1rp01:
    hostname=c121f1rp01.ppd.pok.ibm.com
    machine_type=secondary
    network_type=sn
    hostaddr=9.114.66.65
    location=U1.18-P1-H1/Q4
    netaddr=30.30.30.11
    subnet_mask=255.255.255.0
c121f1rp01:
    hostname=c121f1rp01.ppd.pok.ibm.com
    machine_type=secondary
    network_type=ml
    hostaddr=9.114.66.65
    netaddr=10.10.10.11
    subnet_mask=255.255.255.0
    interface_name=ml0
```

5. Apply the SNI device drivers and patches to the node. Refer to Example 4-24.

Example 4-24 Applying the SNI filesets to nodes

```
# nim -o cust -a lpp_source=lppsource_52ML2 -a fixes=update_all -a \
> installp_flags=-aXY c121f1rp01
```

   The customize operation in Example 4-24 updates the AIX system with the latest patches from the lpp_source and installs the SNI device drivers. You need to reboot the node after installing the SNI drivers.

6. Perform the TCP/IP configuration of the SNI adapters. Refer to Example 4-25.

Example 4-25 Customize the node for configuring the SNI adapters

```
# nim -o cust -a adapter_def=hps_adapter_def c121f1rp01
```

   For the TCP/IP configuration of the SNI adapters, the /etc/firstboot script is created with the configuration commands for the sn# and ml0 interfaces.

   During the customization process of a secondary interface, the NIM methods perform specific operations on the client. In the SNI case, NIM configures the large pages parameters for the shared memory in the kernel. See “VMO configuration” on page 177. You need to reboot the node after running the customize operation.

   The /etc/firstboot is run once after reboot by the script /usr/sbin/fbcheck, from /etc/inittab. In order to be able to configure the ml0 interface, it should be run after the ml0 device is configured at boot time by the script /usr/sni/aix52/rc.ml from /etc/inittab.

Installation and configuration using AIX standard methods

You can also install the SNI adapter software and configure the SNI interfaces using standard AIX methods.

Installing the SNI drivers

Use the installp method and smit or the command line to install the filesets in AIX. The following scenario assumes that the SNI device drivers are downloaded locally, but they can be placed on a NFS file system, too. Perform the following operations on the LPAR:

1. Create a directory for storing the SNI filesets:

   ```
   mkdir /tmp/SNI
   ```

2. Copy the SNI filesets to /tmp/SNI directory. Apply here the latest patches from IBM support Web site.
3. Create the toc file if not created:

   ```
   cd /tmp/SNI
   inutoc .
   ```

4. Run **`installp -agX -d ‘.’ all`**.

5. Reboot the LPAR.
6. Verify the devices were detected by the system. Example 4-26 uses an LPAR with a single link pair attached.

Example 4-26 Listing the SNI devices in AIX

```
LPAR2# lsdev -C |grep sn
sn0          Defined                Switch Network Interface
sn1          Defined                Switch Network Interface
sni0         Available              Switch Network Interface Adapter
sni1         Available              Switch Network Interface Adapter

LPAR2# lsdev -C |grep ml
ml0          Defined                Multilink Network Interface
mlt0         Available              Multilink Communication Adapter
```

   Note that sn# and ml0 interfaces are in a Defined state, because they are not configured yet.

Subnet considerations

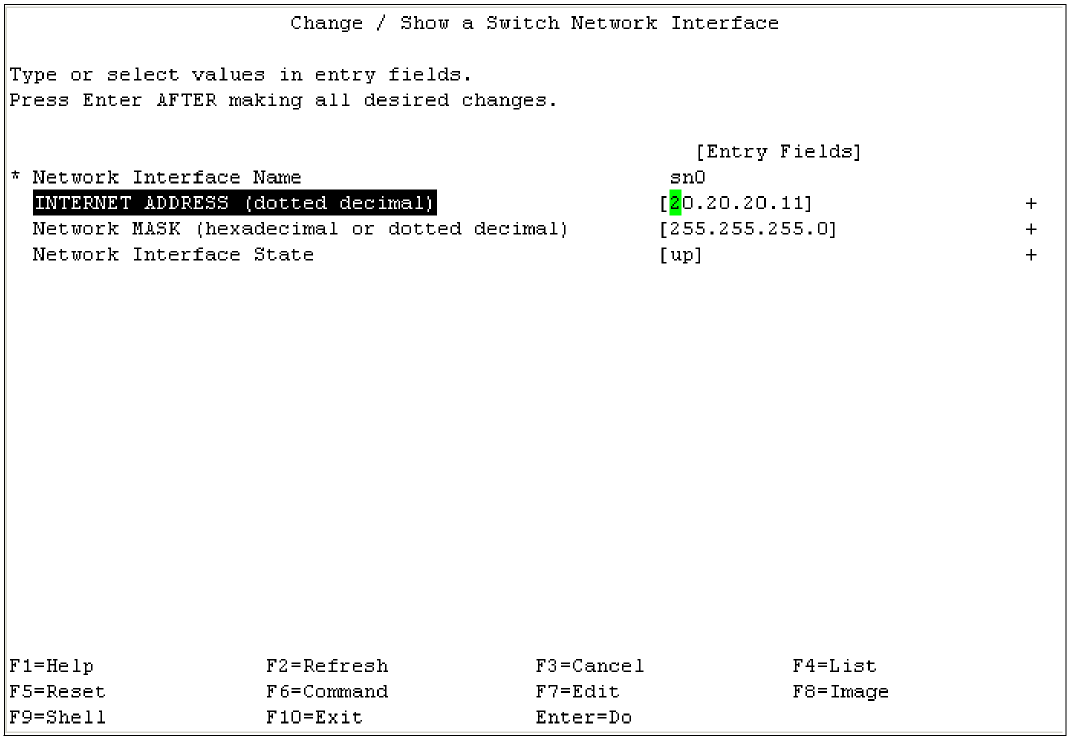
When configuring SNI and multilink interface, note that the multilink interface needs to be configured on a different subnet from any of the Switch Network Interfaces. The Switch Network Interfaces can be configured on the same subnet or on separate subnets, but they cannot be configured on the same subnet as the multilink interface.

Since an operating system instance can have multiple SNIs, it is good practice to put them on as many subnets as possible. Having as many different subnets operating system instances allows the Reliable Scalable Cluster Technology (RSCT) peer domains to more accurately detect the availability of SNIs.

Configuring the TCP/IP interfaces

Configuring the TCP/IP interfaces for the SNI is similar with the configuration of the TCP/IP over other interfaces, such as Ethernet. You can choose different ways for doing it:

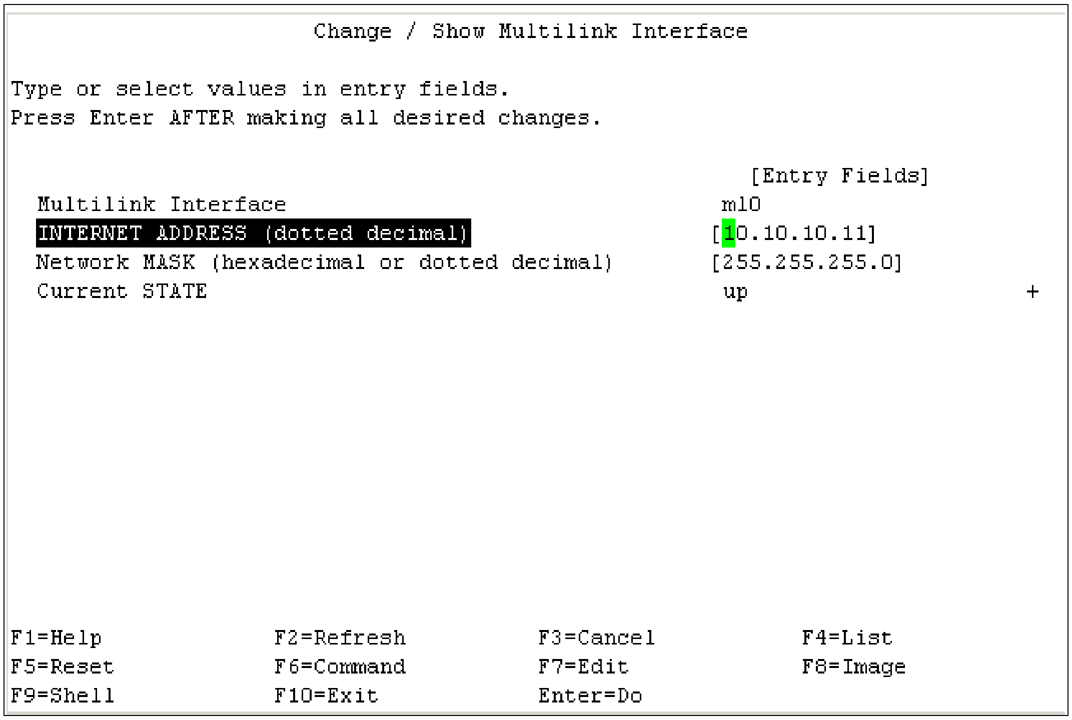
- Using smit menus:
   – `smit chsn`, for the sn interfaces (see Figure 4-11 on page 175).



```
                        Change / Show a Switch Network Interface

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

                                                        [Entry Fields]
* Network Interface Name                              sn0
  INTERNET ADDRESS (dotted decimal)                  [20.20.20.11]             +
  Network MASK (hexadecimal or dotted decimal)       [255.255.255.0]           +
  Network Interface State                            [up]                      +











F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 4-11 Configure the sn devices using smit

- `smit chml`, for the ml0 interface (see Figure 4-12)



```
                        Change / Show Multilink Interface

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

                                                        [Entry Fields]
  Multilink Interface                                 ml0
  INTERNET ADDRESS (dotted decimal)                  [10.10.10.11]
  Network MASK (hexadecimal or dotted decimal)       [255.255.255.0]
  Current STATE                                       up                       +











F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 4-12 Configure the ml0 interface in smit

- The same operations can be done from the command line, using the **chdev** command. Example 4-27 on page 176 uses a configuration with a single link pair in LPAR.

Example 4-27 Assign IP address to sn# and ml0 interfaces

```
LPAR1# chdev -l sn0 -a netaddr=20.20.20.11 -a netmask=255.255.255.0 -a state=up
sn0 changed
LPAR1# chdev -l sn1 -a netaddr=30.30.30.11 -a netmask=255.255.255.0 -a state=up
sn1 changed
LPAR1# chdev -l ml0 -a netaddr=10.10.10.11 -a netmask=255.255.255.0 -a state=up
ml0 changed
```

List the attributes of the defined interfaces. See Example 4-28.

Example 4-28 List the ODM attributes of the sn# and ml0 interfaces

```
LPAR1# lsattr -El sn0
mtu     65504         Interface MTU            True
netaddr 20.20.20.11   Network Address          True
netmask 255.255.255.0 Network Mask             True
state   up            Current Interface Status True
LPAR1# lsattr -El sn1
mtu     65504         Interface MTU            True
netaddr 30.30.30.11   Network Address          True
netmask 255.255.255.0 Network Mask             True
state   up            Current Interface Status True
LPAR1# lsattr -El ml0
netaddr 10.10.10.11   N/A True
netmask 255.255.255.0 N/A True
state   up            N/A True
```

- Using the **ifconfig** command. Refer to Example 4-29.

Example 4-29 Using ifconfig to configure the sn# and ml0 interfaces

```
LPAR1# ifconfig sn0 20.20.20.11 netmask 255.255.255.0 up
LPAR1# ifconfig sn1 30.30.30.11 netmask 255.255.255.0 up
LPAR1# ifconfig ml0 10.10.10.11 netmask 255.255.255.0 up
LPAR1# netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      0.2.55.6a.53.52       5464     0     4880     0     0
en0   1500  192.168.100 192.168.100.71        5464     0     4880     0     0
en1   1500  link#3      0.2.55.6a.4e.fc       2776     0     2561     0     0
en1   1500  172.16.1    172.16.1.100          2776     0     2561     0     0
en2   1500  link#4      0.2.55.6a.30.4a       2522     0     2510     0     0
en2   1500  172.16.2    172.16.2.100          2522     0     2510     0     0
en3   1500  link#5      0.2.55.6a.51.ef       2519     0     2509     0     0
en3   1500  172.16.3    172.16.3.100          2519     0     2509     0     0
sn0   65504 link#6                               3     0        2     0     0
sn0   65504 20.20.20    20.20.20.11              3     0        2     0     0
sn1   65504 link#7                               2     0        1     0     0
sn1   65504 30.30.30    30.30.30.11              2     0        1     0     0
ml0   65504 link#8                               0     0        0     0     0
ml0   65504 10.10.10    10.10.10.11              0     0        0     0     0
lo0   16896 link#1                            3548     0     3566     0     0
lo0   16896 127         127.0.0.1             3548     0     3566     0     0
lo0   16896 ::1                               3548     0     3566     0     0
```

Note: In order to configure the ml0 interface you must configure at least one sn# interface. Regardless the number of the sn# interfaces available on your LPAR, you have a single instance of multilink interface ml0. Refer to “AIX support for pSeries HPS” on page 148.

VMO configuration

Some kernel parameters need to be changed when using the sn# or multilink device. Use **vmo** command to request kernel large page space function. We recommend the following values:

- Large page size = 16777216
- Large page regions = 64

  Command used:

  ```
  vmo -r -o lgpg_size=16777216 -o lgpg_regions=64
  ```

  > **Note:** You need to reboot for the changes to take effect.

4.2.8 Verifying the switch connections

- For TCP/IP connectivity, you can use:
  - The **ping** command
  - Standard TCP/IP applications such as telnet, FTP, rlogin, rsh, etc.
- Check the status of the switch using the SNM menus on HMC. See 4.3, “Switch administration” on page 180.
- Check the status of the nrd daemon.

  The nrd subsystem monitors the SNI interfaces and maintains the statistics of the error events that occurred at the Switch Network Interface level. These include hardware fatal errors (mp_down), software fatal errors (sw_mp_down), and checkstops. See Example 4-30.

Example 4-30 Checking the status of the nrd subsystem

```
lpar1# lssrc -ls nrd
Subsystem         Group            PID          Status
 nrd              nrd              393448       active

 Subsystem started: Wed Nov  5 21:30:53 2003

 Number of requests since startup:
 stops -----------------> 0
 cancels ---------------> 0
 refresh ---------------> 0
 status ----------------> 5
 traceson --------------> 0
 tracesoff -------------> 0
 current trace level -> 2 (all messages, default value)
 SRC socket -------------> 0x00000005
 nrd socket -------------> 0x00000006
 nrd port ---------------> 0x0000dfdf
 status_counters.rejected ----------------> 0x00000000
 status_counters.total.handled -----------> 0x00000000
 status_counters.total.ch_fail_start -----> 0x00000000
 status_counters.total.ch_fail_complete --> 0x00000000
 status_counters.total.ch_other ----------> 0x00000000
 status_counters.total.ch_stop -----------> 0x00000000
 status_counters.total.mp_down -----------> 0x00000000
 status_counters.total.sw_mp_down --------> 0x00000000
 status_counters.total.mp_up -------------> 0x00000000
 status_counters.total.mp_dump_req -------> 0x00000000
 status_counters.total.tce_lmt -----------> 0x00000000
 status_counters.total.zc ----------------> 0x00000000
 status_counters.total.dropped -----------> 0x00000000
```

```
status_counters.total.invalid_req -------> 0x00000000

Trace file path: /var/adm/sni/nrd.trace
```

If you have specific jobs such as PE jobs or TSM backups, be sure to test this. You may need additional tuning to fully utilize the performance capabilities of the HPS.

4.2.9 Switch tuning

The following parameters of the SNI interface can be used for tuning the performance communication over the switch network. Use the **chgsni** command for changing them.

Window sizes

For SNIs, the following ODM attributes specify bounds on window sizes and memory usage:

win_poolsize	Total pinned node memory available for all user-space receive FIFOs
win_maxsize	Maximum memory used per window
win_minsize	Minimum memory used per window

Note: You do not need to reboot your machine after making changes to these attributes.

All applications on the machine can use the SNI. The following communication protocols are supported on the SNI:

- Standard TCP/IP communication through AIX sockets or message-passing libraries
- Dedicated user-space access through message-passing libraries

A distinct communication port between an application on the server and the switch is called a window. Each window has its own send FIFO and receive FIFO as well as a set of variables that describes the status of the FIFOs. The variables are used to transfer data to and from the window's FIFOs and the SNI device. The following list describes the different types of windows:

- IP: This reserved window is responsible for the IP communication among nodes.
- Service: This reserved window manages configuration and monitoring of the SNI.
- VSP: This window is reserved for VSD communication if you install VSD on your system, and you use LAPI or KLAPI.
- User Space: These windows permit high-speed data communication among user applications.

The total device memory reserved for all SNI user space windows that are used as interface network FIFO buffers is specified by the win_poolsize ODM attribute. The two other attributes, win_maxsize and win_minsize, represent the maximum and minimum memory used per window. These three new attributes are dynamically changeable. These attributes, in addition to spoolsize and rpoolsize, can be changed using the **chgsni** command. For example, to change the maximum window size to 1 MB, enter either of the commands shown in the Example 4-31.

Example 4-31 Using chgsni to change the sni ODM attributes

```
# chgsni -l sni0 -a win_maxsize=0x100000
```

or

```
# chgsni -l sni0 -a win_maxsize=1048576
```

The ODM default values, minimum values, and maximum values of the win_poolsize, win_maxsize, and win_minsize attributes are given in Table 4-5.

Table 4-5 Default, minimum and maximum values for of the attributes

Attribute	Default value	Minimum value	Maximum value
win_poolsize	128 MB/80 MB	max_windows * win_minsize	256 MB/80 MB
win_maxsize	16 MB/16 MB	256 KB	win_poolsize/16 MB
win_minsize	1 MB/1 MB	256 KB	(win_poolsize/maximu m_ windows)

Switch pool allocation

The size of the buffers allocated by the SNI device driver starts at 4096 bytes, and increases to 65536 bytes in values of multiples of 2.

If the size of the data being sent is just slightly larger than 4 KB, 8 KB, 16 KB, 32 KB, or 64 KB, for example, the buffer allocated from the pool is the next larger size up. This can cause as low as 50 percent efficiency in usage of the buffer pools because more than half of the pool can go unused in certain circumstances.

When assembling TCP/IP packets, there is always one mbuf from the IP mbuf pool used to assemble the packet header information in addition to any data buffers from the spool. If the mbuf pool size is too small, and the system runs out of mbufs, the packet is dropped. The mbuf pool is used globally for all IP traffic, and is set using the thewall tunable with the **no** command. For more information about the **no** command, see *AIX 5L Version 5.2 Commands Reference, Volume 4: n through r, SC23-4118-06*.

When you are sending 4 KB of data over the switch, an mbuf from the mbuf pool will be used, as well as one 4 KB spool buffer for the data. If the amount of data being sent is less than 200 bytes, no buffer from the spool is allocated, because there is space in the mbuf used for assembling the headers to stage the data. However, if sending 256 bytes of data, one mbuf for the IP headers and one 4 KB send pool buffer for the data will be used. In this situation, 15/16 of the buffer space in the send pool is wasted. These same scenarios apply to the receive pool when a packet is received on a node.

The key for peak efficiency of the spool and rpool buffers is to send messages that are at or just below the buffer allocation sizes, or less than 200 bytes.

Switch buffer pool allocation considerations

When tuning the rpool and spool, it is important to know the expected network traffic. If the size of the buffers for the applications is not optimum, much of the spool and rpool will be wasted. This inefficient usage requires the rpool and spool sizes to be increased. When allocating the rpool and spool, realize that this space is pinned kernel space in physical memory. This takes space away from user applications and is particularly important in small memory nodes.

If there are a small number of active sockets, there is usually enough rpool and spool space that can be allocated. A system in which a node has a large number of sockets opened across the switch can easily run out of spool space when all sockets transmit at once. For example, 300 sockets, each sending 33 KB of data, will far exceed the 16 MB limit for the spool. Or, 1100 sockets, each sending 1 KB packets, will also exceed the maximum limit. The

HPS allows you to allocate 32 MB of rpool and spool size, and will not pin the memory. You now have 32 MB physically installed in the HPS.

On the receive side of a parallel or client/server implementation, where one node acts as a collector for several other nodes, the rpool has the same problem. Four nodes, each with 600 sockets, each sending two 1 KB packets to one node, will exceed the rpool limit, but those same sockets, each sending twice as much data, 4 KB in one 4 KB packet, will work correctly. The solution is sending a single larger packet rather than several smaller ones.

4.3 Switch administration

The SNM tasks provide all the basic switch administration functions. Many of the switch functions are handled automatically, so you don't have to perform the same operations, as for the older switches. The E-commands no longer exist for the HPS.

Most of the functions are for viewing, but not editing. Command-line management can be done via an opened terminal as hscroot user. Almost all of the functions available in the command line are also available in the SNM GUI. The following list provides the operations available for the HPS:

- Power commands
  - Switch board power on/off
  - View switch board environment values
- Query commands
  - View endpoint(s) status
  - View switch network status
  - View link and device details
  - View trace information
- Diagnostic commands
  - Concurrently test active link(s) in the network (verify link)
  - Non-concurrent low-level testing of specified links (line test)
  - Non- concurrent FRU isolation procedure for a link (wrap test) (not available from a command line)
  - Gather debug information (snap) (not available from GUI)

When performing an operation from the GUI or command line, the i_stub_FS low level command is used for communicating with the **fnm** processes on HMC. Depending on the request, this command is able perform such operations as gathering data from the device database, powering the switch on/off, querying the HMCs in the FNM network, and performing diagnostics.

The following sections, details the SNM commands available for administrating the switch.

4.3.1 Power commands

Switch board power on/off (chswpower)

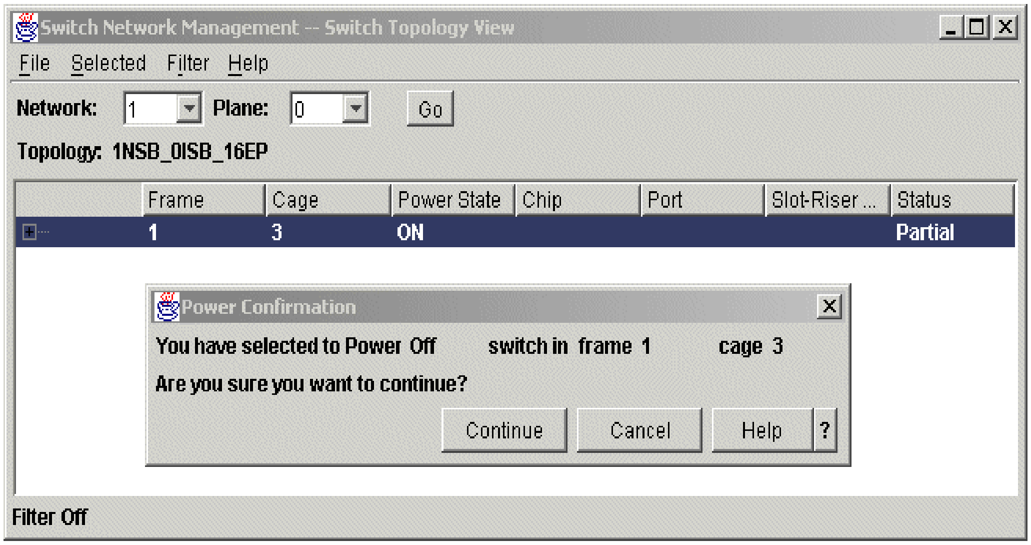
The **chswpower** command:

- Powers the switch on and off
- Issues the **i_stub_FS** command directly, to perform the operations.

You can run this command from WebSM GUI. From the Navigation area, select **Switch Management -> Switch Network Management -> Switch Topology View**.

In the new window select the switch and then select **Selected -> Power -> ON/OFF**.

Running the power off command from GUI is shown in Figure 4-13.



Network: 1 Plane: 0

Topology: 1NSB_0ISB_16EP

	Frame	Cage	Power State	Chip	Port	Slot-Riser ...	Status
⊞	1	3	ON				Partial

You have selected to Power Off switch in frame 1 cage 3

Are you sure you want to continue?

Filter Off

Figure 4-13 Switch Network Management - Switch Power Off

List switch environmentals (lsswenvir)

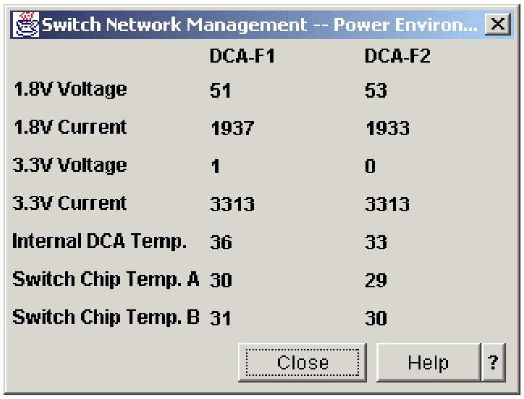
The **lsswenvir** command has the following characteristics:

- Displays switch power environmentals
- Issues the **i_stub_FS** command directly to obtain the data

The switch power environmental information, such as voltage level and temperature, are gathered from the DCAs. To perform this operation from the GUI, select **Switch Management -> Switch Network Management -> Switch Topology View**.

In the new window, select the switch and then select **Selected -> Power Environmentals**.

The output is shown in Figure 4-14.



	DCA-F1	DCA-F2
1.8V Voltage	51	53
1.8V Current	1937	1933
3.3V Voltage	1	0
3.3V Current	3313	3313
Internal DCA Temp.	36	33
Switch Chip Temp. A	30	29
Switch Chip Temp. B	31	30

Figure 4-14 Display power environmentals

4.3.2 Query commands

On the HMC, you can also use the command-line interface to check the status of the switch network. This section presents some of the available commands, with the restriction that these commands may change in future versions.

Display the end point view data (lsswendpt)

The **lsswendpt** command is used for displaying the CECs connections to the switch network. It has the following characteristics:

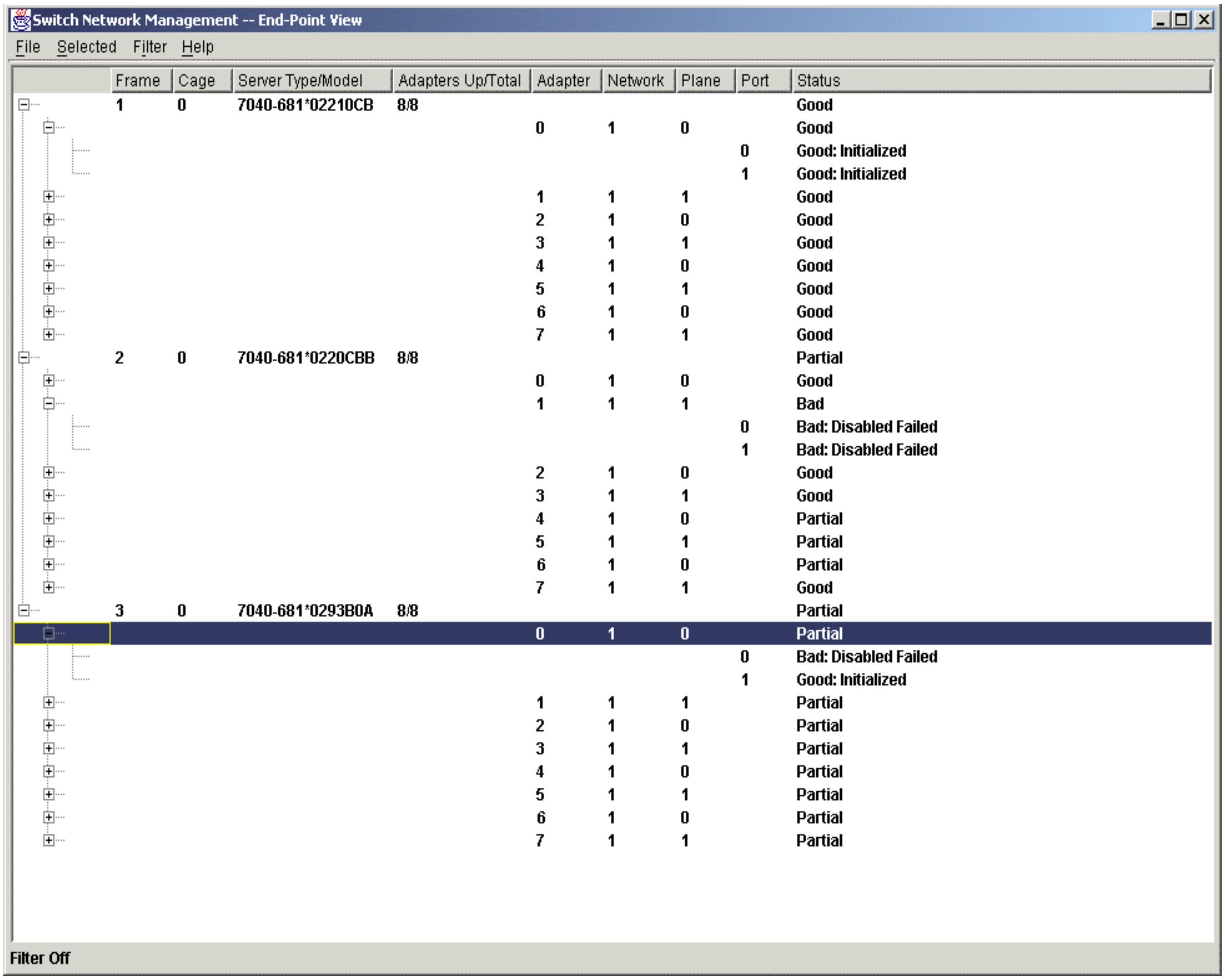
- Displays the end point view data.
- Issues **SnmInfo** command, which runs **i_stub_FS** and gets device database information.
- The lines of the output mimic the GUI panel display. See Example 4-32.
- Default is to show all data. You can select which fields to show using **-F** option.

Example 4-32 Sample list software endpoint output

```
[hscroot@c121hmc1 hscroot]$ /opt/hsc/bin/command/lsswtopol -n 1 -p 0
 frame  cage  power  chip  port  slot_riser  status
     2     3     ON                          Partial
                          0                  Partial
                                0      C10-T1  Bad: No Neighbor
                                1      C10-T2  Bad: No Neighbor
                                2       C9-T1  Bad: No Neighbor
                                3       C9-T2  Bad: No Neighbor
                                4           -  Good: Initialized
                                5           -  Good: Initialized
                                6           -  Good: Initialized
                                7           -  Good: Initialized
...
```

To access this function from the GUI interface on HMC, select **Switch Management -> Switch Network Management -> End Point view**.

You get output similar to Figure 4-15 on page 183.



Frame	Cage	Server Type/Model	Adapters Up/Total	Adapter	Network	Plane	Port	Status
1	0	7040-681*02210CB	8/8					Good
				0	1	0		Good
							0	Good: Initialized
							1	Good: Initialized
				1	1	1		Good
				2	1	0		Good
				3	1	1		Good
				4	1	0		Good
				5	1	1		Good
				6	1	0		Good
				7	1	1		Good
2	0	7040-681*0220CBB	8/8					Partial
				0	1	0		Good
				1	1	1		Bad
							0	Bad: Disabled Failed
							1	Bad: Disabled Failed
				2	1	0		Good
				3	1	1		Good
				4	1	0		Partial
				5	1	1		Partial
				6	1	0		Partial
				7	1	1		Good
3	0	7040-681*0293B0A	8/8					Partial
				0	1	0		Partial
							0	Bad: Disabled Failed
							1	Good: Initialized
				1	1	1		Partial
				2	1	0		Partial
				3	1	1		Partial
				4	1	0		Partial
				5	1	1		Partial
				6	1	0		Partial
				7	1	1		Partial

Filter Off

Figure 4-15 Switch Network Management - Endpoint view

Displays the topology view data (lsswtopol)

Use the **lsswtopol** command to display information about the switch connections. The command has the following characteristics:

- Displays switch topology view data.
- Issues **SnmInfo** command, which runs **i_stub_FS** and gets device database information.
- The lines of the output mimic the panel display (see Example 4-33).
- Default is to show all data. You can select which fields to show using **-F.**

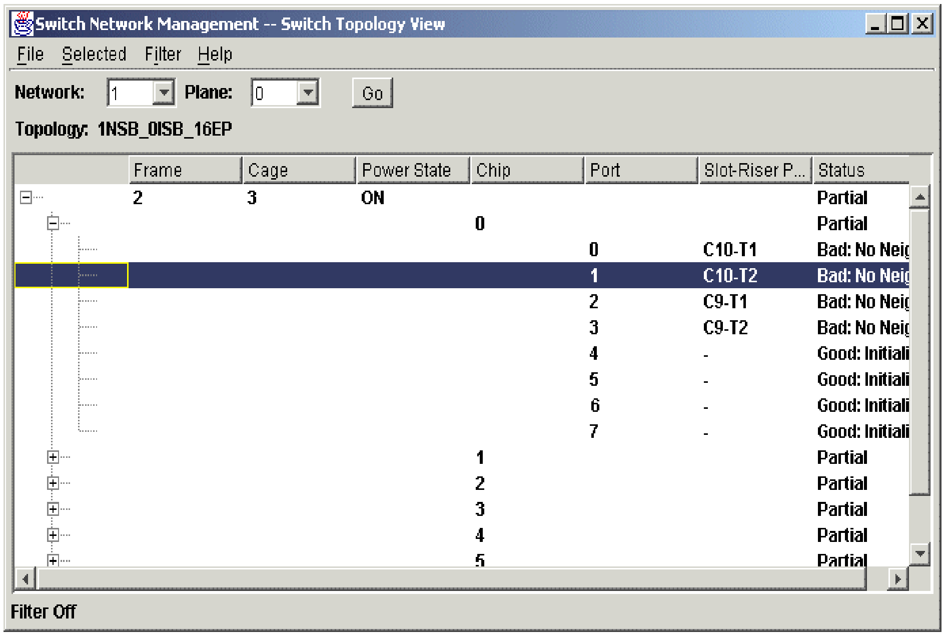
Example 4-33 List topology output

```
[hscroot@c121hmc1 hscroot]$ /opt/hsc/bin/command/lsswtopol -n 1 -p 0
 frame  cage  power  chip  port  slot_riser  status
     2     3     ON                          Partial
                        0                    Partial
                              0      C10-T1  Bad: No Neighbor
                              1      C10-T2  Bad: No Neighbor
                              2       C9-T1  Bad: No Neighbor
                              3       C9-T2  Bad: No Neighbor
                              4           -  Good: Initialized
```

```
                                5          -  Good: Initialized
                                6          -  Good: Initialized
                                7          -  Good: Initialized
.........
```

The same operation can be run from GUI. In the Navigation Area of the WebSM, select Switch **Management -> Switch Network Management -> Switch Topology View**.

The window displayed is shown in Figure 4-16.



Network: 1 Plane: 0

Topology: 1NSB_0ISB_16EP

Frame	Cage	Power State	Chip	Port	Slot-Riser P...	Status
2	3	ON				Partial
			0			Partial
				0	C10-T1	Bad: No Neig
				1	C10-T2	Bad: No Neig
				2	C9-T1	Bad: No Neig
				3	C9-T2	Bad: No Neig
				4	-	Good: Initiali
				5	-	Good: Initiali
				6	-	Good: Initiali
				7	-	Good: Initiali
			1			Partial
			2			Partial
			3			Partial
			4			Partial
			5			Partial

Filter Off

Figure 4-16 SNM switch topology view

Note: While this command shows the link status, there is no concept of host or switch responds. The link status is purely a hardware function and not a communication function. If your interface is down to AIX, your link may still be good.

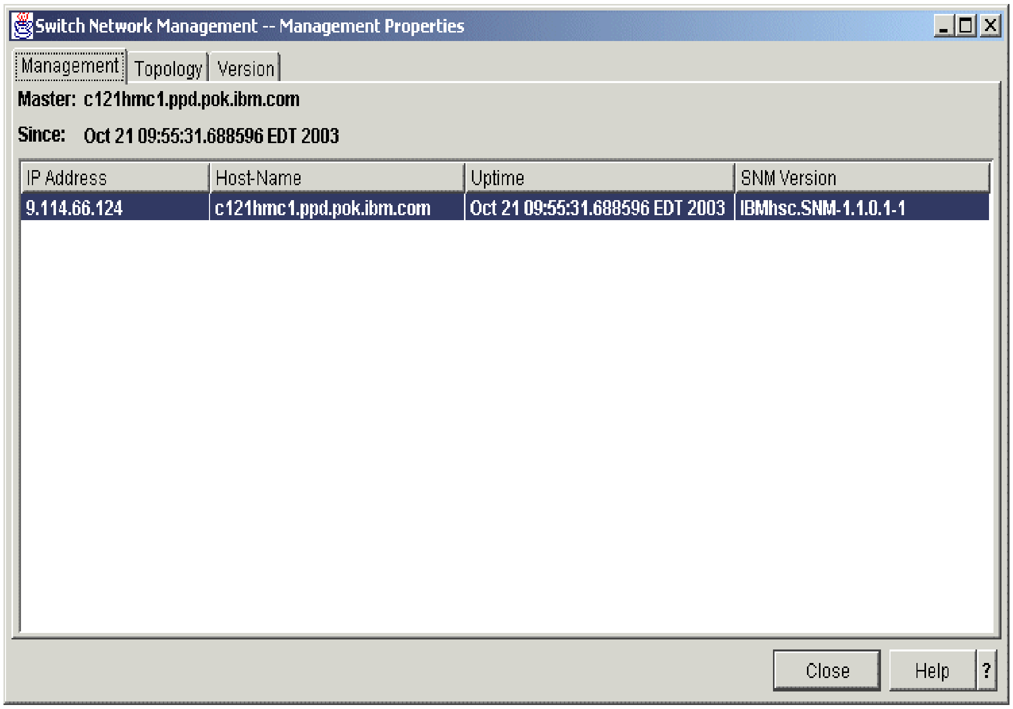
Display the management properties (`lsswmanprop`)

Use the **`lsswmanprop`** to display properties of the SNM. This command has the following characteristics:

- Displays the management properties view.
- Uses the **`-t`** flag to simulate selection of a tab on the GUI window.
- Issues the **`SnmInfo`** command, which runs **`i_stub_FS`** and gets device database information, and displays data.
- Default is to show all data. You can select which fields to show using the **`–F`** option (not supported for the **`–t ver`** option).

Display the SNM properties from WebSM GUI:

- Management view (see Figure 4-17 on page 185): Select **Switch Management -> Switch Network Management -> Management Properties -> Management** tab.



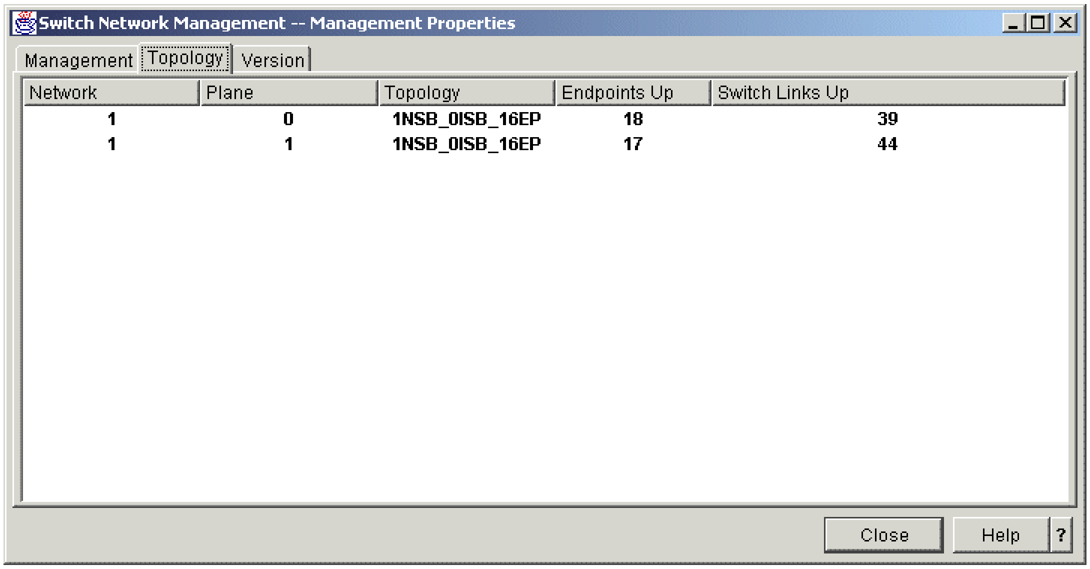
Master: c121hmc1.ppd.pok.ibm.com

Since: Oct 21 09:55:31.688596 EDT 2003

IP Address	Host-Name	Uptime	SNM Version
9.114.66.124	c121hmc1.ppd.pok.ibm.com	Oct 21 09:55:31.688596 EDT 2003	IBMhsc.SNM-1.1.0.1-1

Figure 4-17 Switch Management Properties - Management view

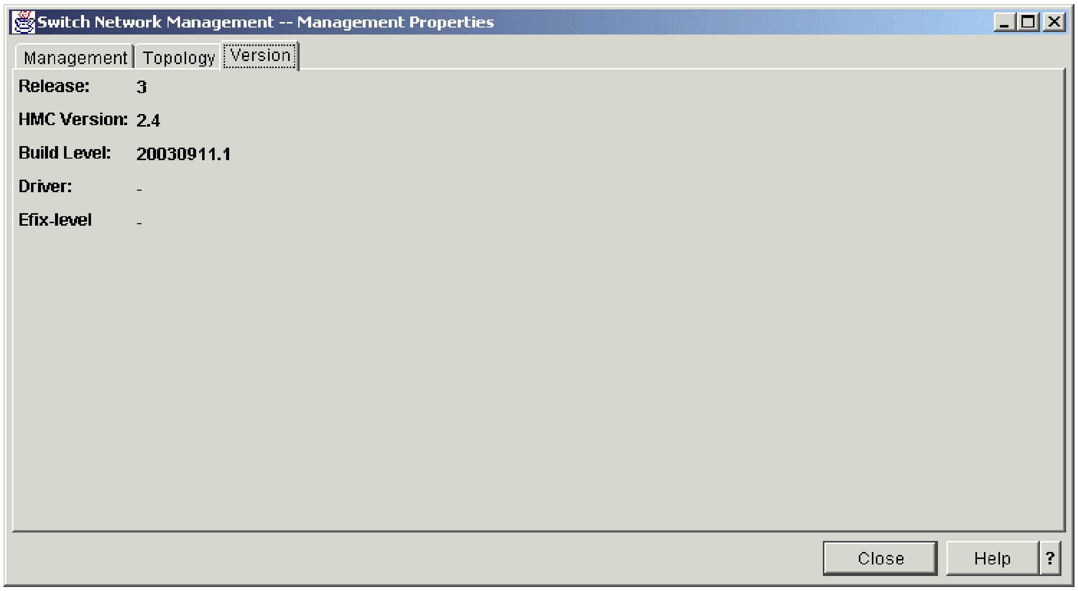
- Topology view (see Figure 4-18): Select **Switch Management -> Switch Network Management -> Management Properties -> Topology** tab.



Network	Plane	Topology	Endpoints Up	Switch Links Up
1	0	1NSB_0ISB_16EP	18	39
1	1	1NSB_0ISB_16EP	17	44

Figure 4-18 Switch Management Properties - Topology view

- Version view (see Figure 4-19 on page 186): Select **Switch Management -> Switch Network Management -> Management Properties -> Version** tab.



Release: 3

HMC Version: 2.4

Build Level: 20030911.1

Driver: -

Efix-level -

Figure 4-19 Switch Management Properties - Version View

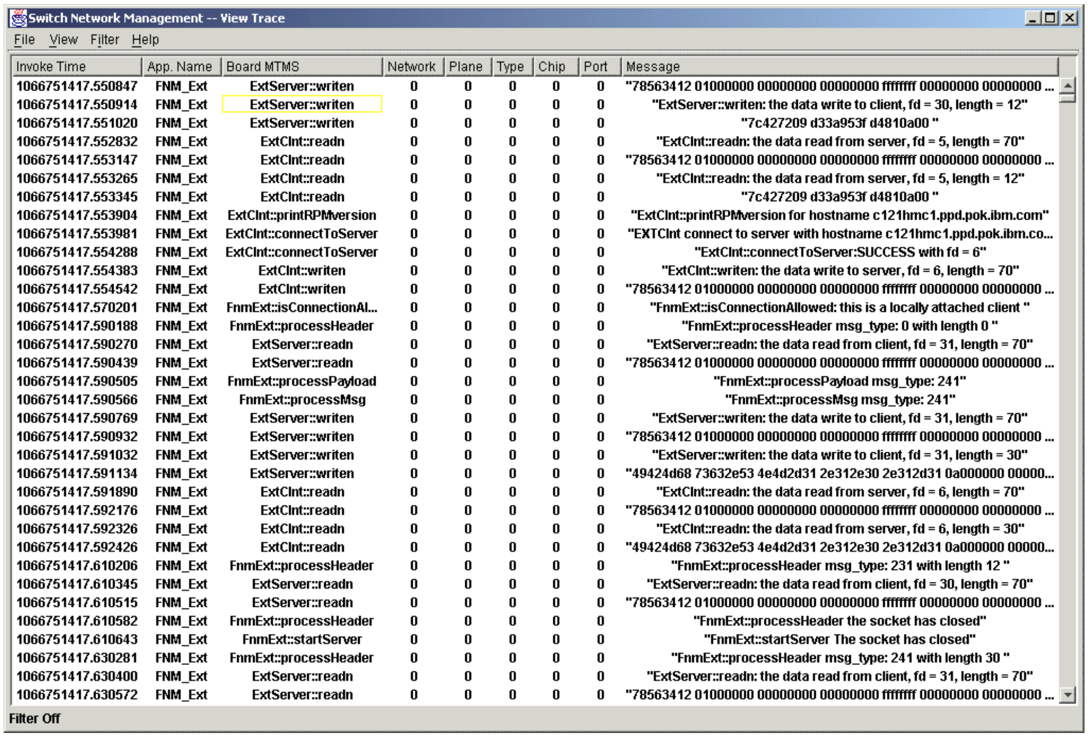
Display the SNM trace log (lsswtrace)

The SNM maintains a log of the events that occurred on the `fnm` processes. You can display the trace log of SNM using the `lsswtrace` command. It has the following characteristics:

- View trace
- Issues `SnmInfo` command, which runs i_stub_FS and gets fnmtrace.txt info (or other file info).
- When it displays the data, the lines of output mimics the GUI window display.
- Default is to show a subset of the fields. It can show all fields using the `--all` option.
- Can select individual fields using the `-F` option.

To access this function in GUI, from the Navigation area of WebSM, select **Switch Management -> Switch Network Management -> View Trace**.

The output is shown in Figure 4-20 on page 187.



Invoke Time	App. Name	Board MTMS	Network	Plane	Type	Chip	Port	Message
1066751417.550847	FNM_Ext	ExtServer::writen	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.550914	FNM_Ext	ExtServer::writen	0	0	0	0	0	"ExtServer::writen: the data write to client, fd = 30, length = 12"
1066751417.551020	FNM_Ext	ExtServer::writen	0	0	0	0	0	"7c427209 d33a953f d4810a00 "
1066751417.552832	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"ExtClnt::readn: the data read from server, fd = 5, length = 70"
1066751417.553147	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.553265	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"ExtClnt::readn: the data read from server, fd = 5, length = 12"
1066751417.553345	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"7c427209 d33a953f d4810a00 "
1066751417.553904	FNM_Ext	ExtClnt::printRPMversion	0	0	0	0	0	"ExtClnt::printRPMversion for hostname c121hmc1.ppd.pok.ibm.com"
1066751417.553981	FNM_Ext	ExtClnt::connectToServer	0	0	0	0	0	"EXTClnt connect to server with hostname c121hmc1.ppd.pok.ibm.co...
1066751417.554288	FNM_Ext	ExtClnt::connectToServer	0	0	0	0	0	"ExtClnt::connectToServer:SUCCESS with fd = 6"
1066751417.554383	FNM_Ext	ExtClnt::writen	0	0	0	0	0	"ExtClnt::writen: the data write to server, fd = 6, length = 70"
1066751417.554542	FNM_Ext	ExtClnt::writen	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.570201	FNM_Ext	FnmExt::isConnectionAl...	0	0	0	0	0	"FnmExt::isConnectionAllowed: this is a locally attached client "
1066751417.590188	FNM_Ext	FnmExt::processHeader	0	0	0	0	0	"FnmExt::processHeader msg_type: 0 with length 0 "
1066751417.590270	FNM_Ext	ExtServer::readn	0	0	0	0	0	"ExtServer::readn: the data read from client, fd = 31, length = 70"
1066751417.590439	FNM_Ext	ExtServer::readn	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.590505	FNM_Ext	FnmExt::processPayload	0	0	0	0	0	"FnmExt::processPayload msg_type: 241"
1066751417.590566	FNM_Ext	FnmExt::processMsg	0	0	0	0	0	"FnmExt::processMsg msg_type: 241"
1066751417.590769	FNM_Ext	ExtServer::writen	0	0	0	0	0	"ExtServer::writen: the data write to client, fd = 31, length = 70"
1066751417.590932	FNM_Ext	ExtServer::writen	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.591032	FNM_Ext	ExtServer::writen	0	0	0	0	0	"ExtServer::writen: the data write to client, fd = 31, length = 30"
1066751417.591134	FNM_Ext	ExtServer::writen	0	0	0	0	0	"49424d68 73632e53 4e4d2d31 2e312e30 2e312d31 0a000000 00000...
1066751417.591890	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"ExtClnt::readn: the data read from server, fd = 6, length = 70"
1066751417.592176	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.592326	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"ExtClnt::readn: the data read from server, fd = 6, length = 30"
1066751417.592426	FNM_Ext	ExtClnt::readn	0	0	0	0	0	"49424d68 73632e53 4e4d2d31 2e312e30 2e312d31 0a000000 00000...
1066751417.610206	FNM_Ext	FnmExt::processHeader	0	0	0	0	0	"FnmExt::processHeader msg_type: 231 with length 12 "
1066751417.610345	FNM_Ext	ExtServer::readn	0	0	0	0	0	"ExtServer::readn: the data read from client, fd = 30, length = 70"
1066751417.610515	FNM_Ext	ExtServer::readn	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...
1066751417.610582	FNM_Ext	FnmExt::processHeader	0	0	0	0	0	"FnmExt::processHeader the socket has closed"
1066751417.610643	FNM_Ext	FnmExt::startServer	0	0	0	0	0	"FnmExt::startServer The socket has closed"
1066751417.630281	FNM_Ext	FnmExt::processHeader	0	0	0	0	0	"FnmExt::processHeader msg_type: 241 with length 30 "
1066751417.630400	FNM_Ext	ExtServer::readn	0	0	0	0	0	"ExtServer::readn: the data read from client, fd = 31, length = 70"
1066751417.630572	FNM_Ext	ExtServer::readn	0	0	0	0	0	"78563412 01000000 00000000 00000000 ffffffff 00000000 00000000 ...

Figure 4-20 Switch Network Management - Trace view

4.3.3 Diagnostic operations

The switch diagnostic consists of testing the logical link, the physical line continuity, and a diagnostic wrap test.

Link verification (verifylink)

- Performs a quick test for the hardware components: chip, riser, adapter.
- Issues the **`i_stub_FS`** command directly to perform diagnostics.
- Can run against a single link, all links on a chip or adapter, or all links on a switch or server.
- Performs a nondisruptive test.

Test line continuity (testlinecont)

- Performs an extensive test on the switch link.
- Issues the **`i_stub_FS`** command directly to perform diagnostics.
- Can run against a single link, all links on a chip or adapter, or all links on a switch or server
- The test is disruptive, so a confirmation prompt is issued for each test. You can suppress the prompt using the **`-f`** flag when using the command line.

Wrap test

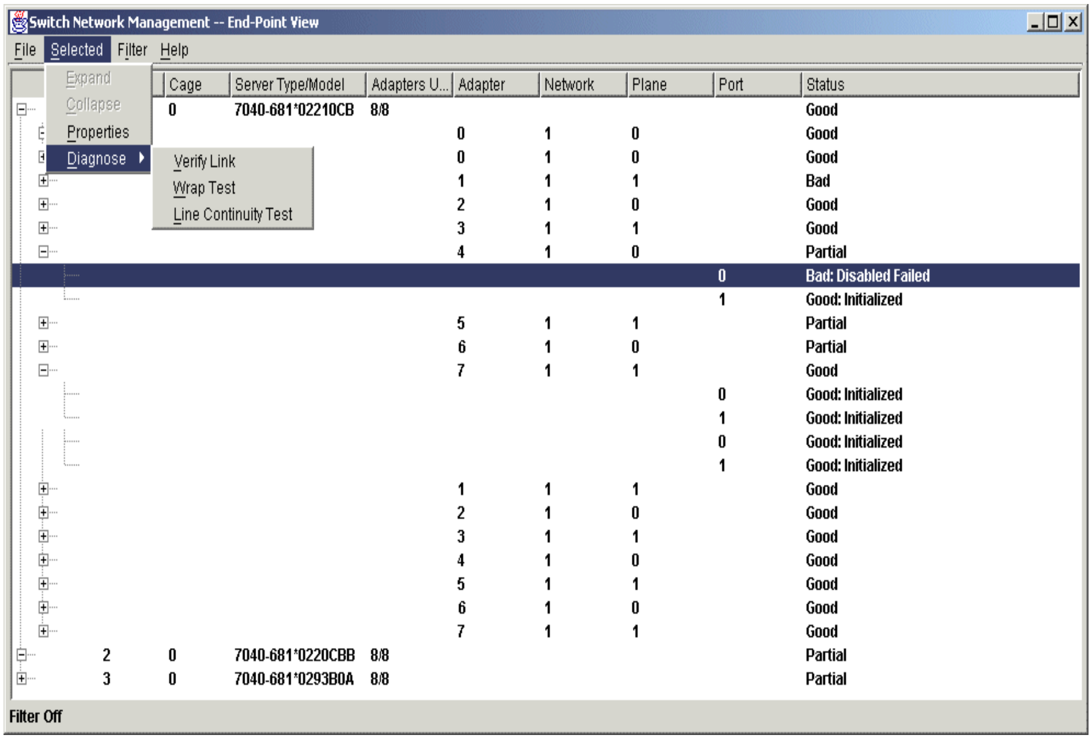
This is a user-interactive test. The user is required to plug and unplug cables and riser cards. The objective is to isolate a faulty component:

- Cable
- Switch riser card
- Adapter
- Switch planar

You can access the diagnostics function over the switch by using the HMC GUI. Select **Switch Management -> Switch Network Management -> End Point View**.

In the new window, select the target for the test (as example a link), then select **Selected -> Diagnose** and choose one of the test operations.

The diagnostics menu is shown in Figure 4-21.



	Cage	Server Type/Model	Adapters U...	Adapter	Network	Plane	Port	Status
	0	7040-681*02210CB	8/8					Good
				0	1	0		Good
				0	1	0		Good
				1	1	1		Bad
				2	1	0		Good
				3	1	1		Good
				4	1	0		Partial
							0	Bad: Disabled Failed
							1	Good: Initialized
				5	1	1		Partial
				6	1	0		Partial
				7	1	1		Good
							0	Good: Initialized
							1	Good: Initialized
							0	Good: Initialized
							1	Good: Initialized
				1	1	1		Good
				2	1	0		Good
				3	1	1		Good
				4	1	0		Good
				5	1	1		Good
				6	1	0		Good
				7	1	1		Good
2	0	7040-681*0220CBB	8/8					Partial
3	0	7040-681*0293B0A	8/8					Partial

Figure 4-21 Switch diagnostic menus

4.4 References

- *pSeries High Performance Switch Planning, Installation and Service*, GA22-7951
- *Switch Network Interface for eServer pSeries High Performance Switch Guide and Reference*, SC23-4869
- *IBM Cluster Systems Management for AIX 5L Planning and Installation Guide*, SA22-7919
- *Network Installation Management Guide and Reference*, SC23-4385

5

Clustering applications scenarios

In this chapter we provide some scenarios using the IBM eServer pSeries High Performance Switch (HPS) in real-life application environments. We present the failover test results of the following hardware and components:

- HPS logical adapters (sn0, sn1, ml0) and physical connectivity
- IBM Virtual Shared Disk
- IBM General Parallel File System as the storage-providing application for SAMBA

5.1 Testing the switch adapter failover

In this scenario, we test the logical and physical failures in our HPS environment.

5.1.1 Scenario description

The purpose of this scenario is to prove the fault tolerance of the Switch Network Interfaces. The internal construction of an SNI adapter provides high availability for the Switch Network Interface protocol devices such as *sn0*, *sn1* and *ml0*. A single cable link failure on an SNI card failure should be transparent to the operating system and the applications. The only visible effect is a delay of a few microseconds when the physical link failure occurs. All recovery is performed by the hardware, and no network packets are lost. When we change the adapter status to “down” (**`chdev -l sn0 -a state=down`**), we did not observe any effects. When we changed the status of the adapter from “up” to “detached”, we observe a temporary packet loss during the driver reconfiguration.

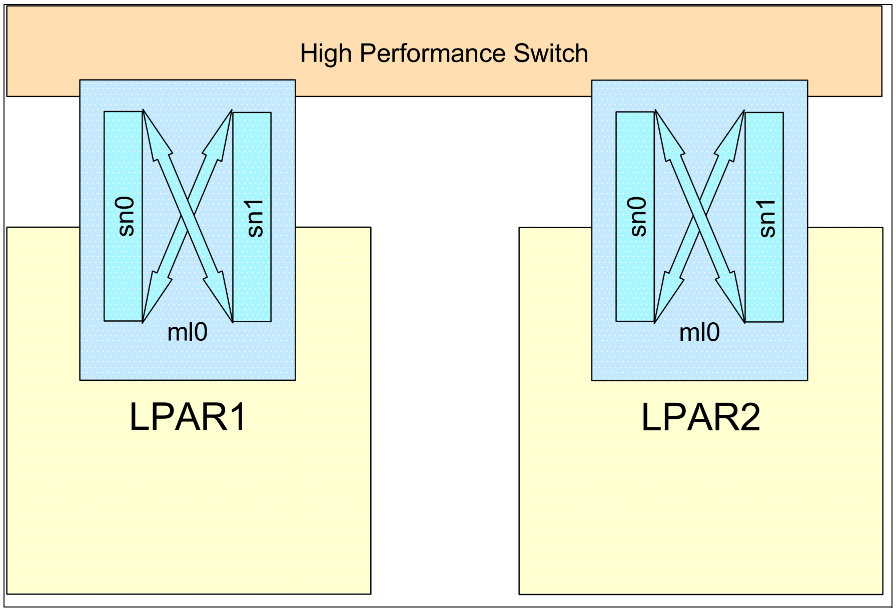


Figure 5-1 Our scenario configuration

In our scenario we use an IBM @server pSeries Model p690 with two LPARS and one IBM @server pSeries High Performance Switch (HPS). The server has two four-port SNI books but only two MCMs; therefore, only two SNI ports are assigned to each partition, as shown in Figure 5-1.

Table 5-1 shows the network configuration for the HPS SNI adapters on the LPAR1.

Table 5-1 The network configuration for p690_LPAR1

LPAR p690_LPAR1	SNI	IP address	Host name
	sn0	20.20.20.11	p690_LPAR1_sn0.itso.ibm.com
	sn1	30.30.30.11	p690_LPAR1_sn1.itso.ibm.com
	ml0	10.10.10.11	p690_LPAR1_ml0.itso.ibm.com

Table 5-2 on page 191 shows the network configuration for the HPS SNI adapters on the LPAR2.

Table 5-2 The network configuration for p690_LPAR2

LPAR p690_LPAR2	SNI	IP address	hostname
	sn0	20.20.20.12	p690_LPAR2_sn0.itso.ibm.com
	sn1	30.30.30.12	p690_LPAR2_sn1.itso.ibm.com
	ml0	10.10.10.12	p690_LPAR2_ml0.itso.ibm.com

5.1.2 Logical interface down and detach scenario

We issue a **`ping`** command from LPAR1 ml0 adapter to LPAR2 ml0 adapter. Then we use the **`chdev -l <sn> -a down`** and **`chdev -l <sn> -a detach`** commands to cause an interface failure on LPAR2 and observe the effects. Follow these steps:

1. Ping LPAR1 through each switch interface to verify communication, as shown in Example 5-1.

Example 5-1 Test output using the ping command

```
LPAR2# ping p690_LPAR1_sn0
PING p690_LPAR1_sn0.itso.ibm.com: (20.20.20.11): 56 data bytes
64 bytes from 20.20.20.11: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 20.20.20.11: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 20.20.20.11: icmp_seq=2 ttl=255 time=0 ms
^C
----p690_LPAR1_sn0.itso.ibm.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

LPAR2# ping p690_LPAR1_sn1
PING p690_LPAR1_sn1.itso.ibm.com: (30.30.30.11): 56 data bytes
64 bytes from 30.30.30.11: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 30.30.30.11: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 30.30.30.11: icmp_seq=2 ttl=255 time=0 ms
^C
----p690_LPAR1_sn1.itso.ibm.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

LPAR2# ping p690_LPAR1_ml0
PING p690_LPAR1_ml0.itso.ibm.com: (10.10.10.11): 56 data bytes
64 bytes from 10.10.10.11: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 10.10.10.11: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 10.10.10.11: icmp_seq=2 ttl=255 time=0 ms
^C
----p690_LPAR1_ml0.itso.ibm.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

2. Reset network statistics using the **`netstat`** command with parameters **`-Zc, -Zi, -Zm and -Zs`**. Verify this using the **`netstat -D`** command as shown in Example 5-2.

Example 5-2 netstat -D command output

```
LPAR2# netstat -D | egrep “sn|ml|drops”

Source                        Ipkts              Opkts     Idrops     Odrops
sn_dmx1                        2278                N/A          0        N/A
sn_dmx0                        2636                N/A          0        N/A
sn_if0                           32                 32          0          0
```

```
sn_if1                                 32                    32           0           0
ml_if0                                  0                     0           0           0
```

3. Start the **ping** test from LPAR1 to LPAR2, through sn0 as shown in Example 5-3.

Example 5-3 ping through sn0 adapter

```
LPAR1#  ping p690_LPAR2_ml0
PING p690_LPAR2_ml0.itso.ibm.com: (10.10.10.12): 56 data bytes
64 bytes from 10.10.10.12: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=2 ttl=255 time=0 ms
.
.
```

4. Bring sn0 on LPAR2 down:

```
LPAR2# chdev -l sn0 -a state=down
sn0 changed
```

5. Verify that there is no effect in ping from the **ping p690_LPAR2_ml0** command, as shown in Example 5-4.

Example 5-4 ping command verification

```
.
.
64 bytes from 10.10.10.12: icmp_seq=6 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=7 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=8 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=9 ttl=255 time=0 ms
.
.
```

This means that the network down on one of the sn<x> adapters is transparent to the ml0 multilink adapter, where <x> is the logical number of the sn interface.

5.1.3 Logical interface detach

Follow these steps:

1. Restore the status of the *sn0* adapter issuing the following command:

```
LPAR2# chdev -l sn0 -a state=up
sn0 changed
```

2. Detach the *sn0* adapter on LPAR2:

```
LPAR2# chdev -l sn0 -a state=detach
sn0 changed
```

3. Verify the ping response again as shown in Example 5-5.

Example 5-5 ping command verification

```
.
.
64 bytes from 10.10.10.12: icmp_seq=16 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=17 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=18 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=19 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=20 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=23 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=25 ttl=255 time=0 ms
```

```
64 bytes from 10.10.10.12: icmp_seq=26 ttl=255 time=0 ms
64 bytes from 10.10.10.12: icmp_seq=27 ttl=255 time=0 ms
.
.
```

When the adapter is logically detached, the device driver requires some time to reconfigure the multilink adapter.The output of the **ping** command shows a packet loss. However, verifying the otput of the **netstat -D** command, as per Example 5-6, shows no packet loss. As such, we conclude that the packet is lost in the switch buffers but outside the realm of the protocol stack. TCP should be able to detect this and retransmit any missing packets.

The traffic through the ml0 adapter stops when both the sn0 and sn1 adapters are detached, and restores immediately when either one comes back up.

Example 5-6 The netstat -D outputs

```
LPAR1 #> netstat -D | egrep "sn|ml|drops"
Source                          Ipkts                Opkts     Idrops     Odrops
sn_dmx0                          2866                  N/A          0        N/A
sn_dmx1                          3221                  N/A          0        N/A
sn_if0                           1236                 1494          0          0
sn_if1                           1763                 1662          0          0
ml_if0                              0                 1151          0          0

LPAR2 #> netstat -D | egrep "sn|ml|drops"
Source                          Ipkts                Opkts     Idrops     Odrops
sn_dmx1                          3173                  N/A          0        N/A
sn_dmx0                          3025                  N/A          0        N/A
sn_if0                             78                   77          0          0
sn_if1                            927                  940          0          0
ml_if0                              0                  489          0          0
```

5.1.4 Cable disconnected scenario

In this scenario we issue a **ping** command from LPAR1 through all switch adapters to LPAR2. We then pull out the cable and observe. These are the steps:

1. Start the **ping** command for all adapters.
2. Pull one of the cables from an SNI IO book, as shown in Figure 5-2 on page 194.

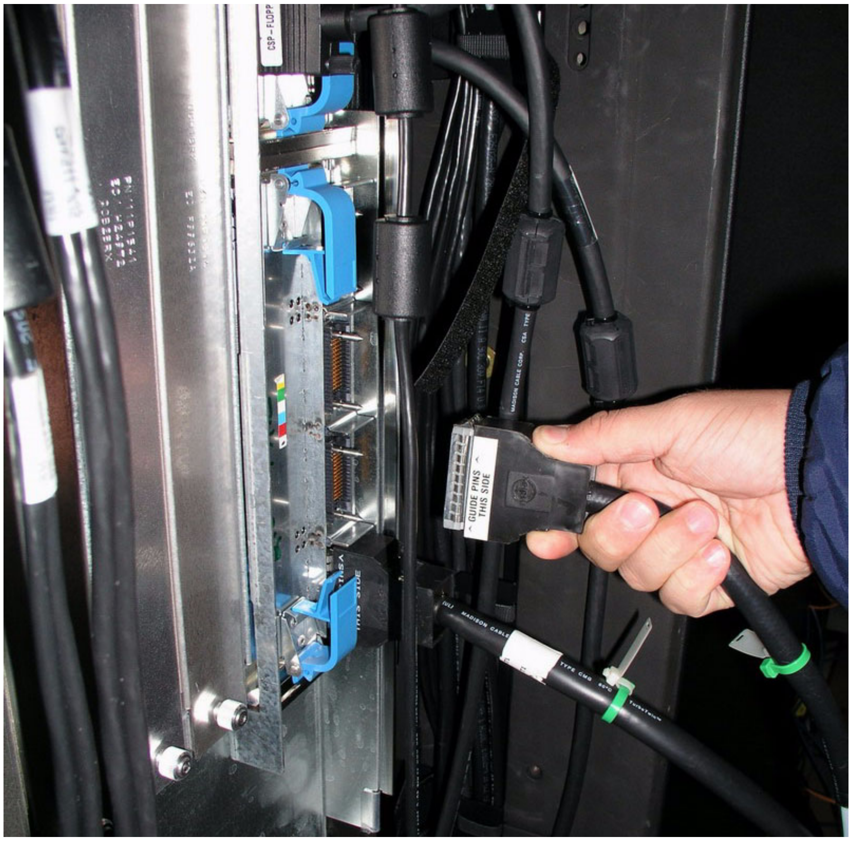


Figure 5-2 Cable fault scenario

3. We notice only a small delay in packet response, as shown in Example 5-7.

Example 5-7 Output from the ping command for the sn0 adapter

```
.
.
64 bytes from 20.20.20.12: icmp_seq=58 ttl=255 time=0 ms
64 bytes from 20.20.20.12: icmp_seq=59 ttl=255 time=0 ms
64 bytes from 20.20.20.12: icmp_seq=60 ttl=255 time=97 ms
64 bytes from 20.20.20.12: icmp_seq=61 ttl=255 time=0 ms
64 bytes from 20.20.20.12: icmp_seq=62 ttl=255 time=0 ms
.
```

Note: sn1 and ml0 did not register any delays.

5.2 VSD implementation on an HPS

In this scenario we describe the installation and configuration of a VSDs cluster using the HPS as the communication network. For more information regarding VSD concepts refer to 3.4, "IBM Virtual Shared Disks" on page 102.

5.2.1 Scenario description

The purpose of this scenario is to create a fault-tolerant VSD environment using the HPS as the communication device, as shown in Figure 5-3 on page 195. We create a pair of VSDs on two vpath devices connected to nodes LPAR1 and LPAR2. To test the fault tolerance of this solution, we generate traffic on the VSDs and halt the LPAR2.

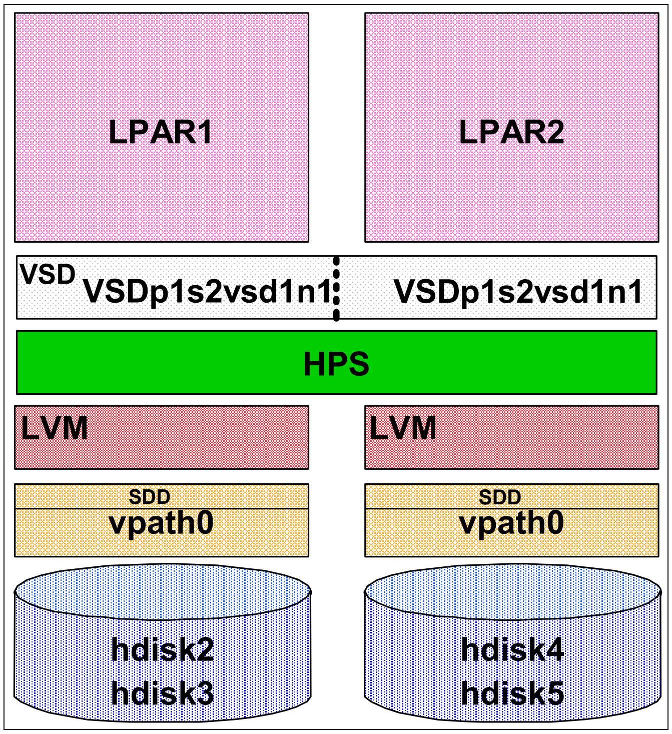


Figure 5-3 VSD scenario diagram

5.2.2 Our environment

For our tests, we have used a p690 server connected to the IBM HPS, and attached to an ESS800 storage server via a Storage Area Network (SAN).

Hardware components

For the purposes of this scenario, we use the following components:

- An IBM @server pSeries Model 690 with two LPARs
- Two 4-port SNI book, two links per LPAR
- IBM IBM @server pSeries High Performance Switch (HPS)
- Two FC Adapters for each LPAR
- McData 6064 Enterprise Fiber Director
- One IBM ESS800, two 46 GB LUNs

Software components

We use the following software components:

- AIX 5L Version 5.2 with ML2 with APAR IY47306
- IBM Virtual Shared Disk Version 4.1 (part of AIX)
- IBM GPFS Version 2.1 APAR IY36172
- IBM RSCT 2.3.1 (part of AIX)

Switch network configuration

Table 5-3 on page 196 shows the switch network configuration on LPAR1.

Table 5-3 The network configuration for p690_LPAR1

LPAR p690_LPAR1	Interface	IP address	IP label
	en0	192.168.100.71	p690_LPAR1.itso.ibm.com
	sn0	20.20.20.11	p690_LPAR1_sn0.itso.ibm.com
	sn1	30.30.30.11	p690_LPAR1_sn1.itso.ibm.com
	ml0	10.10.10.11	p690_LPAR1_ml0.itso.ibm.com

Table 5-4 shows the switch network configuration on LPAR2.

Table 5-4 The network configuration for p690_LPAR2

LPAR p690_LPAR2	Interface	IP address	IP label
	en0	192.168.100.72	p690_LPAR2.itso.ibm.com
	sn0	20.20.20.12	p690_LPAR2_sn0.itso.ibm.com
	sn1	30.30.30.12	p690_LPAR2_sn1.itso.ibm.com
	ml0	10.10.10.12	p690_LPAR2_ml0.itso.ibm.com

5.2.3 VSD installation

Example 5-8 shows the installed VSD filesets. We install these filesets using the standard **installp** command.

Example 5-8 VSD filesets installed

```
LPAR1 #> lslpp -l | grep vsd
  rsct.vsd.cmds             4.1.0.1  APPLIED    VSD Commands
  rsct.vsd.rvsd             4.1.0.1  APPLIED    Recoverable VSD
  rsct.vsd.vsdd             4.1.0.1  APPLIED    VSD Device Driver
  rsct.vsd.vsdrm            4.1.0.1  APPLIED    VSD Resource Manager
  rsct.vsd.cmds             4.1.0.0  COMMITTED  VSD Commands
  rsct.vsd.rvsd             4.1.0.1  APPLIED    Recoverable VSD
  rsct.vsd.vsdd             4.1.0.0  COMMITTED  VSD Device Driver
  rsct.vsd.vsdrm            4.1.0.0  COMMITTED  VSD Resource Manager
```

5.2.4 Creating the RSCT peer domain

To configure the nodes into an RSCT peer domain, we need to:

1. Prepare the initial security environment on each node that will be in the peer domain using the **preprpnode** command.
2. Create a new peer domain definition by issuing the **mkrpdomain** command.
3. Bring the peer domain online using the **startrpdomain** command

Preparing the nodes for the peer domain

The **preprpnode** command is used to establish the initial trust between each node that will be in the peer domain, and the node from which you will issue the **mkrpdomain** command. This command automatically exchanges public keys between nodes.

To prepare the nodes for the RSCT peer domain on the switch network, we issue the commands as specified in Example 5-9 on page 197.

Example 5-9 Node preparation

On LPAR1:

```
   LPAR1 #> preprpnode p690_LPAR1_ml0.itso.ibm.com
```

On LPAR2:

```
LPAR2 #> preprpnode p690_LPAR1_ml0.itso.ibm.com
```

Note that we used the ml0 IP label of LPAR1 for the **preprpnode** command on both nodes.

Creating a new peer domain

The **mkrpdomain** command creates a new peer domain definition. A peer domain definition consists of:

- A peer domain name
- The list of nodes included in that peer domain
- The UDP port numbers to be used by Topology Services and Group Services inter-node communication

To create the new peer domain, we use the command **mkrpdomain**, as shown in Example 5-10.

Example 5-10 Peer domain creation

```
LPAR1 #> mkrpdomain VSD_domain p690_LPAR1_sn0.itso.ibm.com p690_LPAR2_sn0.itso.ibm.com
```

Starting the peer domain

We start the peer domain using the **startrpdomain** command as specified in the Example 5-11.

Example 5-11 Starting the peer domain

```
LPAR1 #> startrpdomain -VT VSD_domain
/usr/sbin/rsct/bin/ctdspmsg startrpdomain configrmcli.cat IMsgstartrpdomainStart VSD_domain
Starting the peer domain "VSD_domain".
startrpdomain: calling lsrsrc-api
startrpdomain: lsrsrci-api returned 0
startrpdomain: calling startrsrc-api
startrsrc-api results:
0x201a 0xffff 0xc451f567 0xeaa8a392 0x0ed0b21b 0xf1db6ec8
startrpdomain: startrsrc-api returned 0
/usr/sbin/rsct/bin/ctdspmsg startrpdomain configrmcli.cat IMsgstartrpdomainEnd VSD_domain
Completed starting the peer domain "VSD_domain".
```

Verifying the status of the peer domain

We verify the status of the peer domain using the **lsrpdomain** and **lsrpnode** commands as shown in Example 5-12.

Example 5-12 Peer domain status verification

```
LPAR1 #> lsrpdomain
Name       OpState RSCTActiveVersion MixedVersions TSPort GSPort
VSD_domain Online  2.3.1.0           No            12347  12348
LPAR1 #> lsrpnode
Name           OpState RSCTVersion
p690_LPAR1_ml0 Online  2.3.1.0
p690_LPAR2_ml0 Online  2.3.1.0
```

5.2.5 Designating the VSD nodes

Prior to designating the VSD nodes, we configured the following parameters for the *ml0* interface:

- The minimum buddy buffer size is 4096
- The maximum buddy buffer size is 262144
- The maximum number of buddy buffers is 4000
- The maximum IP message size is 61440

For more information on setting initial tuning values, refer to 3.4.14, “HPS considerations when using the IP protocol for data transmissions” on page 111.

To designate the VSD nodes, we used the **vsdnode** command, as shown in Example 5-13.

Example 5-13 vsdnode command

```
LPAR1 #> vsdnode 1 2 ml0 4096 262144 4000 61440
0x202d 0xffff 0xc451f567 0xeaa8a392 0x0ed0b3bd 0x7bf9a1bd
0x202d 0xffff 0xc451f567 0xeaa8a392 0x0ed0b3bd 0xc116ab17
vsdnode: Nodes 1,2 have been designated as vsd nodes.
```

Tip: You may use the **vsdnode** command from any node that is online in the RSCT peer domain, and the operational parameters you are setting can be applied to one or more nodes in the peer domain. If the operational parameters should be the same for all nodes, you need to run the **vsdnode** command only once. If the operational parameters should differ between nodes, you need to run this command once for each set of nodes that have similar Virtual Shared Disk parameters.

5.2.6 Creating the VSD

The **createvsd** command in Example 5-14 creates the VSDp1s2vsd1n1 shared drive on the nodes. This command automatically creates the volume group and the logical volume on LPAR1, and then imports it on LPAR2.

Example 5-14 Create the VSDp1s2vsd1n1 VSD

```
LPAR1 #> createvsd -n 1/2:vpath0/ -s 5000 -g VSDp1s2vg -v VSDp1s2vsd -l VSDp1s2lv
createvsd: calls Getopts.
createvsd: parsing node_list.
createvsd: creates task tables.
createvsd: calls checkclvm.perl on the nodes p690_LPAR1_ml0.itso.ibm.com
createvsd: calls domkvglv.perl.
OK:1:mkvg4vp -f -y VSDp1s2vg -s 4 vpath0
OK:1:mklv -a c -y VSDp1s2lv1n1 -e x VSDp1s2vg 1250 vpath0
It took about 2 seconds in mkvglv.
createvsd: calls dovaryoffvg.perl VSDp1s2vg on the primary node p690_LPAR1.itso.ibm.com
OK:1:chvg -a n VSDp1s2vg
OK:1:varyoffvg VSDp1s2vg
createvsd: calls doimportvg.perl VSDp1s2vg on the nodes p690_LPAR2.itso.ibm.com with
0022be2a31fa63ca
importvg : VSDp1s2vg
importvg : OK:2:importvg -y VSDp1s2vg hdisk4
importvg : OK:2:chvg -a n VSDp1s2vg
importvg : timestamp 2 VSDp1s2vg 3fa18fa606222c22
importvg : OK:2:dpovgfix VSDp1s2vg success
importvg : OK:2:varyoffvg VSDp1s2vg
importvg : It took about 3 seconds.
It took about 3 seconds in importvg.
```

```
createvsd: calls vsdvg.
OK:1:vsdvg -g VSDp1s2vgn1b2 VSDp1s2vg 1 2
It took about 4 seconds in vsdvg.
createvsd: calls dovaryonvg.perl VSDp1s2vg on pri nodes p690_LPAR1.itso.ibm.com
OK:1:varyonvg  VSDp1s2vg
createvsd: calls defvsd.
OK:1:defvsd VSDp1s2lv1n1 VSDp1s2vgn1b2 VSDp1s2vsd1n1
It took about 2 seconds in defvsd.
```

The **createvsd** command in Example 5-15 creates the VSDp2s1vsd1n2 shared drive on LPAR2 and imports it on LPAR1.

Example 5-15 Create the VSDp2s1vsd1n2 VSD

```
LPAR1 #> createvsd -n 2/1:vpath1/ -s 5000 -g VSDp2s1vg -v VSDp2s1vsd -l VSDp2s1lv
createvsd: calls Getopts.
createvsd: parsing node_list.
createvsd: creates task tables.
createvsd: calls checkclvm.perl on the nodes p690_LPAR2_ml0.itso.ibm.com
createvsd: calls domkvglv.perl.
OK:2:mkvg4vp -f -y VSDp2s1vg -s 4 vpath1
OK:2:mklv -a c -y VSDp2s1lv1n2 -e x VSDp2s1vg 1250 vpath1
It took about 1 seconds in mkvglv.
createvsd: calls dovaryoffvg.perl VSDp2s1vg on the primary node p690_LPAR2.itso.ibm.com
OK:2:chvg -a n VSDp2s1vg
OK:2:varyoffvg VSDp2s1vg
createvsd: calls doimportvg.perl VSDp2s1vg on the nodes p690_LPAR1.itso.ibm.com with
0022be2a31fa6653
importvg : VSDp2s1vg
importvg : OK:1:importvg -y VSDp2s1vg vpath1
importvg : OK:1:chvg -a n VSDp2s1vg
importvg : timestamp 1 VSDp2s1vg 3fa195be1af18086
importvg : OK:1:dpovgfix VSDp2s1vg success
importvg : OK:1:varyoffvg VSDp2s1vg
importvg : It took about 1 seconds.
It took about 2 seconds in importvg.
createvsd: calls vsdvg.
OK:1:vsdvg -g VSDp2s1vgn2b1 VSDp2s1vg 2 1
It took about 5 seconds in vsdvg.
createvsd: calls dovaryonvg.perl VSDp2s1vg on pri nodes p690_LPAR2.itso.ibm.com
OK:2:varyonvg  VSDp2s1vg
createvsd: calls defvsd.
OK:1:defvsd VSDp2s1lv1n2 VSDp2s1vgn2b1 VSDp2s1vsd1n2
It took about 2 seconds in defvsd.
```

We specified the *vpath* devices as the local disk drives. The **createvsd** recognizes this and calls the **mkvg4vp**, which removes the PVID from the related hdisks.

Refer to Table 5-5 for the detailed VSD configuration.

Table 5-5 The VSD structure on the nodes

VSD name	Primary Node	Secondary Node	Global VG name	Local VG name	LV Name
VSDp1s2vsd1n1	LPAR1	LPAR2	VSDp1s2vgn1b2	VSDp1s2vg	VSDp1s2lv1n1

VSD name	Primary Node	Secondary Node	Global VG name	Local VG name	LV Name
VSDp2s1vsd1n2	LPAR2	LPAR1	VSDp2s1vgn2b1	VSDp2s1vg	VSDp2s1lv1n2

Important: Since secure shell (ssh) is not curently supported for VDS configuration, the /.rhosts file must be created on each node, containing all IP labels to be used for VSD communication in the VSD cluster.

Setting the restricted level of VSD software

In order for the VSD to work correctly, we must restrict the version of the VSD level for all nodes to Version 4.1 by issuing the command as specified in Example 5-16.

Example 5-16 Restrict VSD level

```
LPAR2 #> rvsdrestrict -s RVSD4.1
rvsdrestrict level is RVSD4.1
```

Activating the VSD

The VSD can be started by issuing the **ha_vsd reset** command on each node as specified in Example 5-17.

Example 5-17 ha_vsd reset command on the nodes

```
LPAR1 #> ha_vsd reset
Stopping RVSD subsystems.
0513-004 The Subsystem or Group, rvsd, is currently inoperative.
/opt/rsct/vsd/bin/cfgvsd -a
ls: 0653-341 The file /dev/VSD0 does not exist.
10/30/03 18:45:06 cfgvsd:  cfgvsd:LoadDD: successful
Starting RVSD recovery daemon.
ha.vsd: 2506-111 Thu Oct 30 18:45:07 EST 2003 The rvsdrestrict command forces RVSD to
reduce its function to 4.1.0.0.
0513-059 The rvsd Subsystem has been started. Subsystem PID is 114768.

LPAR2 #> ha_vsd reset
Stopping RVSD subsystems.
0513-004 The Subsystem or Group, rvsd, is currently inoperative.
ls: 0653-341 The file /dev/VSD0 does not exist.
10/30/03 18:45:06 cfgvsd:  cfgvsd:LoadDD: successful
/opt/rsct/vsd/bin/cfgvsd -a
Starting RVSD recovery daemon.
ha.vsd: 2506-111 Thu Oct 30 18:42:12 EST 2003 The rvsdrestrict command forces RVSD to
reduce its function to 4.1.0.0.
0513-059 The rvsd Subsystem has been started. Subsystem PID is 700582.
```

5.2.7 VSD verification

The VSD status verification

Verify the running VSD cluster using VSD and standard system commands by following these steps:

1. Verify the VSD data information using the **vsdatalst** command, as shown in Example 5-18.

Example 5-18 The vsdatalst command

```
LPAR1 #> vsdatalst  -v
              VSD Table
VSD name                        logical volume  Global Volume Group            minor# size_in_MB
------------------------------- --------------- ------------------------------ ------ ----------
VSDp1s2vsd1n1                   VSDp1s2lv1n1    VSDp1s2vgn1b2                       1 5000
VSDp2s1vsd1n2                   VSDp2s1lv1n2    VSDp2s1vgn2b1                       2 5000

LPAR1 #> vsdatalst  -n
       VSD Node Information
                                               Buddy Buffer
  node                 VSD      IP packet minimum maximum # maxbufs
number host_name       adapter    size     size    size
------ --------------- -------- --------- ------- ------- ---------
     1 p690_LPAR1_ml0. ml0          61440    4096  262144    4000
     2 p690_LPAR2_ml0. ml0          61440    4096  262144    4000

LPAR1 #> vsdatalst  -g
      VSD Global Volume Group Information
                                                Server Node Numbers
Global Volume Group name        Local VG name   primary      backup eio_recovery     recovery
server_list                          vsd_type
------------------------------- --------------- -------      ------ ------------     --------
-----------------------------------  --------
VSDp1s2vgn1b2              VSDp1s2vg         1       2            1       0    0
VSD
VSDp2s1vgn2b1              VSDp2s1vg         2       1            1       0    0
VSD
```

2. Verify the VSD status with the **lsvsd -l** command on one of the nodes as shown in Example 5-19. The state column lists the current status of the shared drives.

Example 5-19 lsvsd -l

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name        size(MB) server_list
 1      ACT    1      39        1      VSDp1s2vsd1n1     5000         1
 2      ACT    2       0        0      VSDp2s1vsd1n2     5000         2

```

3. Check the volume group status on the nodes using the **lspv** command on each node, as shown in Example 5-20.

Example 5-20 Volume group state on the nodes

```
LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
vpath1          0022be2a31fa6653                    VSDp2s1vg

LPAR2 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg
vpath1          0022be2a31fa6653                    VSDp2s1vg       active
```

Since we are not using concurrent VSD, the node LPAR1 is the primary server for the VSDp1s2vsd1n1 VSD and the related volume group is varied on this node. The VSDp2s1vg volume group is active on LPAR2, since LPAR2 is the primary server for it. This is a two-node non-concurrent VSD configuration with mutual takeover.

VSD write and read tests

In Example 5-21, on LPAR1 we write the /foo file to the VSDp2s1vsd1n2 VSD, and then read the contents of this VSD to the /bar file. The traffic passes through the HPS and is being stored on the vpath connected to LPAR2. We calculate the checksum of the source and destination files using the **sum** command to make sure they are exactly the same.

Example 5-21 VSD write and read test

```
LPAR1 #> sum /foo
25995     1 /foo

LPAR1 #> dd if=/foo of=rVSDp2s1vsd1n2 bs=1024
0+1 records in.
0+1 records out.

LPAR1 #> dd if=VSDp2s1vsd1n2 of=/bar bs=1024
0+1 records in.
0+1 records out.

LPAR1 #> sum /bar
25995     1 /bar
```

5.2.8 VSD failover

In this scenario, we test the failover and the recovery of the VSDs by simulating a node failure. We test the VSD accessibility by writing a file from LPAR1, through the switch to a VSD attached to LPAR2. This scenario utilizes the single-node quorum without the Control Workstation feature in VSD 4.1.

1. Verify status of the VSDs

   Verify the status of the VSDs by using the **lsvsd -l** and **lspv** commands. Example 5-22 shows the configuration on LPAR1 and Example 5-23 shows the configuration on LPAR2.

Example 5-22 VSD status on LPAR1

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name         size(MB) server_list
 1      ACT    1      39       1      VSDp1s2vsd1n1    5000        1
 2      ACT    2       0       0      VSDp2s1vsd1n2    5000        2
LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
vpath1          0022be2a31fa6653                    VSDp2s1vg
```

Example 5-23 VSD status on LPAR2

```
LPAR2 lsvsd -l
minor  state server lv_major lv_minor vsd-name         size(MB) server_list
 1      ACT    1       0       0      VSDp1s2vsd1n1    5000        1
 2      ACT    2      40       1      VSDp2s1vsd1n2    5000        2

LPAR2 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg
vpath1          0022be2a31fa6653                    VSDp2s1vg       active
```

2. Write a large file to a VSD

   We use the *foo* file on the node LPAR1 to transfer data to the VSD VSDp2s1vsd1n2, which is attached to the node LPAR2. See Example 5-24. For this purpose, we use the **backup** command. The traffic is sent through the switch.

Example 5-24 File transfer to VSDp2s1vsd1n2 vsd

```
LPAR1 #> ls -l .
total 245128
-rw-r--r--   1 root     system    125505536 Oct 31 13:32 foo
LPAR1 #> echo “foo” | backup -iqvf /dev/VSDp2s1vsd1n2
Backing up to /dev/VSDp2s1vsd1n2.
Cluster 262144 bytes (512 blocks).
Volume 1 on /dev/VSDp2s1vsd1n2
```

3. Halt the node LPAR2

   We use the **fasthalt -q** command to halt the node LPAR2 as shown in Example 5-25.

Example 5-25 Halt the LPAR2 node

```
LPAR2 #> sync;sync;sync;fasthalt -q
....Halt completed....
```

4. Verify the failover of the VSD

   In this step we verify the failover of the VSDs from LPAR2 to LPAR1. For this purpose we use the **lsvsd -l** and **lspv** commands on LPAR1. We notice the status change of the VSDp2s1vsd1n2 VSD from ACT(ive) to SUS(pended). For about a minute the **backup** command seems to freeze, and we cannot see any traffic on the VSDs. We verify this using the **lsvsd -s** command. See Example 5-26.

Example 5-26 VSD failover verification

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                           size(MB)
server_list
 1      ACT    1      39       1      VSDp1s2vsd1n1                        5000         1
 2      SUS   -1       0       0      VSDp2s1vsd1n2                        5000
LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
vpath1          0022be2a31fa6653                    VSDp2s1vg

LPAR1 #> lsvsd -s
 lc-rd  lc-wt  rm-rd  rm-wt   c-rd   c-wt   p-rd   p-wt         br        bw vsd-name
     0      0      0      0      0      0      0      0          0         0 VSDp1s2vsd1n1
     0 120369      0      0      0 120370      0      0          0    240738 VSDp2s1vsd1n2
```

After about a minute, we run **lsvsd -l** and **lspv** again on the node LPAR1. The volume group VSDp2s1vg is varied on the node LPAR1 and the VSDp2s1vsd1n2 is in active state again. The **lsvsd -s** command confirms the traffic on this VSD as shown in Example 5-27.

Example 5-27 VSD status - second verification

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                           size(MB)
server_list
 1      ACT    1      39       1      VSDp1s2vsd1n1                        5000         1
 2      ACT    1      40       1      VSDp2s1vsd1n2                        5000         1

LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
vpath1          0022be2a31fa6653                    VSDp2s1vg       active

LPAR1 #> lsvsd -s
 lc-rd  lc-wt  rm-rd  rm-wt   c-rd   c-wt   p-rd   p-wt         br        bw vsd-name
     0      0      0      0      0      0      0      0          0         0 VSDp1s2vsd1n1
     0  122564         0         0         0  120370         0    2195            0      245128  VSDp2s1vsd1n2
```

5. Bring the node LPAR2 up

   We activate LPAR2 on the HMC, as shown in Example 5-4 on page 204.

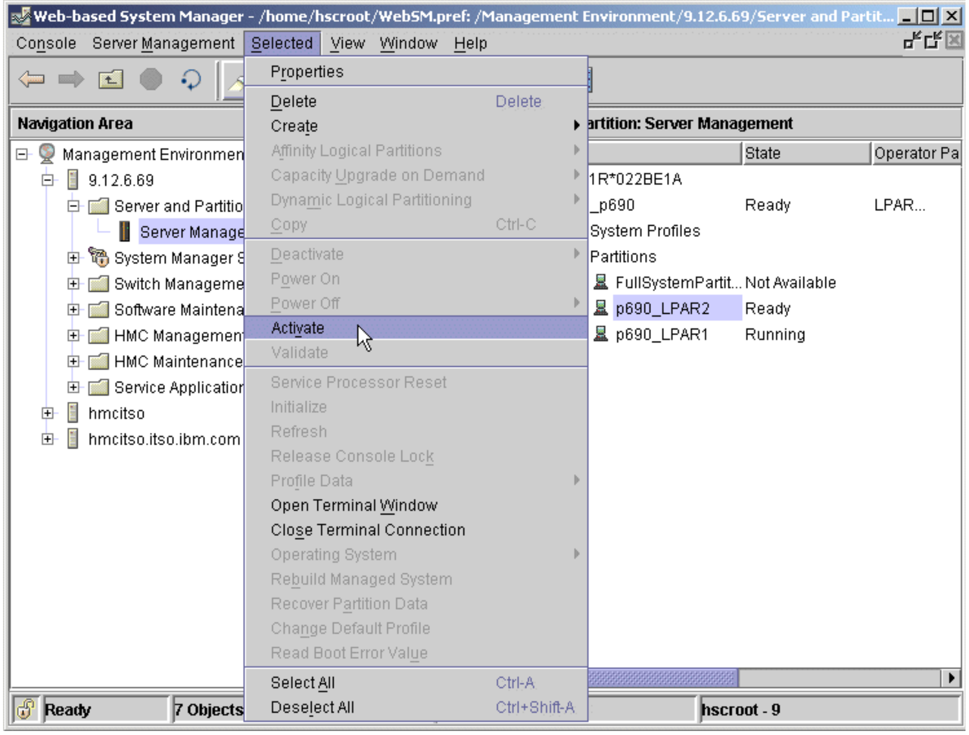


Figure 5-4 LPAR2 activation

6. Verify the reintegration of LPAR2

   When the node is started, we verify the status of the VSD cluster using the **lsvsd** and **lspv** commands on both LPAR1 and LPAR2 nodes as shown in Example 5-28.

   When the node LPAR2 starts, it reintegrates with the cluster automatically, and takes over the resources for which it is the primary node to.

Example 5-28 LPAR2 reintegration verification

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                          size(MB)
server_list
 1      ACT     1      39        1      VSDp1s2vsd1n1                     5000          1
 2      ACT     2       0        0      VSDp2s1vsd1n2                     5000          2

LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
vpath1          0022be2a31fa6653                    VSDp2s1vg

LPAR2 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                          size(MB)
server_list
 1      ACT     1       0        0      VSDp1s2vsd1n1                     5000          1
 2      ACT     2      40        1      VSDp2s1vsd1n2                     5000          2

LPAR2 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg
```

```
vpath1          0022be2a31fa6653                    VSDp2s1vg       active
```

7. Data consistency verification

   To verify the data written to VSDp2s1vsd1n2 VSD is consistent after the failure of the node LPAR2 and no information is lost, we use the **restore** command. See Example 5-29.

Example 5-29 Data consistency verification

```
LPAR1 #> restore -Tqvf /dev/VSDp2s1vsd1n2
New volume on /dev/VSDp2s1vsd1n2:
Cluster size is 262144 bytes (512 blocks).
The volume number is 1.
The backup date is: Fri Oct 31 15:55:18 EST 2003
Files are backed up by name.
The user is root.
   125505536 foo
The total size is 125505536 bytes.
The number of archived files is 1.
```

5.3 GPFS on VSD

The purpose of this scenario is to create the GPFS environment running on two server nodes on top of the VSDs previously created in 5.2, "VSD implementation on an HPS" on page 194. Then we create the SAMBA network share from one of the GPFS file systems, and test the GPFS failover by halting the server nodes. The diagram of this scenario is shown in Figure 5-5.

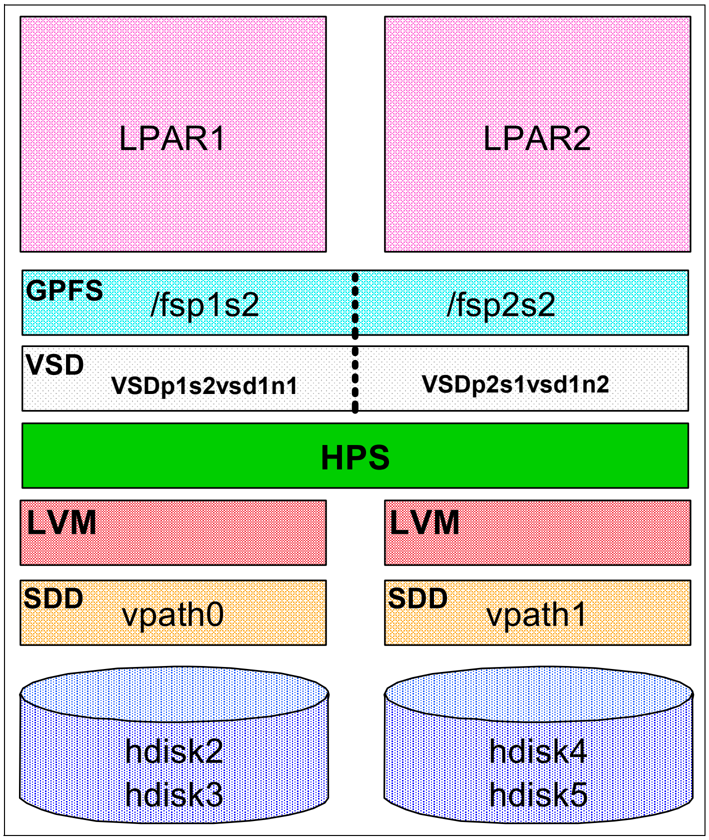


Figure 5-5 The GPFS scenario diagram

5.3.1 Our environment

For this scenario, we use the following hardware and software components.

Hardware components

- An IBM eServer pSeries Model 690 with two LPARs
- Two 4-port SNI books, two links are available per LPAR
- IBM eServer pSeries High Performance Switch (HPS)
- IBM ESS, two LUNs connected to both LPARs via FC

Software components

- AIX 5L Version 5.2 with ML2
- IBM Virtual Shared Disk Version 4.1
- IBM GPFS Version 2.1 APAR
- IBM RSCT Version 2.3.1

5.3.2 GPFS software installation and configuration

In this scenario for GPFS, we use the same RSCT peer domain cluster previously defined for the VSD in 5.2.4, “Creating the RSCT peer domain” on page 196.

1. Install the GPFS filesets

   For this scenario, we install the GPFS filesets as specified in Example 5-30.

Example 5-30 GPFS filesets

```
LPAR1 #> lslpp -l | grep mmfs
  mmfs.base.cmds             3.5.0.6  APPLIED    GPFS File Manager Commands
  mmfs.base.rte              3.5.0.9  APPLIED    GPFS File Manager
  mmfs.gpfs.rte              2.1.0.9  APPLIED    GPFS File Manager
  mmfs.msg.en_US             3.5.0.5  APPLIED    GPFS Server Messages - U.S.
  mmfs.base.rte              3.5.0.9  APPLIED    GPFS File Manager
  mmfs.gpfs.rte              2.1.0.9  APPLIED    GPFS File Manager
  mmfs.gpfsdocs.data         3.5.0.2  APPLIED    GPFS Server Manpages and
```

2. Set up the PATH and MANPATH variables

   To access the GPFS commands, you have to set up the PATH variable by adding the directory that contains the GPFS binaries: /usr/lpp/mmfs/bin.

   There are three sets of man pages shipped with the GPFS for AIX 5L program product. There is one set for the GPFS cluster type SP, one set that covers both the GPFS cluster types RPD and HACMP, and one set for the GPFS cluster type LC. In order to access the correct set of man pages for GPFS on RPD, you must set your MANPATH environment variable to include /usr/lpp/mmfs/gpfsdocs/man/aix.

   In our environment, we use an RSCT peer domain (RPD). Therefore, we set up the variables on both nodes as specified in Example 5-31.

Example 5-31 PATH and MANPATH variables

```
echo “export PATH=$PATH:/usr/lpp/mmfs/bin” >> /.profile
echo “export MANPATH=:/usr/lpp/mmfs/gpfsdocs/man/aix” >> /.profile
```

3. Set up the “ipqmaxlen” network option

   The `ipqmaxlen` network option should be considered when configuring GPFS. The `ipqmaxlen` parameter controls the number of incoming packets that can exist on the IP interrupt queue. Since both GPFS and IBM Virtual Shared Disk use IP, the default of 128 is usually not enough. We recommended setting this to 512. See Example 5-32.

Example 5-32 ipqmaxlen setup

```
# no -r -o ipqmaxlen=512
```

Important: In order for this setting to take effect, the system must be rebooted.

4. Create the nodelist

   In order for the **mmcrcluster** to succeed, create a file containing the names of the nodes in the GPFS cluster as shown in Example 5-33.

Example 5-33 GPFS nodelist file

```
LPAR1 #> cat /.nodes
p690_LPAR1_ml0.itso.ibm.com
p690_LPAR2_ml0.itso.ibm.com
```

5. Create the GPFS cluster

   The **mmcrcluster** command creates a GPFS cluster from an existing RSCT peer domain. There can only be one GPFS cluster per RSCT peer domain. Upon successful completion of the **mmcrcluster** command, the /var/mmfs/gen/mmsdrfs and the /var/mmfs/gen/mmfsNodeData files are created on node in the cluster.

   These files should not be deleted under any circumstances.

   We use the **mmcrcluster** command as shown in Example 5-34 to create the GPFS cluster.

   This creates the GPFS cluster from nodes p690_LPAR1_ml0.itso.ibm.com and p690_LPAR2_ml0.itso.ibm.com, sets up the LPAR1 as the primary server and LPAR2 as the secondary server for this domain, and uses the **rsh** and **rcp** as the remote execution commands.

Example 5-34 Create the GPFS cluster

```
LPAR1 #> mmcrcluster -t rpd -p p690_LPAR1_ml0.itso.ibm.com -s p690_LPAR2_ml0.itso.ibm.com\
>                      -n /.nodes -r /usr/bin/rsh -R /usr/bin/rcp
Mon Nov  3 14:44:24 EST 2003: 6027-1664 mmcrcluster: Processing node
p690_LPAR1_ml0.itso.ibm.com
Mon Nov  3 14:44:25 EST 2003: 6027-1664 mmcrcluster: Processing node
p690_LPAR2_ml0.itso.ibm.com
mmcrcluster: Command successfully completed
mmcrcluster: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

6. Verify the GPFS cluster

   We use the **mmlscluster** command to verify the GPFS cluster, as shown in Example 5-35.

Example 5-35 GPFS cluster verification

```
LPAR1 #> mmlscluster

GPFS cluster information
========================
  Cluster id:  gpfs031103194423
  Remote shell command:      /usr/bin/rsh
  Remote file copy command:  /usr/bin/rcp

GPFS cluster data repository servers:
```

```
-------------------------------------
  Primary server:    p690_LPAR1_ml0.itso.ibm.com
  Secondary server:  p690_LPAR2_ml0.itso.ibm.com

Cluster nodes that are not assigned to a nodeset:
-------------------------------------------------
   1  p690_LPAR1_ml0  10.10.10.11      p690_LPAR1_ml0.itso.ibm.com
   2  p690_LPAR2_ml0  10.10.10.12      p690_LPAR2_ml0.itso.ibm.com
```

7. Create the GPFS nodeset

   The **mmconfig** command defines a new GPFS nodeset and configures GPFS prior to creating file systems. See Example 5-36. This command creates the *itso* nodeset from all nodes in the GPFS cluster and enables the single-node quorum feature. For more information regarding the single-node quorum feature, refer to 3.5.8, "GPFS quorum considerations" on page 125.

Example 5-36 Create a GPFS nodeset

```
LPAR1 #> mmconfig -a -C itso -U yes
mmconfig: Command successfully completed
mmconfig: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

8. Create the GPFS disk descriptor file (DescFile)

   The DescFile is the file containing the map of VSDs for the GPFS environment.

   For our purposes, we created the content in the DescFile as shown in the Table 5-6. We utilized the VSDs created in 5.2, "VSD implementation on an HPS" on page 194.

Table 5-6 The DescFile contents

Disk Name	Primary Server Name	Backup Server Name	Disk Usage	Failure Group
VSDp1s2vsd1n1	p690_LPAR1_ml0.itso.ibm.com	p690_LPAR2_ml0.itso.ibm.com	dataAndMetadata	1
VSDp2s1vsd1n2	p690_LPAR2_ml0.itso.ibm.com	p690_LPAR1_ml0.itso.ibm.com	dataAndMetadata	2

Our sample DescFile /.descfile is shown in Example 5-37.

Example 5-37 Our DescFile

```
VSDp1s2vsd1n1:p690_LPAR1_ml0.itso.ibm.com:p690_LPAR2_ml0.itso.ibm.com:dataAndMetadata:1
VSDp2s1vsd1n2:p690_LPAR2_ml0.itso.ibm.com:p690_LPAR1_ml0.itso.ibm.com:dataAndMetadata:2
```

9. Define the VSDs for GPFS using the **mmcrvsd** command.

   The **mmcrvsd** command is used to create or register Virtual Shared Disks for use by GPFS. When used to create Virtual Shared Disks, it follows the convention of creating one local volume group, one local logical volume, one global volume group, and one Virtual Shared Disk per physical volume. After the Virtual Shared Disk is created, it is configured and started on each node with a defined Virtual Shared Disk adapter.

   Where possible **mmcrvsd** creates and starts Virtual Shared Disk components in parallel. For instance, when multiple physical disk servers are specified in the disk descriptor file, their LVM components are created in parallel. Starting of all Virtual Shared Disks, on all nodes, always occurs in parallel.

   We use the **mmcrvsd** command to register the VDS disks in GPFS. We will use the Virtual Shared Disks that we created in 5.2, "VSD implementation on an HPS" on page 194. In this case, the name of the previously created Virtual Shared Disk is passed to the **mmcrvsd** command as the disk name in a disk descriptor. When **mmcrvsd** recognizes that the Virtual Shared Disk already exists, it stores this information in the GPFS configuration database (/var/mmfs/gen/mmsdrfs).

Example 5-38 shows the output from the **mmcrvsd** command in our environment.

Example 5-38 The mmcrvsd command output

```
LPAR1 #> mmcrvsd -F /.descfile

-------------------------------------------------------------------------------
Step 0:  Setting up environment.

-------------------------------------------------------------------------------
Step 1:  Making logical volumes.


-------------------------------------------------------------------------------
Step 2:  Varying off volume groups on primary nodes.

-------------------------------------------------------------------------------
Step 3:  Importing volume groups on any backup nodes.


-------------------------------------------------------------------------------
Step 4:  Varying on volume groups on primary nodes.

-------------------------------------------------------------------------------
Step 5:  Making global volume groups.

-------------------------------------------------------------------------------
Step 6:  Defining virtual shared disks.

-------------------------------------------------------------------------------
Step 7:  Writing new descriptor file for use by subsequent GPFS disk commands.

-------------------------------------------------------------------------------
Step 8:  Starting virtual shared disks on all nodes.

-------------------------------------------------------------------------------
Finished.
```

Tip: The **mmcrvsd** command may also be restarted, should one of the steps fail.

10.Start the GPFS cluster

Start the GPFS cluster using the **mmstartup** command, as shown in Example 5-39.

Example 5-39 Start the GPFS cluster

```
LPAR1 #> mmstartup -C itso
Mon Nov  3 16:18:52 EST 2003: 6027-1642 mmstartup: Starting GPFS ...
p690_LPAR1_ml0.itso.ibm.com:  0513-059 The mmfs Subsystem has been started. Subsystem PID
is 401506.
p690_LPAR2_ml0.itso.ibm.com:  0513-059 The mmfs Subsystem has been started. Subsystem PID
is 331970.
```

We then verify the progress of the GPFS start in the log file /var/adm/ras/mmfs.log.latest as shown in Example 5-40 on page 210.

Example 5-40 The mmfs.log.latest file

```
LPAR1 #> tail -30 mmfs.log.latest
Mon Nov  3 16:18:56 EST 2003 runmmfs starting
Removing old /var/adm/ras/mmfs.log.* files:
mv: 0653-401 Cannot rename /var/adm/ras/mmfs.log.previous to
/var/adm/ras/mmfs.log.previous.save:
             A file or directory in the path name does not exist.
Loading kernel extension from /usr/lpp/mmfs/bin . . .
/usr/lpp/mmfs/bin/aix64/mmfs64 loaded and configured.
Mon Nov  3 16:18:57 2003: GPFS: 6027-310 mmfsd64 initializing. {Version: 3.5.0.0   Built:
Sep  3 2003 22:47:23} ...
Mon Nov  3 16:18:57 2003: GPFS: 6027-1531 useSPSecurity no
Mon Nov  3 16:18:58 2003: GPFS: 6027-1853 Waiting for ml0 adapter group subscription
Mon Nov  3 16:18:58 2003: GPFS: 6027-1850 Successfully subscribed to ml0 group
Mon Nov  3 16:18:58 2003: GPFS: 6027-841 Cluster type: 'RPD'
Mon Nov  3 16:18:58 2003: GPFS: 6027-1865 Two-node nodeset: useSingleNodeQuorum yes.
Mon Nov  3 16:18:58 2003: Using TCP communication protocol
Mon Nov  3 16:18:58 2003: GPFS: 6027-1709 Accepted and connected to 10.10.10.12
Mon Nov  3 16:18:58 EST 2003 /var/mmfs/etc/gpfsready invoked
Mon Nov  3 16:18:58 2003: GPFS: 6027-300 mmfsd ready
```

11.Create the GPFS file system using the **mmcrfs** command

We use the **mmcrfs** command to create a GPFS file system. The first three options must be, in order, Mountpoint, Device, and either DiskDescList or DescFile. The block size and replication factors chosen will affect file system performance. There is a limitation of 32 GPFS file systems within a GPFS nodeset.

When issuing the **mmcrfs** command, the size of the command must not exceed the shell limit. If the shell limit is insufficient to specify the file system on the command line, shorten the command string by using the **-F** option and specifying the disk descriptors in a file. If the command is interrupted for any reason, you must use the **-v no** option on the next invocation of the command to ensure that the disks will be reused regardless of their previous status.

Upon successful execution of the **mmcrfs** command, these tasks are completed:

- Mount point directory is created on all nodes in the GPFS nodeset.
- File system is formatted.

We create the /fsp1s2 file system with associated /dev/fsp1s2dev device. In Example 5-41 This file system resides on the VSDp1s2vsd1n1 VSD and is in the failover group 1.

Example 5-41 /fsp1s2 file system creation

```
LPAR1 #> mmcrfs /fsp1s2 /dev/fsp1s2dev \
> "VSDp1s2vsd1n1:p690_LPAR1_ml0.itso.ibm.com:\
> p690_LPAR2_ml0.itso.ibm.com:dataAndMetadata:1" -n 2 -v no

GPFS: 6027-531 The following disks of fsp1s2dev will be formatted on node
p690_LPAR1.itso.ibm.com:
    VSDp1s2vsd1n1: size 5120000 KB
GPFS: 6027-540 Formatting file system ...
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Flushing Allocation Maps
GPFS: 6027-572 Completed creation of file system /dev/fsp1s2dev.
mmcrfs: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

We then create the /fsp1s2 file system with associated /dev/fsp1s2dev device. In Example 5-41 on page 210. This file system resides on the VSDp1s2vsd1n1 VSD and is in the failover group 2.

Example 5-42 /fsp1s2 file system creation

```
LPAR1 #> mmcrfs /fsp2s1 /dev/fsp2s1dev \
> "VSDp2s1vsd1n2:p690_LPAR2_ml0.itso.ibm.com:\
> p690_LPAR1_ml0.itso.ibm.com:dataAndMetadata:2" -n 2 -v no

GPFS: 6027-531 The following disks of fsp2s1dev will be formatted on node
p690_LPAR2.itso.ibm.com:
    VSDp2s1vsd1n2: size 5120000 KB
GPFS: 6027-540 Formatting file system ...
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Flushing Allocation Maps
GPFS: 6027-572 Completed creation of file system /dev/fsp2s1dev.
mmcrfs: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

12. Mount the GPFS file systems.

    The **mmcrfs** command used in step 11 on page 210 prepared the /etc/filsystems file by including the names of these file systems. See Example 5-43.

Example 5-43 The /etc/filesystem file verification

```
grep -p fsp /etc/filesystems
/fsp1s2:
        dev             = /dev/fsp1s2dev
        vfs             = mmfs
        nodename        = -
        mount           = mmfs
        type            = mmfs
        account         = false

/fsp2s1:
        dev             = /dev/fsp2s1dev
        vfs             = mmfs
        nodename        = -
        mount           = mmfs
        type            = mmfs
        account         = false
```

We use the standard AIX **mount** command to mount the newly created file systems on both nodes and verify their status using the **df -k** command as shown in Example 5-44.

Example 5-44 Mounting the filesystems

```
LPAR1 #> mount /fsp1s2
LPAR1 #> mount /fsp2s1
LPAR1 #> df -k | grep fs
/dev/fsp1s2dev     5120000   5113856    1%        9     1% /fsp1s2
/dev/fsp2s1dev     5120000   5113856    1%        9     1% /fsp2s1

LPAR2 #> mount /fsp1s2
LPAR2 #> mount /fsp2s1
```

```
LPAR2 #> df -k | grep fs
/dev/fsp1s2dev     5120000   5113856    1%        9     1% /fsp1s2
/dev/fsp2s1dev     5120000   5113856    1%        9     1% /fsp2s1
```

5.3.3 GPFS verification

In order to verify each software layer, we use UNIX, VSD and GPFS commands.

1. Verify the VSD data information using the **vsdatalst** command, as shown in Example 5-45.

Example 5-45 The vsdatalst command

```
LPAR1 #> vsdatalst  -v
              VSD Table
VSD name                        logical volume  Global Volume Group             minor# size_in_MB
------------------------------- --------------- ------------------------------- ------ ----------
VSDp1s2vsd1n1                   VSDp1s2lv1n1    VSDp1s2vgn1b2                        1 5000
VSDp2s1vsd1n2                   VSDp2s1lv1n2    VSDp2s1vgn2b1                        2 5000

LPAR1 #> vsdatalst  -n
       VSD Node Information
                                               Buddy Buffer
  node                 VSD      IP packet minimum maximum # maxbufs
number host_name       adapter    size     size    size
------ --------------- -------- --------- ------- ------- ---------
     1 p690_LPAR1_ml0. ml0          61440    4096  262144    4000
     2 p690_LPAR2_ml0. ml0          61440    4096  262144    4000

LPAR1 #> vsdatalst  -g
      VSD Global Volume Group Information
                                                Server Node Numbers
Global Volume Group name        Local VG name   primary      backup eio_recovery     recovery
server_list                          vsd_type
------------------------------- --------------- -------      ------ ------------     --------
------------------------------------  --------
VSDp1s2vgn1b2              VSDp1s2vg         1       2            1        0    0
VSD
VSDp2s1vgn2b1              VSDp2s1vg         2       1            1        0    0
VSD
```

2. Verify the VSD status with the **lsvsd -l** command on one of the nodes as shown in Example 5-46. The state column lists the current status of the shared drives.

Example 5-46 lsvsd -l

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                           size(MB)
server_list
 1      ACT    1      39        1      VSDp1s2vsd1n1                       5000          1
 2      ACT    2       0        0      VSDp2s1vsd1n2                       5000          2
```

3. Check the volume group status on the nodes using the **lspv** command on each node, as shown in Example 5-47.

Example 5-47 Volume group state on the nodes

```
LPAR1 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg       active
```

```
vpath1          0022be2a31fa6653                    VSDp2s1vg

LPAR2 #> lspv | grep vpath
vpath0          0022be2a31fa63ca                    VSDp1s2vg
vpath1          0022be2a31fa6653                    VSDp2s1vg       active
```

4. Verify the GPFS cluster configuration using the **mmlscluster** command, as shown in Example 5-48.

Example 5-48   mmlscluster

```
LPAR1 #> mmlscluster

GPFS cluster information
========================
  Cluster id:  gpfs031104011509
  Remote shell command:      /usr/bin/rsh
  Remote file copy command:  /usr/bin/rcp

GPFS cluster data repository servers:
-------------------------------------
  Primary server:    p690_LPAR1_ml0.itso.ibm.com
  Secondary server:  p690_LPAR2_ml0.itso.ibm.com

Nodes in nodeset itso:
----------------------
   1  p690_LPAR1_ml0  10.10.10.11      p690_LPAR1_ml0.itso.ibm.com
   2  p690_LPAR2_ml0  10.10.10.12      p690_LPAR2_ml0.itso.ibm.com
```

5. Verify the nodeset configuration by issuing the **mmlsconfig** commands, as shown in Example 5-49.

Example 5-49   mmlsconfig

```
LPAR1 #>  mmlsconfig
Configuration data for nodeset itso:
------------------------------------
clusterType rpd
comm_protocol TCP
multinode yes
autoload no
useSingleNodeQuorum yes
wait4RVSD yes
group Gpfs.itso
recgroup GpfsRec.itso

File systems in nodeset itso:
-----------------------------
/dev/fsp1s2dev
/dev/fsp2s1dev
```

6. Verify the availability of GPFS disks by issuing the **mmlsdisk** command. In Example 5-50, notice the disks show the “up” status in the availability column.

Example 5-50   mmlsdisk

```
LPAR1 #> mmlsdisk /dev/fsp1s2dev
disk         driver   sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- -------- ----- ------------- ------------
VSDp1s2vsd1n1 disk        512       1 yes      yes   ready         up
```

```
LPAR1 #> mmlsdisk /dev/fsp2s1dev
disk         driver   sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- -------- ----- ------------- ------------
VSDp2s1vsd1n2 disk        512       2 yes      yes   ready         up
```

7. Verify the file systems by issuing the **df** command on both nodes. See Example 5-51.

Example 5-51 output from df command on both nodes

```
LPAR1 #> df -k | grep fsp
/dev/fsp1s2dev     5120000   4472576   13%       10      1% /fsp1s2
/dev/fsp2s1dev     5120000   4640256   10%       10      1% /fsp2s1

LPAR2 #> df -k | grep fsp
/dev/fsp2s1dev     5120000   4640256   10%       10      1% /fsp2s1
/dev/fsp1s2dev     5120000   4472576   13%       10      1% /fsp1s2
```

Write and read test

We verify the GPFS file systems are readable and writable. For this purpose we create a *foo* file in /fsp1s2 file system using the **dd** command, and then we copy this file into the /fsp2s1 file system. See Example 5-52.

Example 5-52 Write and read test

```
LPAR1 #> dd if=/dev/zero of=/fsp1s2/foo count=1000 bs=1024K
1000+0 records in.
1000+0 records out.

LPAR1 #> ls -l /fsp1s2
total 512
-rw-r--r--   1 root     system   1048576000 Nov 03 21:00 foo

LPAR1 #> cd /fsp1s2 && echo "./foo" | backup -iqvf - | (cd /fsp2s1/ && restore -xqvf -)
a   1048576000 /fsp1s2/foo
The total size is 1048576000 bytes.
x   1048576000 /fsp1s2/foo

LPAR2 #> ls -l /fsp2s1
total 512
-rw-r--r--   1 root     system   1048576000 Nov 03 21:00 foo
```

The status of the **backup** command confirms the file has been copied successfully.

5.3.4 GPFS failover scenario

In this scenario, we create a SAMBA share on one of the GPFS file systems created in the previous scenario. LPAR1 is the server for the SAMBA file server daemon. The share is located on the /fsp2s1 GPFS file system. This file system exists on a VSD that is physically connected to LPAR2. We map the SAMBA share on a Windows workstation and copy a large *foo* file onto it. We then halt the LPAR2 node and watch the effects.

Prepare the SAMBA environment

Below are the steps that we take in order to prepare the SAMBA environment. For the scenario purposes, we use the SAMBA product Version 2.2.3.

1. Prepare the SAMBA configuration file.

Edit the /usr/local/samba/lib/smb.conf file. The contents are shown in Example 5-53.

Example 5-53 smb.conf file

```
[global]
     netbios name = p690
     workgroup = MYWORKGROUP
     security = user
     log file = /usr/local/samba/var/samba.log
     log level = 1
     socket options = TCP_NODELAY IPTOS_LOWDELAY SO_RCVBUF=8192 SO_SNDBUF=8192
     wins support = yes
     domain logons = yes
     logon drive = p:
     logon home = \\p690\%U
     os level = 99
     preferred master = yes

[fsp2s1]
     path = /fsp2s1
     guest ok = yes
     writeable = yes
     create mode = 0666
     directory mode = 0777
```

2. Verify the contents of the /fsp2s1 file systems.

   We use the **ls -l** command to list files in this file system. See Example 5-54.

Example 5-54 Contents of the /fsp2s1 file system

```
LPAR1 #> cd /fsp2s1
LPAR1 #> ls -l
total 512
-rw-r--r--   1 root     system   1048576000 Nov 03 21:00 foo
```

3. Start the SAMBA server.

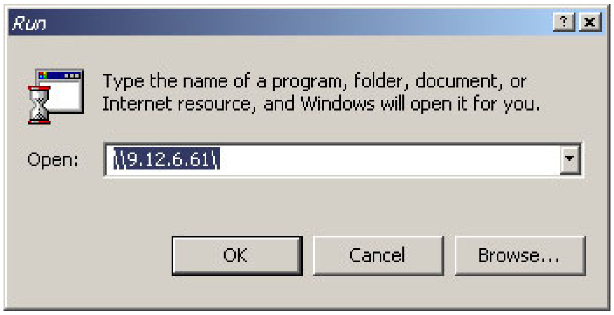
   We start the SAMBA server using the **start_smb.sh** script located in the /usr/local/samba directory. We verify the SAMBA server is running by issuing the **ps** command as shown in Example 5-55.

Example 5-55 SAMBA server verification

```
LPAR1 #> ps -ef | grep mbd
    root  98468      1   0 22:03:28      -  0:00 /usr/local/bin/nmbd
    root 102410  98468   0 22:03:28      -  0:00 /usr/local/bin/nmbd
    root 344312 376970   0 22:03:51      -  0:00 /usr/local/bin/smbd
    root 376970      1   0 22:03:28      -  0:00 /usr/local/bin/smbd
```

4. Access the network share on the Windows workstation.

   We access the SAMBA share using the Windows network share searching mechanism as shown in Example 5-6 on page 216.

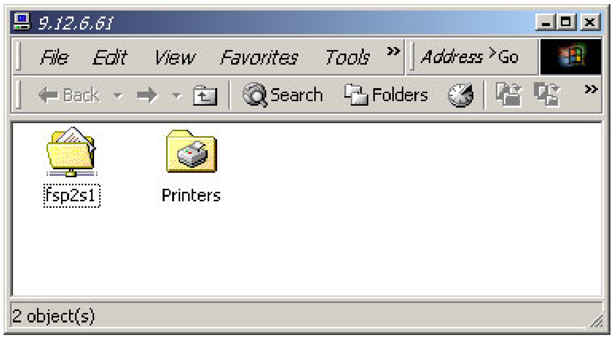


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: \\9.12.6.61\

Figure 5-6 Access the SAMBA share

The network drives shared by the SAMBA server appears after specifying the user name and password. See Figure 5-7.

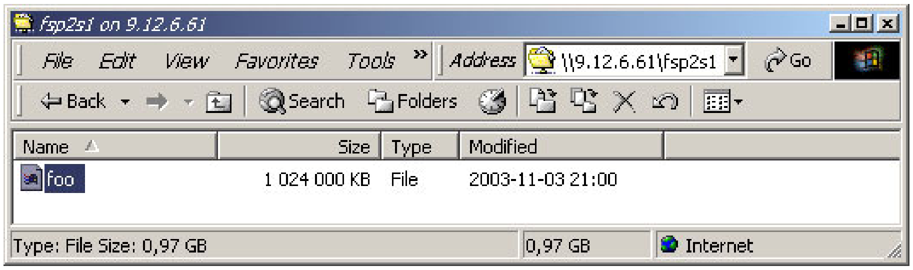


2 object(s)

Figure 5-7 The network object of SAMBA server

5. Verify the contents of the fsp2s1 SAMBA share.

   We verify the contents of the shared /fsp2s1 file system by double-clicking the appropriate icon in the shares list. See Figure 5-8.



Address \\9.12.6.61\fsp2s1

Name	Size	Type	Modified
foo	1 024 000 KB	File	2003-11-03 21:00

Type: File Size: 0,97 GB | 0,97 GB

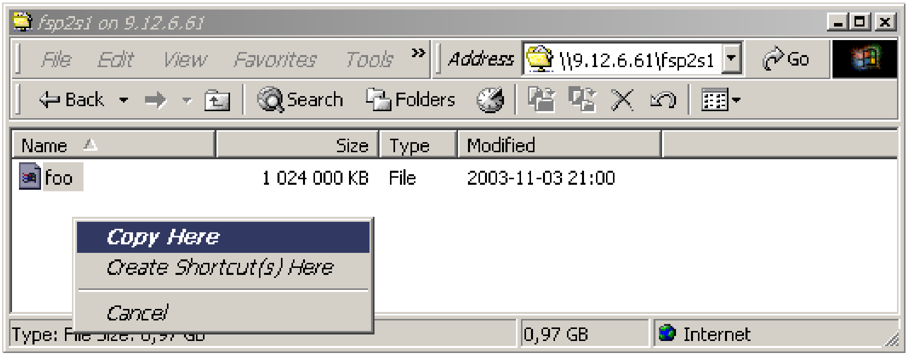
Figure 5-8 The contents of the SAMBA share

GPFS failover test

In the following steps we test the GPFS failover. We copy the *foo* file to the same directory and calculate its checksum. Then we halt the node LPAR2 to cause the GPFS failover.

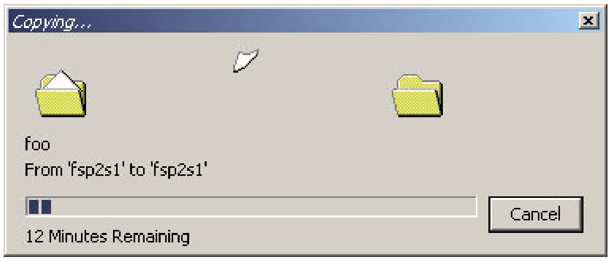
1. Copy the *foo* file

   Copy the *foo* file to the same directory. See Figure 5-9 on page 217 and Figure 5-10 on page 217.



Name	Size	Type	Modified
foo	1 024 000 KB	File	2003-11-03 21:00

Figure 5-9 Copy the foo file

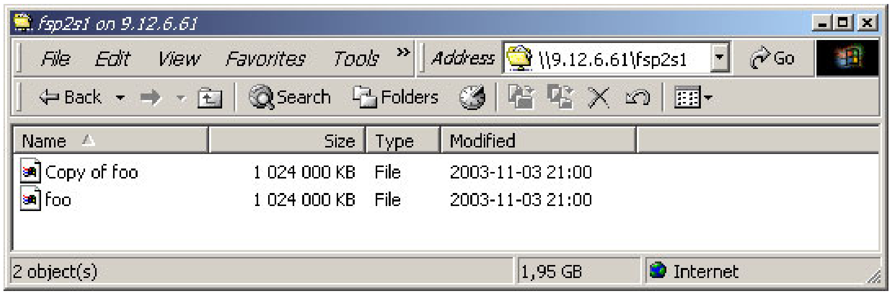


foo
From 'fsp2s1' to 'fsp2s1'

12 Minutes Remaining

Figure 5-10 File copy in progress

Verify the file has been copied correctly, as shown in Figure 5-11 and Example 5-56.



Name	Size	Type	Modified
Copy of foo	1 024 000 KB	File	2003-11-03 21:00
foo	1 024 000 KB	File	2003-11-03 21:00

Figure 5-11 Verify the copied file on the Windows workstation

Example 5-56 Verify the copied file on the SAMBA server

```
LPAR1 #> ls -l /fsp2s1
total 109056
-rw-rw-rw-   1 root     system   1048576000 Nov 03 21:00 Copy of foo
-rw-r--r--   1 root     system   1048576000 Nov 03 21:00 foo

LPAR1 #> sum /fsp2s1/*
36100 1024000 Copy of foo
36100 1024000 foo
```

2. Step 2. Halt the LPAR2 node

   We use the **fasthalt** command to halt the LPAR2. See Example 5-57.

Example 5-57 Halt the LPAR2

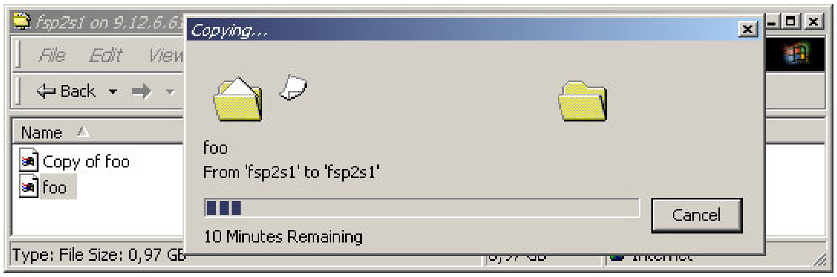
```
LPAR2 #> sync;sync;sync;fasthalt -q
```

```
....Halt completed....
```

This causes the VSDs to failover, and GPFS file systems should be accessible after a short period of time without having to re-open the share.

3. Verify the SAMBA share accessibility

   In order to verify the SAMBA share accessibility, we copy the *foo* file again. See Figure 5-12.



foo
From 'fsp2s1' to 'fsp2s1'
10 Minutes Remaining

Figure 5-12 Copy the foo file again

The file is being copied. This means the network share is accessible.

We also verify the status of the VSDs and GPFS file systems by using the **lsvsd** and **mmlsdisk** commands, as shown in Example 5-58.

Example 5-58 VSD and GPFS verification

```
LPAR1 #> lsvsd -l
minor  state server lv_major lv_minor vsd-name                           size(MB)
server_list
 1      ACT     1      39        1      VSDp1s2vsd1n1                        5000          1
 2      ACT     1      40        1      VSDp2s1vsd1n2                        5000          1

LPAR1 #> mmlsdisk /dev/fsp1s2dev
disk         driver   sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- -------- ----- ------------- ------------
VSDp1s2vsd1n1 disk        512       1 yes      yes   ready         up

LPAR1 #> mmlsdisk /dev/fsp2s1dev
disk         driver   sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- -------- ----- ------------- ------------
VSDp2s1vsd1n2 disk        512       2 yes      yes   ready         up
```

Part 3

Appendixes

Part 3 contains the appendixes. The information in each appendix is relevant to the topics of this redbook but difficult to include in any specific chapter. This information is in no particular order.

A

Quick references

The following are resources or references to resources that are relevant to the topics in this redbook.

A.1 AIX

A.1.1 NIM resource list

Example: A-1 bosinst.data file

```
control_flow:
    CONSOLE = Default
    INSTALL_METHOD = overwrite
    PROMPT = no
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = yes
    RUN_STARTUP = no
    RM_INST_ROOTS = no
    ERROR_EXIT =
    CUSTOMIZATION_FILE =
    TCB = no
    INSTALL_TYPE =
    BUNDLES =
    RECOVER_DEVICES = no
    BOSINST_DEBUG = no
    ACCEPT_LICENSES = yes
    DESKTOP = CDE
    INSTALL_DEVICES_AND_UPDATES = yes
    IMPORT_USER_VGS = no
    ENABLE_64BIT_KERNEL = yes
    CREATE_JFS2_FS = yes
    ALL_DEVICES_KERNELS = yes
    GRAPHICS_BUNDLE = yes
    DOC_SERVICES_BUNDLE = yes
    NETSCAPE_BUNDLE = no
    HTTP_SERVER_BUNDLE = no
    KERBEROS_5_BUNDLE = no
    SERVER_BUNDLE = yes
    ALT_DISK_INSTALL_BUNDLE = no
    REMOVE_JAVA_118 = no

target_disk_data:
    HDISKNAME = hdisk0

locale:
    BOSINST_LANG = en_US
    CULTURAL_CONVENTION = en_US
    MESSAGES = en_US
    KEYBOARD = en_US
```

Example: A-2 resolv.conf file

```
nameserver      9.114.66.2
domain  ppd.pok.ibm.com
search ppd.pok.ibm.com
```

Example: A-3 Example file of adapter definition for configuring the sn# and ml0 interfaces

```
###CSM_ADAPTERS_STANZA_FILE###
c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=sn
     netaddr=20.20.20.11
     location=U1.18-P1-H1/Q3
     subnet_mask=255.255.255.0

c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=sn
     netaddr=30.30.30.11
     location=U1.18-P1-H1/Q4
     subnet_mask=255.255.255.0

c121f1rp01.ppd.pok.ibm.com:
     machine_type=secondary
     network_type=ml
     interface_name=ml0
     netaddr=10.10.10.11
     subnet_mask=255.255.255.0
```

A.1.2 Peer domain management

- Preparation and security
  - preprpnode - Prepare a node for a peer domain for proper security and access
- RSCT peer domain commands:
  - mkrpdomain - Create a peer domain from one or more nodes.
  - rmrpdomain - Remove a peer domain
  - lsrpdomain - List characteristics of a peer domain
  - startrpdomain - Bring a peer domain online
  - stoprpdomain - Bring a peer domain offline
- RSCT peer node commands:
  - addrpnode - Add a node to the peer domain.
  - rmrpnode - Remove a node from a peer domain.
  - lsrpnode - List information about nodes in a peer domain.
  - startrpnode - Bring a node online
  - stoprpnode - Bring a node offline
- Communication group commands:
  - mkcomg - Create a communication group (HATS heartbeat ring)
  - rmcomg - Remove a communication group
  - lscomg - List information about a communication group
  - chcomg - Change a communication group

Caveats

Before issuing the first RMC generic command, set your desired management scope:

```
export CT_MANAGEMENT_SCOPE = 2
```

Where `2` is for an RSCT peer domain, `0` or unset is for a local node, and c`t_node_id` is stored in IBM.PeerNode along with the node's name (IP or hostname: Recommended "hostname" of box).

- Do not change all IP addresses on a node at the same time.
- Wait for lsrsrc IBM.NetworkInterface to show change before doing the last interface.

For more details, refer to *IBM Reliable Scalable Cluster Technology for Linux, RSCT Guide and Reference,* SA22-7892-01.

These are the same issues as with Service Agent and DLPAR. Hostname resolution and uniqueness of the node ID are critical.

RISC System Cluster Technology / Resource Monitoring and Control (RSCT/RMC)

A.2 HMC procedures

A.2.1 HMC quick network configuration

1. Enter `# hostname hmc1.flibityjibbit.com`
2. Put console window onto toolbar:

   ```
   # vi /usr/X11R6/lib/X11/icewm/toolbar
   ```

   Uncomment the first line.
3. Enable telnet and FTP:

   ```
   # /usr/sbin/ntsysv
   ```

   Scroll down then select **telnet** and **wu-ftpd**.
4. Activate the newly enabled services:

   ```
   # /sbin/chkconfig telnet on
   # /sbin/chkconfig wu-ftpd on
   ```
5. Configure the network:

   ```
   # vi /etc/sysconfig/network
      NETWORKING=yes
      HOSTNAME="hmc1.flibityjibit.com"
      GATEWAY="192.168.1.1"
      GATEWAYDEV="e0"
      FORWARD_IPV4="no"
   # vi /etc/sysconfig/network-scripts/ifcfg-eth0
      DEVICE=eth0
      ONBOOT=yes
      BOOTPROTO=static
      IPADDR="192.168.1.100"
      NETMASK="255.255.255.0"
   # /etc/resolv.conf
      domain flibityjibit.com
      search internal.flibityjibit.com flibityjibit.com holdingcompany.flibityjibit.com
      nameserver 192.168.1.2
   ```

A.2.1.1 Cage numbering for Regatta H+

Table A-1 Cage numbering for p690+

Cage Number	Cage Location	Service Location
0 (CEC)	[a]CEC (U17-U34)	U1.18
1	FR A U09-U12	U1.9
2	FR A U05-U08	U1.5
3	FR A U01-U04	U1.1
4	FR A U13-U16	U1.13
4 is not available if IBFs are installed.		
5	FR Z U01-U04	U2.1
6	FR Z U05-U08	U2.5
7	FR Z U09-U12	U2.9
8	FR Z U13-U16	U2.13
9	FR Z U19-U22	U2.19
9 is only available if IBFs are installed		

[a] *7040-681 is also known as p690 and Regatta H and H+*

A.2.1.2 Cage numbering for Regatta IH

Table A-2 Cage numbering for p655

Cage Number	Cage Location	Service Location
1	FR A U01, U03	U1.1
2	FR A U02,U04	U1.2
3	FR A U05, U07	U1.5
4	FR A U06, U08	U1.6
5	FR A U09, U11	U1.9
6	FR A U10, U12	U1.10
7	FR A U13, U15	U1.13
8	FR A U14, U16	U1.14
U1.17 and U1.18 reserved for IBF		
9	FR A U19, U21	U1.19
10	FR A U20, U22	U1.20
11	FR A U23, U25	U1.23
12	FR A U24, U26	U1.24
13	FR A U27, U29	U1.27
14	FR A U28, U30	U1.28
15	FR A U31, U33	U1.31

Cage Number	Cage Location	Service Location
16	FR A U32, U34	U1.32
Z-Frame[a] not used		

[a] 7039-651 is also known as p655 or Regatta IH

A.2.1.3 Cage numbering for switch-only frame

Table A-3 Cage numbering for switch-only frame

Cage Number	Cage Location	Service Location
1	FR A U01-U04	U1.1
3	FR A U05-U08	U1.5
5	FR A U09-U12	U1.9
7 (IBF Option)	FR A U13-U16	U1.13
U1.17 and U1.8 reserved for IBF.		
9	FR A U19-U22	U1.19
11	FR A U23-U26	U1.23
13	FR A U27-U30	U1.27
15	FR A U31-U34	U1.31
17	FR Z U01-U04	U2.1
19	FR Z U05-U08	U2.5
21	FR Z U09-U12	U2.9
23	FR Z U13-U16	U2.13
25	FR Z U19-U22	U2.19
27	FR Z U23-U26	U2.23
29	FR Z U27-U30	U2.27
31	FR Z U31-U34	U2.31

A.2.2 Service focal point failures

Use this subsection if Service Focal Point is not receiving service events from your LPARs, or if DLPAR is "not configured".

Check for any of these failures:

- `HMC# lspartition -debug`: Missing or active<0> and no O/S shown.
- `HMC# lsrsrc IBM.ManagedNode`: Missing some or all nodes.
- `HMC# CT_CONTACT=<LPAR> lsrsrc IBM.ManagementServer:` Could not authenticate user.
- `LPAR# lsrsrc IBM.ManagementServer`: Missing the HMC.
- `LPAR# CT_CONTACT=<HMC> lsrsrc IBM.ManagedNode:` Could not authenticate user.
- `BOTH# more /var/ct/cfg/ctrmc.acls`: Should show several IBM.* stanzas.
- `BOTH# ctsthl -l | grep Host`: Should show HMC and LPARs.

- `HMC# lsrsrc IBM.ServiceEvent`: Doesn't show entries that are on LPARs. This means there is a hostname resolution problem or ACL problem.
- `LPAR# lssrc -a | grep IBM`: IBM.DRM and IBM.HostRM are not listed.

Note: It is normal for IBM.CSMAgentRM to be inoperative on HMC.

A.2.2.1 Corrective operations

1. Verify AIX fileset levels
   - `# instfix -ik | grep ML`: Must be AIX 5.1.0.0 ML02 or higher. DLPAR requires AIX 5.2.
   - `# rpm -q -a`: Open.CIMOM should be installed on LPARs from Linux Toolkit CD.
   - `# lslpp -L rsct.core.*`: These should be 2.2.1.30 or later.
2. Verify HMC levels

   From the local HMC select **Help -> About/**. Or enter:

   ```
   # /opt/hsc/bin/lshsc -v | grep RM
   ```

   DLPAR required HMC should be R3 V1.2 or later.
3. Check name resolution
   - nsswitch.conf and netsvc.conf should match among all systems for local vs bind.
   - If using bind/dns, /etc/resolv.conf should match.
   - nslookup / host commands should match forward and reverse from all hosts.
   - Host name on all systems should be correct.
   - Make sure ping from and to HMC and LPAR shows proper host names.
4. Change the HMC's NIC driver

   Sometimes this is necessary due to the default driver being fairly poor at half duplex, add-in cards, and just generally not being as good.

   Become root on the HMC.

   ```
   # vi /etc/modules.conf
   ```

   Change the alias for eepro100 to e100.

   Save and reboot.
5. Remove bad entries for the HMC

   This can be caused by config changes or recent setup.

   Log in to the LPAR:

   ```
   # rmrsrc -s "Name!=\"\" " IBM.ManagementServer
   # /usr/sbin/rsct/install/bin/recfgct
   # cd /var/ct/cfg
   # mv ct_has.thl ct_has.thl.old
   # mv ctrmc.acls ctrmc.acls.old
   # rmcctrl -z; rmcctrl -A; rmcctrl -p
   ```
6. From the HMC, clear security and note list

   ```
   # cd /var/ct/cfg
   # mv ct_has.thl ct_has.thl.old
   # mv ctrmc.acls ctrmc.acls.old
   # rmrsrc -s "Name!=\"\" " IBM.ManagedNode
   # rmcctrl -z; rmcctrl -A; rmcctrl -p
   ```

Wait 15 minutes after you have done the above. If none of these corrects the problem, call support for more detailed instructions.

A.2.3 Service Agent rebuild

This procedure covers rebuilding Service Agent should it become corrupted and impossible to issue your Save Upgrade Data or use Service Agent.

1. Download HMC corrective service image

   This should match what your HMC is running:

   ```
   # tn lpar#
   ```

   Log in as root.

   ```
   # cd /tmp
      # ftp techsupport.services.ibm.com   or 207.25.253.26
      Name: anonymous
      Password: email@address
      ftp>cd /eserver/pseries/hmc/fixes/upgrades
      ftp>bin
      ftp>mget HMC_Update_R#V#.#.zip
      ftp>bye
   # ls
   ```

   You should see the HMC fix here.

2. Access HMC command line

   Presently the HMC as 3.2.3 instituted a restricted shell through SSH. If you have bypassed this, you may SSH in. If you have not bypassed this, press Alt+F1 from the HMC console.

   ```
   Login: hscroot
   Password: abc123 (or your site's password)
   # su -
      password: passw0rd
   # cd /tmp
   # mkdir hmcfix
   # cd hmcfix
   # ftp lpar#
      Name: root
      Password: password - > the valid root password on that box
      ftp>cd /tmp
      ftp>bin
      ftp>mget HMC*
      ftp>bye
   # ls
   ```

   You should see the HMC fix here.

   ```
   # unzip HMC_Update_R#V#.#.zip
   ```

   All the .rpm filesets explode out in the directory.

3. Uninstall the service agent fileset

   ```
   # rpm --erase SvcAgentHSC-#.#.#-0
      Stopping the ODS...
      Stopping the ESS...
      Stopping the Advanced User Interface Screen....
      Advanced User Interface Stopped.

      error: cannot remove /var/svcagent/logs - directory not empty
      error: cannot remove /var/svcagent/locks - directory not empty
      error: cannot remove /usr/svcagent - directory not empty
   ```

```
error: cannot remove /home/svcagent - directory not empty
```

This should match the version you downloaded. If not, then the above command will fail. That means you downloaded the wrong corrective service package.

```
# rm -r /var/svcagent
```

4. Reinstall Service Agent

```
# rpm -i SvcAgentHSC-1.2.0-0.i386.rpm   (or 1.0.0-1 for HMC v2)
Upgrading Service Agent...
```

5. Configure Service Agent
   a. Go to the HMC GUI.
   b. Select **Service Applications -> Service Agent -> Start Service Agent Processes**.

      This should go through a whole bunch of messages but finally starts the process.
   c. Now click **Service Agent UI** - registration/customization.
   d. Enter your password of `password`.
   e. Fill in all the Service Agent fields for the client, customize the dialer, etc.
   f. Then try the test pmr again.

A.2.4 Upgrading the frame microcode using the instfru command

Useful instfru options:

1. List the ports and frames accessible from HMC:

   ```
   instfru -lp
   ```

2. List the available frame code versions installed on your HMC:

   ```
   instfru -lv
   ```

3. List what frame components would be updated for a specified code version:

   ```
   instfru -ld -a -v <mcode_ver> -p <frame#>
   ```

4. Update the frame code:

   ```
   instfru -d -a -v <mcode_ver> -p <frame#>
   ```

A.2.5 Output example of /usr/local/hsctool/hps_check.pl script

The script must be run as root user on HMC. Example A-4 shows an output of the script for a configuration with two pSeries HPSs and three CECs. Each CEC has two 4-link SNI adapters. Each SNI adapter is connected to a switch using the four links on the book.

Example: A-4 Output from /usr/local/hsctool/hps_check.pl script

```
-----------------------------------------------------------------------------------------------------------
---
VPORT:  fffffe01  CEC_NAME: c121                        CEC_MTMS:  7040-681 02210CB
FRAME:         1  HMC_REG:  1eff  POLL_FREQ:   45       CEC_STATE: (01) CEC is IPL_READY.
                  FNM_REG:  0100  HMC_CONN:    01       OP_PANEL:  LPAR...

SNI Mapping:                                                        Switch Neighbor:
Lpar Name                  Lpar# Sni# => Adapter#  Csp#  Cronus# => Frame Cage Chip Port : Timed?  MPA  TOD
                               5    0           0     2        5        2    3    5    0      YES  YES  MAS
                               5    1           1     3        4        3    3    5    0      YES  YES  SLV
                               5    2           2     6       13        2    3    7    0      YES  YES  SLV
                               5    3           3     7       12        3    3    7    0      YES  YES  SLV
```

```
                               5    4           4     8        6        2    3    5    2      YES  YES  SLV
                               5    5           5     9        7        3    3    5    2      YES  YES  SLV
                               5    6           6    12       14        2    3    7    2      YES  YES  SLV
                               5    7           7    13       15        3    3    7    2      YES  YES  SLV

Mapping:                Neighbor:        Summary:       Registers:
PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TIMED     0x24030  MP_AVAIL   0x6050   TOD
0x20000
YY   0   2   5   1    4    2   3   5   0   YES YES MAS  00000000 00000000  10000000 00000000  81fce1ac
cf9bb3fc
YY   1   3   4   1   20    3   3   5   0   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
d39f4248
YY   2   6  13   1   12    2   3   7   0   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
d8795561
YY   3   7  12   1   28    3   3   7   0   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
dc7a12ad
YY   4   8   6   1    6    2   3   5   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e152a216
YY   5   9   7   1   22    3   3   5   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e554014f
YY   6  12  14   1   14    2   3   7   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
ea2c93c0
YY   7  13  15   1   30    3   3   7   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
ee2dd786

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  PHYS_ID   0x2B000  NEIGH_ID  0x23020  WHO_MAST
0x21040
YY   0   2   5   1    4    2   3   5   0   YES YES MAS  10040000 00000000  04000001 20002035  a0000000
00000000
YY   1   3   4   1   20    3   3   5   0   YES YES SLV  10140000 00000000  04000001 30003035  a0000000
00000000
YY   2   6  13   1   12    2   3   7   0   YES YES SLV  100c0000 00000000  04000001 20002037  a0000000
00000000
YY   3   7  12   1   28    3   3   7   0   YES YES SLV  101c0000 00000000  04000001 30003037  a0000000
00000000
YY   4   8   6   1    6    2   3   5   2   YES YES SLV  10060000 00000000  04000005 20002035  a0000000
00000000
YY   5   9   7   1   22    3   3   5   2   YES YES SLV  10160000 00000000  04000005 30003035  a0000000
00000000
YY   6  12  14   1   14    2   3   7   2   YES YES SLV  100e0000 00000000  04000005 20002037  a0000000
00000000
YY   7  13  15   1   30    3   3   7   2   YES YES SLV  101e0000 00000000  04000005 30003037  a0000000
00000000

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TOD_GEN   0x21030  TOD_MAST  0x21000  TOD_BACK
0x21010
YY   0   2   5   1    4    2   3   5   0   YES YES MAS  a070e000 00000000  80000000 00000000  00000000
00000000
YY   1   3   4   1   20    3   3   5   0   YES YES SLV  1070e000 00000000  00000000 00000000  00000000
00000000
YY   2   6  13   1   12    2   3   7   0   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   3   7  12   1   28    3   3   7   0   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   4   8   6   1    6    2   3   5   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   5   9   7   1   22    3   3   5   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   6  12  14   1   14    2   3   7   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
```

```
YY   7  13  15   1   30    3   3   7   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000

-----------------------------------------------------------------------------------------------------------
---
VPORT:  fffffe02  CEC_NAME: c131                        CEC_MTMS:  7040-681 0220CBB
FRAME:         2  HMC_REG:  1eff  POLL_FREQ:   45       CEC_STATE: (01) CEC is IPL_READY.
                  FNM_REG:  0100  HMC_CONN:    01       OP_PANEL:  LPAR...

SNI Mapping:                                                        Switch Neighbor:
Lpar Name                  Lpar# Sni# => Adapter#  Csp#  Cronus# => Frame Cage Chip Port : Timed?  MPA  TOD
                               5    0           0     2        5        2    3    6    0      YES  YES  BAK
                               5    1           1     3        4        3    3    6    0      YES  YES  BAK
                               5    2           2     6       13        2    3    4    0      YES  YES  SLV
                               5    3           3     7       12        3    3    4    0      YES  YES  SLV
                               5    4           4     8        6        2    3    6    2      YES  YES  SLV
                               5    5           5     9        7        3    3    6    2      YES  YES  SLV
                               5    6           6    12       14        2    3    4    2      YES  YES  SLV
                               5    7           7    13       15        3    3    4    2      YES  YES  SLV

Mapping:                Neighbor:        Summary:       Registers:
PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TIMED     0x24030  MP_AVAIL   0x6050  TOD
0x20000
YY   0   2   5   1    8    2   3   6   0   YES YES BAK  00000000 00000000  10000000 00000000  81fce1ac
cf9bd3ec
YY   1   3   4   1   24    3   3   6   0   YES YES BAK  00000000 00000000  10000000 00000000  81fce1ac
d39ff143
YY   2   6  13   1    0    2   3   4   0   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
d879928c
YY   3   7  12   1   16    3   3   4   0   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
dc7a69c3
YY   4   8   6   1   10    2   3   6   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e151a9e3
YY   5   9   7   1   26    3   3   6   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e552845d
YY   6  12  14   1    2    2   3   4   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
ea2c1cba
YY   7  13  15   1   18    3   3   4   2   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
ee2dc43f

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  PHYS_ID   0x2B000  NEIGH_ID  0x23020  WHO_MAST
0x21040
YY   0   2   5   1    8    2   3   6   0   YES YES BAK  10080000 00000000  04000001 20002036  a0000000
00000000
YY   1   3   4   1   24    3   3   6   0   YES YES BAK  10180000 00000000  04000001 30003036  a0000000
00000000
YY   2   6  13   1    0    2   3   4   0   YES YES SLV  10000000 00000000  04000001 20002034  a0000000
00000000
YY   3   7  12   1   16    3   3   4   0   YES YES SLV  10100000 00000000  04000001 30003034  a0000000
00000000
YY   4   8   6   1   10    2   3   6   2   YES YES SLV  100a0000 00000000  04000005 20002036  a0000000
00000000
YY   5   9   7   1   26    3   3   6   2   YES YES SLV  101a0000 00000000  04000005 30003036  a0000000
00000000
YY   6  12  14   1    2    2   3   4   2   YES YES SLV  10020000 00000000  04000005 20002034  a0000000
00000000
YY   7  13  15   1   18    3   3   4   2   YES YES SLV  10120000 00000000  04000005 30003034  a0000000
00000000
```

```
PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TOD_GEN   0x21030  TOD_MAST  0x21000  TOD_BACK
0x21010
YY   0   2   5   1    8    2   3   6   0   YES YES BAK  c070e000 00000000  00000000 00000000  80000000
00000000
YY   1   3   4   1   24    3   3   6   0   YES YES BAK  b070e000 00000000  00000000 00000000  80000000
00000000
YY   2   6  13   1    0    2   3   4   0   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   3   7  12   1   16    3   3   4   0   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   4   8   6   1   10    2   3   6   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   5   9   7   1   26    3   3   6   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   6  12  14   1    2    2   3   4   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   7  13  15   1   18    3   3   4   2   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000

-----------------------------------------------------------------------------------------------------------
---
VPORT:  fffffe03  CEC_NAME: c132                        CEC_MTMS:  7040-681 0293B0A
FRAME:         3  HMC_REG:  1eff  POLL_FREQ:   45       CEC_STATE: (01) CEC is IPL_READY.
                  FNM_REG:  0100  HMC_CONN:    01       OP_PANEL:  LPAR...

SNI Mapping:                                                        Switch Neighbor:
Lpar Name                  Lpar# Sni# => Adapter#  Csp#  Cronus# => Frame Cage Chip Port : Timed?  MPA  TOD
                               1    0           0     2        5        2    3    5    1      YES  YES  BAK
                               1    1           1     3        4        3    3    5    1      YES  YES  BAK
                               2    0           6    12       14        2    3    7    3      YES  YES  SLV
                               2    1           7    13       15        3    3    7    3      YES  YES  SLV
                               3    0           4     8        6        2    3    5    3      YES  YES  SLV
                               3    1           5     9        7        3    3    5    3      YES  YES  SLV
                               4    0           2     6       13        2    3    7    1      YES  YES  SLV
                               4    1           3     7       12        3    3    7    1      YES  YES  SLV

Mapping:                Neighbor:        Summary:       Registers:
PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TIMED     0x24030  MP_AVAIL   0x6050  TOD
0x20000
YY   0   2   5   1    5    2   3   5   1   YES YES BAK  00010000 00000000  10000000 00000000  81fce1ac
d0345705
YY   1   3   4   1   21    3   3   5   1   YES YES BAK  00000000 00000000  10000000 00000000  81fce1ac
d4354f71
YY   2   6  13   1   13    2   3   7   1   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
d90cb739
YY   3   7  12   1   29    3   3   7   1   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
dd0f47ba
YY   4   8   6   1    7    2   3   5   3   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e1e69988
YY   5   9   7   1   23    3   3   5   3   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
e5e79788
YY   6  12  14   1   15    2   3   7   3   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
eac12481
YY   7  13  15   1   31    3   3   7   3   YES YES SLV  00000000 00000000  10000000 00000000  81fce1ac
eec18e69

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  PHYS_ID   0x2B000  NEIGH_ID  0x23020  WHO_MAST
0x21040
YY   0   2   5   1    5    2   3   5   1   YES YES BAK  10050000 00000000  04000003 20002035  a0000000
00000000
```

```
YY   1   3   4   1   21    3   3   5   1   YES YES BAK  10150000 00000000  04000003 30003035  a0000000
00000000
YY   2   6  13   1   13    2   3   7   1   YES YES SLV  100d0000 00000000  04000003 20002037  a0000000
00000000
YY   3   7  12   1   29    3   3   7   1   YES YES SLV  101d0000 00000000  04000003 30003037  a0000000
00000000
YY   4   8   6   1    7    2   3   5   3   YES YES SLV  10070000 00000000  04000007 20002035  a0000000
00000000
YY   5   9   7   1   23    3   3   5   3   YES YES SLV  10170000 00000000  04000007 30003035  a0000000
00000000
YY   6  12  14   1   15    2   3   7   3   YES YES SLV  100f0000 00000000  04000007 20002037  a0000000
00000000
YY   7  13  15   1   31    3   3   7   3   YES YES SLV  101f0000 00000000  04000007 30003037  a0000000
00000000

PF Adp Csp Crn Net Endp Fram Cag Chp Prt Timed MPA TOD  TOD_GEN   0x21030  TOD_MAST  0x21000  TOD_BACK
0x21010
YY   0   2   5   1    5    2   3   5   1   YES YES BAK  e070e000 00000000  00000000 00000000  80000000
00000000
YY   1   3   4   1   21    3   3   5   1   YES YES BAK  d070e000 00000000  00000000 00000000  80000000
00000000
YY   2   6  13   1   13    2   3   7   1   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   3   7  12   1   29    3   3   7   1   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   4   8   6   1    7    2   3   5   3   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   5   9   7   1   23    3   3   5   3   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   6  12  14   1   15    2   3   7   3   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
YY   7  13  15   1   31    3   3   7   3   YES YES SLV  0070e000 00000000  00000000 00000000  00000000
00000000
```

B

Troubleshooting

In this appendix, we touch on some of the resources that can be used in troubleshooting an HPS environment.

B.1 Symptoms of common problems

Several customers have reported problems logging in locally to the console after installing HMC 3.2.4. There is a workaround for this problem:

1. From the console, press Ctrl+Alt+F1.

   ```
   login: hscroot
   password: abc123
   $ su root
   password: passw0rd
   # /sbin/ldconfig /usr/X11R6/lib
   # exit
   ```

2. Press Ctrl+Alt+F2.
3. Log in as hscroot from the console GUI.
4. Remote via SSH

   ```
   $ ssh hmcname -l hscroot
   $ lshsc -v
   ```

   a. Make note of the contents of the SE field (serial number).

      ```
      $ exit
      ```

   b. Call support and request a PE passcode.

   c. Create a user from the HMC GUI named “hscpe”.

   d. Set its password to what was given by support:

      ```
      $ ssh hmcname -l hscpe
      $ pesh <SERIAL>
      password: <pepasscode>
      $ su root
      password: passw0rd
      # /sbin/ldconfig /usr/X11R6/lib
      # exit
      ```

   e. Have someone log in as hscroot from the console GUI.

B.2 Typical switch issues

The following are some of the issues that may appear in an HPS environment:

1. ConfigRM issues
   - Domain is not active (**`lsrpdomain`**)
   - Node may not be online (**`lsrpnode`**)
   - Adapter is not harvested or inactive (**`lsrsrc IBM.NetworkInterface`**).
2. HATS/HAGS issues
   - Adapter is reported down (**`lssrc -ls cthats`**).
   - HAGS domain not established (**`lssrc -ls cthags`**, or **`hagsns -s cthags`**).
   - HAGS protocol is not finished (**`hagsvote -ls cthags`**).
3. HAGSGLSM issues
   - Membership not formed (**`lssrc -ls cthagsglsm`**)

B.3 Tracing

To set up the HMC tracing mechanisms to collect detailed information about the HPS management software stack, do the following:

1. Turn on Hardware Server (HS) tracing

   All hardware service interface events will be logged to /var/hsc/log/bpa_logger.log.

   a. To enable tracing to /var/hsc/log/iqzdtrac.trm:

   ```
   su root
   mkdir -p /console/data
   ```

   b. Edit /console/data/iqzdtctl.trm and add:

   ```
   AHDW*  TFDL
   ```

   c. Edit /etc/inittab and add at the bottom:

   ```
   # Run the hdwr_svr trace all of the time.
   iqz:2345:respawn:/usr/bin/iqzdtcon -s100000 1>/dev/null 2>&1
   /sbin/telinit q
   ```

   The trace will now go to /var/hsc/log/iqzdtrac.trm.

2. jt - Switch JTAG access

   jt provides low-level JTAG access to the switch as shown in Example B-1.

   – Engineering tool

     - Interactive access to switch registers
     - Interactive BPA commands
     - Interface for switch trace

   – Test tool

     - Error injection
     - Status monitoring

   – Performance tool

     - Start/stop/read performance counters "jstat"

   – CE tool

     - Low- level switch access

Example: B-1 Sample JTAG session

```
$ jt
> d localhost:3:7
> q d
  0: localhost:3:0 (e518f409) ACTIVE BPA
  1: localhost:3:7 (e518f409) ACTIVE DCA
  2: localhost:1:0 (e518f406) AVAILABLE BPA
> j 4 34
  1  4 - 20009074
> j (0:3) (32 33)
  1  0 - 01fa5c1d 372bc49c
     1 - 01fa5c1d 36a22535
     2 - 01fa5c1d 372bc49f
     3 - 01fa5c1d 36a22535
> ! cat timing.jt
# Read the timing states of the external link chip ports
# pjl - 30.1.03

! echo '  port:  0   1    2   3'
```

```
j (0:7)(b c)

> f timing
  port:  0   1    2   3
  1  0 - 00800080 00800080
     1 - 00800080 00800080
     2 - 00800080 00800080
     ...
> h
Switch JTAG Utility - Compiled: Aug 21 2003 12:26:30
       ack (a) - (no operands) - Acknowledge any asynchronous messages
       ...
     power (P) - <on|off> - Turn power to the switch board on or off
      read (j) - [chip@|(chip list)] [reg@|(reg list)] - Read JTAG reg[s] from chip[s]
       ...
> help read

read  [chip@|(chip list)] [reg@|(reg list)] - Read JTAG reg[s] from chip[s]

Display switch or interposer (FBC/LDC) registers.
Ranges are specified with colons - "j ( 0:3 5 7 ) ( 1a:1d )"
All numeric args are in hex.  "read" without operands repeats
the last read.
```

3. Create an IP dump for debug

   ```
   /usr/sbin/ifsn_dump -Z snX > filename
   ```

4. Automatic snaps

   ```
   /var/adm/sni/snaps/*.sni.snap.tar.Z
   ```

5. No recovery for your daemon traces

   ```
   /var/adm/sni/nrd.trace
   ```

6. Adapter dumps

   To collect dump data from the SNI adapters, perform the following operations:

   a. Acquire the dump

      ```
      cd /var/adm/sni/sniX <X: 0...7>
      uncompress sun_ucode_dump(.n).Z
      ```

   b. Format the log

      ```
      /colony/CSS/fed/bin/canopus_format_log -d
      ```

   c. Check for trace length

      ```
      cat UC_info
      ```

   d. Format the trace data

      ```
      /colony/CSS/fed/bin/canopus_format_trace UC_trace.bin UC_trace.trc  <length>
      ```

7. FNM daemon tracing

   See Example B-2 on page 240.

B.3.1 Common tracing errors

Some of the common tracing errors are:

1. No trace written to trace file

   a. Make sure FNM trace library, /opt/hsc/lib/libtrace.so, exists.

   b. Make sure trace daemon is running:

      ```
      # ps -wef | grep /opt/hsc/bin/tracelogd
      ```

   c. If trace daemon is not running, check whether startFNMTrace script is running.
   d. If it is running, check /tmp/startFNMtrace.log for timestamp when tracelogd exits.
   e. Look at daemon output in /tmp/tracelogd.stdout and /tmp/tracelogd.stderr. These record all system call failures, errno and error messages.
   f. Make sure /opt/hsc/lib/libeventlog.so exists.
   g. Check /tmp/fnmerrlog.log for errors. It records sys_rc and comp_rc when failure occurs.
   h. Match against iqyyint.h to decypher.
2. Java exceptions thrown
   a. No class or method found calling FNM Logging

      ```
      # locate SNMTracelog.jar
      ```

      /usr/websm/codebase/pluginjars/SNMTracelog.jar should be a link to /opt/ccfw/jars/SNMTracelog.jar.

   b. No class or method found calling com.ibm.hwmca.*

      Make sure /opt/ccfw/ccfw.jar exists.

   c. Failure loading system event messages in log viewer

      Check existence of /opt/ccfw/com/ibm/tracelog/swtevent.properties.

   d. Core dump

      Use **gdb** to debug.

B.3.2 After fixing the problem

There are two ways to start FNM Trace daemon:

```
/etc/inittab (default)
trlg:2345:respawn:/opt/hsc/bin/startFNMTrace 1>/dev/null 2>&1
```

or

```
/opt/hsc/bin/tracelogd &
```

B.3.3 RSCT related

1. RMC trace file

   ```
   rpttr -o dtic /var/ct/IW/log/mc/trace
   ```

2. ConfigRM trace file

   ```
   rpttr -o dtic /var/IW/log/mc/IBM.ConfigRM/trace
   ```

B.4 Config files

This section presents the switch-related configuration files on the HMC.

B.4.1 Switch daemon

Example: B-2 FNM trace configuration file

```
# cat /opt/hsc/data/tracelogd.config
   # detail_level options: TRACE_NOTIFY, STATUS_REPORT, COMMENTS, WARNING
   #      USER_ERR, CONFIG_ERR, SOFTWARE_ERR, HARDWARE_ERR
   # appl_name options: FNM_COMM, FNM_DIAGS, FNM_ELA, FNM_INIT, FNM_LOGGING,
   #      FNM_PROVIDERS, FNM_RECOVERY, FNM_ROUTING, FNM_GUI, FNM_HMC, FNM_DB,
   #      FNM_EXT, FNM_VPD
   byte_limit=6000000
   appl_name=FNM
   detail_level=TRACE_NOTIFY
```

B.4.2 hdwr_svr config files

- /opt/hsc/data/SvcAgentTTYConfig
  - Must exist and may contain up to one serial device name (for example, /dev/ttyS1).
  - The device named will be ignored by hdwr_svr and will be reserved for Service Agent call-home modem.
  - Changes are handled dynamically by hdwr_svr.
- /opt/hsc/data/HmcTTYConfig
  - This file is optional.
  - This file contains a list of serial port devices for hdwr_svr to ignore, listed one per line.
  - Changes are handled dynamically by hdwr_svr.
- /opt/hsc/data/HmcNetConfig
  - File must exist in multi-HMC clusters and must contain IP addresses of all HMCs in cluster, one per line.
  - It can contain # style comments. For example:

    ```
    9.111.113.1     #c678hmc1
    #9.111.113.2     #c678hmc3
    ```

  - Changes are handled dynamically by hdwr_svr.
- /opt/hsc/data/hmcconfig.def
  - This file must exist and contains the default timeouts for several different timers.
  - For file format, please refer to the configuration file preamble.
  - hdwr_svr must be restarted for changes to take effect.
  - This file is designed to ease the process of performance tuning.
  - It should not be changed on a whim.
- /opt/hsc/data/hdwr_svr.allow
  - This file is optional and contains IP addresses that hdwr_svr should allow connections from, one per line.
  - hdwr_svr must be restarted for changes to take effect.
  - This file is used to override the HMC security rule that says that only local clients may connect.
  - It should not be used on customer systems.

B.5 Log files

The following are the switch-related log files:

- Switch related'

  Check the SNI adapter driver log files located in /var/adm/sni/sni_errpt_capture.

- AIX

  Use the error-reporting tools (errpt).

- RSCT

  The RSCT-related log files can be found in the /var/ct directory.

  - HATS log files can be found in /var/ct/<RPD name>/log/cthats.
  - HAGS and HAGSGLSM log files can be found in /var/ct/<RPD name>/log/cthags and /var/log/messages.
  - For startup problems, see the /var/ct/<RM name>.stdout and /var/ct/<RM name>.stderr directories.

B.6 Commands

The following are various utilities we used to set up the switch in our environment:

- Service Processor settings

  - Setting Fast IPL

    i. From the main menu, select option 2 (System Power Control menu).

    ii. From System Power Control Menu, under option 6, the current IPL Speed is listed.

    iii. - Select option 6 to toggle the IPL speed between Fast and Slow.

- Switch related

  Error summary:

  ```
  lssrc -ls nrd
  errpt -N sni
  ```

- RSCT related

  - **lsclcfg**

    To list active peer domain (online or IW), cluster ID (each cluster has a unique cluster id), and local node number (used by HATS/HAGS).

  - **lsnodeid**

    To list node ID of the local node. Each node has a unique node ID.

  - **lsrpdomain**

    To list domain status: OpState, RSCTActiveVersion, MixedVersions.

    - RSCTActiveVersion is the RSCT version active in the domain.
    - OpState values are Online, Offline, Pending online, Pending offline.

  - **lsrpnode**

    - Use **-p** option to list local definition (may be out of sync if the node is offline).
    - To list node status: OpState, RSCTVersion:
      - RSCTActiveVersion is the RSCT version active in the domain.

  - OpState values are Online, Offline, Pending online, Pending offline.

- **lscomg**
  - Use **-i** option to list network interfaces for a communication group.
  - To list communication groups in online domain.
- **chcomg**

  To change a group's membership.
- **lsrsrc IBM.NetworkInterface**
  - To view interfaces in peer domain:

    ```
    export CT_MANAGEMENT_SCOPE = 2
    ```
  - To view interfaces on local node:

    ```
    export CT_MANAGEMENT_SCOPE = 0
    ```
- **lsrsrc IBM.RSCTParameters**

  To list HATS/HAGS tunables.
- **lsrsrc -t IBM.PeerNode**

  ```
  lsrsrc -t IBM.PeerNode Name NodeList NodeIDs OpState RSCTVersion
  ```

  To list node names, node numbers (NodeList attribute), node IDs, OpState and RSCTVersion:
  - RSCTVersion is 2.2.1.20 for AIX 5.1F installed on the node.
  - RSCTVersion is 2.3.1. for AIX 5.2 installed on the node.
- **lssrc -ls IBM.ConfigRM**
  - Shows start time.
  - Shows config version (must match on all nodes).
  - Shows providers (should equal the number of nodes online).
- **lssrc**

  ```
  lssrc -ls cthats
  lssrc -ls cthags
  lssrc -ls cthagsglsm
  ```
  - Similar information to on PSSP non-ct prefaced.
  - To show membership.
- **hagsvote -s cthags**

  To display group leader.
- **nlssrc -c -ls cthagsglsm**

  Provides a different output that is more detailed yet easier to read.
- **ctsnap**

  This command gathers configuration, log, and trace information about RSCT components.

Abbreviations and acronyms

AMD	Air Moving Device. See MDA,
API	Application Programming Interface
ARP	Address Resolution Protocol
BPA	Bulk Power Assembly
BPC	Bulk Power Controller
BPR	Bulk Power Regulator
CEC	Central Electronics Complex.
CSM	Cluster Systems Management
CSS	Communication Support Subsystem (switch)
CUoD	Capacity Upgrade on Demand
DCA-F	Distributed Conversion Assembly for Federation
ESS	Enterprise Storage Server
FNM	Federation Network Manager
FRU	Field Replaceable Unit
GFW	General Firmware (Regatta)
GPFS	General Parallel File System (also known as multimedia filesystem)
HACMP	High Availability Cluster Multi Processing
HMC	Hardware Maintenance Console
HPS	The shortest name for “IBM eServer pSeries High Performance Switch”. This is also known as the “pSeries HPS.”
HSC	Hardware Service Console (same as HMC)
IBM	International Business Machines Corporation
IP	Internet Protocol
IPAT	IP Address Takeover
ISB	Intermediate Switch Board
ITSO	International Technical Support Organization
K-LAPI	Kernel LAPI
LAPI	Low level Application Programming Interface
LPAR	Logical partition
MCM	Multi Chip Module
MDA	Motor Drive Assembly.
MDS	Microcode Descovery Service
MIF	Microcode Image File.
ML	maintenance level, used for defining a set of PTFs installed on the AIX machine
MPI	Message Passing Interface
MPM	Mechanism Pluggable Module
NIC	Network Interface Card
NIM	Network Install Manager
NSB	Node Switch Board
NUMA	Non-Uniform Memory Access
OS	Operating system
PE	Parallel Environment
POST	Power On Self Test
POWER	Performance Optimization With Enhanced RISC
PPC&C	Product Packaging, Power and Cooling.
PSSP	Parallel Systems Support Program
PTF	Program Temporary Fix
RIO	Remote I/O
RMC	Resource Monitoring and Control
RPA	RS/6000 Platform Architecture
RPD	RSCT Peer Domain
RSCT	Reliable Scalable Cluster Technology
RTAS	Runtime Abstarction Services
RVSD	Recoverable Virtual Shared Disk
SMA	Shared Memory Adapter
SNI	Switch Network Interface
SP	System Parallel
SPCC	Switch Port Connector Cards
SPCN	System Power Control Network
TCP	Transmission Control Protocol
TOD	Time of Day
UDP	Universal Datagram Protocol
VMM	Virtual Memory Manager (part of kernel subsystem in AIX)
VPD	Vital Product Data
VSD	Virtual Shared Disk

Glossary

ASCI. Accelerated Strategic Computing Initiative. This is a supercomputing project funded by the U.S. government for advanced computing technology.

BIST. Built In Self Test. Similar to and preceding Power On Self Test (POST).

BPA. Bulk Power Assembly. This is one side of the BPE and contains BPRs, BPCs, and BPDs. The BPA can provide up to 37 UPIC connections and 19.5 KW of power.

BPC. Bulk Power Controller. Distributes BPR 350 V to MDA 1-4 (2A SCB) and DCA 1-3 (9.2A SCB). Communicates with CSP via SPCN interface and with I/O drawer via RS-485 interface. One per BPA is required.

BPD. Bulk Power Distributor. Distributes BPR 350 V to DCA through 9.2A Static Circuit Breakers (SCB) (x10) Controlled by BPC. Total 10 UPIC ports to DCA or I/O drawers (some ports are not used). While there are three slots, zero to two per BPA are supported, with unused slots filled with air baffles.

BPE. Bulk Power Enclosure. This is the entire power conversion and distribution enclosure.

BPF. Bulk Power Fan (airmover). This is the turbine that sits to the right of most components in the BPA and forces cooling air through the power subsystem.

BPR. Bulk Power Regulator. Generates -350 V DC at 18.5A (6.5 KW) from 200-480 V AC input 3 phase. Each BPA may have from one to three BPRs. Each BPR may power two MCMs or four I/O drawers plus one IBF.

CAP. Capacitor. CAP cards provides low-frequency to mid-frequency decoupling. The number of CAP cards is determined by the number of DCAs.(one CAP card per three DCAs). Plugs into the Power Distribution backplane.

CEC. Central Electronics Complex. The compute portion of the server electronics, essentially encompassing the processors and memory. Other subsystems generally considered distinct from the CEC are the I/O subsystem and power and cooling.

CHRP. Common Hardware Reference Platform. This is a set of standards for PCI-based POWER architecture systems.

CSP. Converged Service Processor, mostly converged iSeries™ and pSeries service processor hardware, somewhat converged iSeries and pSeries service processor software.

CUoD. Capacity Upgrade on Demand. Also known as COD. The ability to place more components into a system than are licensed on the premise so that they may be activated at a later time without having to wait for installation. Generally, CUoD components may also provide additional fault tolerance by activating one of the idle components to compensate for a failed but licensed component.

DCA. Distributed Converter Assembly. Converts -350 V DC from BPRs to low DC voltages used by CEC hardware. Minimally, one DCA per MCM is required. Plugs into the power distribution backplane.

EEPROM. Electrically Erasable, Programmable Read Only Memory. A device used to persistently store a small amount of engineering data or VPD. Most IIC devices are 256 bytes. Smart chip EEPROMs are a few kilobytes.

ESW. Engineering Software. Development group that produces the majority of the code for the service processor as well as the firmware for the pSeries platforms. Also used to refer to that group's supported code.

Federation. This is the development code name used for the HPS. Normally code names are not supposed to be provided to customers. However, some components of the switch still use “Fed” or “Federation”.

Flash. EEPROM or the process of updating the contents of an EEPROM.

FNM. Fabric Network Manager. This is the underlying name of the daemon and tools used to set up and initialize an HPS network. This previously was Federation Network Manager.

Frame. The cabinet, and in context of firmware, the power subsystem.

GFW. Global Firmware (see also Global Open Firmware) .

Global Open Firmware. A portion of system firmware that communicates with SPCN, RTAS, CSP and the rest of system firmware.

HPS. High Performance Switch. IBM’s new shared memory switching technology..

Hypervisor. A portion of system firmware used to coordinate program requests among SMAs and LPARs.

IBF. Integrated Battery Feature. 2EIA, 400 V to cover a brief loss of power (7 KW for 30 seconds, 3.5 KW for five minutes). Zero or one per BPR.

IMG. Image file containing system (CEC or CSP) firmware. This contains only the GFW and not power/frame code.

IPL. Initial Program Load. Also known as BootStrapping or Booting.

IPLROS. Initial Program Load Read Only Storage. Another name for system firmware.

LCD. Liquid Crystal Display. Generally used to replace the LED display in the system front panel. Now showing two lines of 16 characters. Status and error codes are displayed during startup.

LED. Light Emitting Diode. Usually this refers to either a single indicator lamp or the system LCD display panel.

LLFW. Low-level firmware. First firmware component that executes on the host CPU. Performs I/O subsystem configuration, for example RIO, PCI. Passes control to Open Firmware.

Local Open Firmware. The portion of system firmware used for IPL.

LPAR. Logical Partition. This is a group of resources combined to allow an operating system instance to run on a subset of system hardware.

MDA. Motor Drive Assembly. Basically a standardized UPIC connected fan.

MOPS. Manual Operations. Name for the large body of ESW code responsible for the initialization of the CEC hardware.

NVRAM. Nonvolatile Random Access Memory. Packaged on the CSP, accessible from the host and from the CSP, battery powered during loss of external power source.

Open Firmware (also OF and O/F) Forth-based firmware that contains device drivers and boot manager. OF builds the device tree and loads the (AIX or other) operating system.

Partition Manager. This is the part of system firmware that handles partitioning resources.

PRD. Processor Run-Time Diagnostics. Component of service processor software that analyzes CEC errors (checkstops, recoverable errors and special unrecoverable errors).

pSeries HPS. See HPS.

RIO. Remote I/O. A bus and cabling mechanism that allows attachment of I/O to CECs and provides good I/O bandwidth. The problem is notable because, for large servers, the I/O is generally packaged in drawers and towers that are physically separated from the CEC, often by several feet. Good I/O bandwidth over cables of this length is a challenge that RIO addresses.

RS232. Serial connection used for the CSP/CEC-to-HMC connection.

RS422. Serial connection used within UPICs.

RS485. Serial connection encapsulated within the power cables from the BPA to various powered components.

RTAS. Run-time Abstraction Services. A hardware abstraction layer provided for the operating system (AIX) for each platform by ESW. RTAS is defined in RS/6000 Platform Architecture (RPA).

SCB. Static Circuit Breaker. This is provided as a power source to each UPIC port on a BPC or BPD.

SMA. Shared Memory Adapter. Obsolete name for an SNI.

SMP. Symmetric Multiprocessor. Flat multi-processor system where each processor has equal access to all of memory, I/O, and interrupts.

SNI. Switch Network Interface. This refers to the GX card or book providing HPS link pairs to a pSeries eServer.

SNM. Switch Network Manager. This is the GUI version of the FNM provided inside the HMC.

SPCN. Serial Power Control Network. This is a network of power and serial run from the power subsystem to individual power components within the frame.

System firmware. (also firmware) A collection of ESW components that execute on the host CPU. Generally, firmware is split into three pieces: low-level firmware, Open Firmware and RTAS. On a Regatta system, may be updated from AIX or via diskette from service processor menus by users, or via the primary I/O debug port by development.

TOD. Time of Day. This is a critical synchronization component for HPS startup.

UEPO. Universal Emergency Power Off. Provides the external switch that shuts off all power in the Regatta system. The green button is for “start of service” and the white button is for “end of service”. The panel is connected to the two BPCs via two interface connectors on the card. The panel can be concurrently replaced by forcing the BPCs into UEPO Bypass mode via the small slide switches on the face of the BPCs. The service pushbuttons are used for service personnel to repair the power subsystem. If the system is off when the service complete button is pressed, the system will power on.

UPIC. Universal Power Interface Cable. This provides power and RS485 connections to each powered device.

VAC. Volts Alternating Current. This is a measure of peak electrical difference from earth ground. Alternating current ideally cycles in a sine-wave pattern with peak and inverse peak being equal voltage difference from earth ground.

VDC. Volts Direct Current. This is a measure of electrical difference from earth ground. Direct current maintains the same voltage over time.

VPD. Vital Product Data. Refers to two different concepts: 1. IBM corporate standard for FRU identification. 2. iSeries and pSeries engineering data stored in EEPROMS. These EEPROMS are accessible either through the IIC bus or a serial bus.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 250. Note that some of the documents referenced here may be available in softcopy only.

- *Linux HPC Cluster Installation,* SG24-6041
- *An Introduction to CSM 1.3 for AIX 5L,* SG24-6859
- *Deploying Oracle 9i RAC on IBM IBM Eserver Cluster 1600 with GPFS*, SG24-6954
- *RS/6000 SP Cluster: The Path to Universal Clustering*, SG24-5374
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615

Other publications

These publications are also relevant as further information sources:

- *Switch Network Interface for eServer pSeries High Performance Switch Guide and Reference,* SC23-4869
- *pSeries High Performance Switch Planning, Installation and Service,* GA22-7951
- *eServer Hardware Management Console for pSeries Maintenance Guide,* SA38-0603
- *pSeries 690 Service Guide*, SA38-0589
- *pSeries High Performance Switch Planning, Installation and Service*, GA22- 7951
- *RSCT Group Services Programming Guide and Reference*, SA22-7888
- *IBM Reliable Scalable Cluster Technology for AIX 5L, Administration Guide*, SA22-7889
- *RSCT for AIX 5L: Technical Reference*, SA22-7890
- *RSCT for AIX 5L: Messages*, SA22-7891
- *IBM Reliable Scalable Cluster Technology for Linux, RSCT Guide and Reference,* SA22-7892-01
- *RS/6000 and @server pSeries Adapter Placement Reference for AIX*, SA38-0538
- *Switch Network Interface for eServer pSeries High Performance Switch Guide and Reference,* SC23-4869
- *IBM Cluster Systems Management for AIX 5L Planning and Installation Guide,* SA22-7919
- *AIX 5L Version 5.2 Commands Reference, Volume 4: n through r,* SC23-4118-06
- *Network Installation Management Guide and Reference,* SC23-4385

Online resources

These Web sites and URLs are also relevant as further information sources:

- FIX Delivery Center for AIX provides fixies for the operating system, Java and compilers

  http://techsupport.services.ibm.com/server/aix.fdc

- Microcode download location:

  http://techsupport.services.ibm.com/server/mdownload2

- Regatta firmware download location:

  http://techsupport.services.ibm.com/server/mdownload2/7040681F.html

- HMC microcode download location:

  https://techsupport.services.ibm.com/server/hmc

- Microcode Discovery Service information:

  http://techsupport.services.ibm.com/server/aix.invscoutMDS

- Vital Product Data capture services location:

  http://techsupport.services.ibm.com/server/aix.invscoutVPD

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

An Introduction to the New IBM @server pSeries High Performance Switch

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

An Introduction to the New IBM eserver pSeries High Performance Switch

Installation and configuration of the IBM eserver HPS

Considerations for configuring applications for the IBM HPS

Experiences with VSD, GPFS and HACMP running over HPS

This IBM Redbook contains information about the first official release of the pSeries High Performance Switch (HPS) and products that may benefit from this equipment. This book includes detailed information about the hardware and software configuration as well as examples of supported and potential configurations.

This redbook will help you install, tailor and configure the new switch on IBM's premier UNIX operating system, Advanced Interactive eXecutive (AIX) 5.2, and shows the differences from earlier generations of similar technology, previously known collectively as the Scalable POWERparallel (SP) Switch. Additionally covered are topics such as Global Parallel File System (GPFS) and High Availability/Cluster Multi-Processing (HACMP).

This redbook gives a broad understanding of this new architecture and its dependencies on the pSeries Hardware Management Console (HMC) and RISC System Cluster Technology (RSCT).

This redbook will help you design and create a solution to exploit the power and performance of the initial release of this new switch and to plan for future generations.

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

SG24-6978-00 ISBN 0738499005