IBM Power Systems 775 for AIX and Linux HPC Solution

Unleashes computing power for HPC workloads

Provides architectural solution overview

Contains sample scenarios

Dino Quintero
Kerry Bosworth
Puneet Chaudhary
Rodrigo Garcia da Silva
ByungUn Ha
Jose Higino
Marc-Eric Kahle
Tsuyoshi Kamenoue
James Pearson
Mark Perez
Fernando Pizzano
Robert Simon
Kai Sun

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Power Systems 775 for AIX and Linux HPC Solution

October 2012

SG24-8003-00

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

First Edition (October 2012)

This edition applies to IBM AIX 7.1, xCAT 2.6.6, IBM GPFS 3.4, IBM LoadLelever, Parallel Environment Runtime Edition for AIX V1.1.

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX 5L™
AIX®
BladeCenter®
DB2®
developerWorks®
Electronic Service Agent™
Focal Point™
Global Technology Services®
GPFS™
HACMP™
IBM®
LoadLeveler®
Power Systems™
POWER6+™
POWER6®
POWER7®
PowerPC®
POWER®
pSeries®
Redbooks®
Redbooks (logo) ®
RS/6000®
System p®
System x®
Tivoli®

The following terms are trademarks of other companies:

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

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

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

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

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

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

Preface

This IBM® Redbooks® publication contains information about the IBM Power Systems™ 775 Supercomputer solution for AIX® and Linux HPC customers. This publication provides details about how to plan, configure, maintain, and run HPC workloads in this environment.

This IBM Redbooks document is targeted to current and future users of the IBM Power Systems 775 Supercomputer (consultants, IT architects, support staff, and IT specialists) responsible for delivering and implementing IBM Power Systems 775 clustering solutions for their enterprise high-performance computing (HPC) applications.

The team who wrote this book

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

Dino Quintero is an IBM Senior Certified IT Specialist with the ITSO in Poughkeepsie, NY. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is currently an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Kerry Bosworth is a Software Engineer in pSeries® Cluster System Test for high-performance computing in Poughkeepsie, New York. Since joining the team four years ago, she worked with the InfiniBand technology on POWER6® AIX, SLES, and Red Hat clusters and the new Power 775 system. She has 12 years of experience at IBM with eight years in IBM Global Services as an AIX Administrator and Service Delivery Manager.

Puneet Chaudhary is a software test specialist with the General Parallel File System team in Poughkeepsie, New York.

Rodrigo Garcia da Silva is a Deep Computing Client Technical Architect at the IBM Systems and Technology Group. He is part of the STG Growth Initiatives Technical Sales Team in Brazil, specializing in High Performance Computing solutions. He has worked at IBM for the past five years and has a total of eight years of experience in the IT industry. He holds a B.S. in Electrical Engineering and his areas of expertise include systems architecture, OS provisioning, Linux, and open source software. He also has a background in intellectual property protection, including publications and a filed patent.

ByungUn Ha is an Accredited IT Specialist and Deep Computing Technical Specialist in Korea. He has over 10 years experience in IBM and has conducted various HPC projects and HPC benchmarks in Korea. He has supported Supercomputing Center at KISTI (Korea Institute of Science and Technology Information) on-site for nine years. His area of expertise include Linux performance and clustering for System X, InfiniBand, AIX Power system, and HPC Software Stack including LoadLeveler®, Parallel Environment, and ESSL/PESSL, C/Fortran Compiler. He is a Redhat Certified Engineer (RHCE) and has a Master's degree in Aerospace Engineering from Seoul National University. He is currently working in Deep Computing team, Growth Initiatives, STG in Korea as a HPC Technical Sales Specialist.

Jose Higino is an Infrastructure IT Specialist for AIX/Linux support and services for IBM Portugal. His areas of knowledge include System X, BladeCenter® and Power Systems planning and implementation, management, virtualization, consolidation, and clustering (HPC and HA) solutions. He is currently the only person responsible for Linux support and services in IBM Portugal. He completed the Red Hat Certified Technician level in 2007, became a CiRBA Certified Virtualization Analyst in 2009, and completed certification in KT Resolve methodology as an SME in 2011. José holds a Master of Computers and Electronics Engineering degree from UNL - FCT (Universidade Nova de Lisboa - Faculdade de Ciências e Technologia), in Portugal.

Marc-Eric Kahle is a POWER® Systems Hardware Support specialist at the IBM Global Technology Services® Central Region Hardware EMEA Back Office in Ehningen, Germany. He has worked in the RS/6000®, POWER System, and AIX fields since 1993. He has worked at IBM Germany since 1987. His areas of expertise include POWER Systems hardware and he is an AIX certified specialist. He has participated in the development of six other IBM Redbooks publications.

Tsuyoshi Kamenoue is a Advisory IT specialist in Power Systems Technical Sales in IBM Japan. He has nine years of experience of working on pSeries, System p®, and Power Systems products especially in HPC area. He holds a Bachelor's degree in System information from the university of Tokyo.

James Pearson is a Product Engineer for pSeries high-end Enterprise systems and HPC cluster offerings since 1998. He has participated in the planning, test, installation and on-going maintenance phases of clustered RISC and pSeries servers for numerous government and commercial customers, beginning with SP2 and continuing through the current Power 775 HPC solution.

Mark Perez is a customer support specialist servicing IBM Cluster 1600.

Fernando Pizzano is a Hardware and Software Bring-up Team Lead in the IBM Advanced Clustering Technology Development Lab, Poughkeepsie, New York. He has over 10 years of information technology experience, the last five years in HPC Development. His areas of expertise include AIX, pSeries High Performance Switch, and IBM System p hardware. He holds an IBM certification in pSeries AIX 5L™ System Support.

Robert Simon is a Senior Software Engineer in STG working in Poughkeepsie, New York. He has worked with IBM since 1987. He currently is a Team Leader in the Software Technical Support Group, which supports the High Performance Clustering software (LoadLeveler, CSM, GPFS™, RSCT, and PPE). He has extensive experience with IBM System p hardware, AIX, HACMP™, and high-performance clustering software. He has participated in the development of three other IBM Redbooks publications.

Kai Sun is a Software Engineer in pSeries Cluster System Test for high performance computing in IBM China System Technology Laboratory, Beijing. Since joining the team in 2011, he has worked with the IBM Power Systems 775 cluster. He has six years of experience at embedded system on Linux and VxWorks platform. He has recently been given an Eminence and Excellence Award by IBM for his work on Power Systems 775 cluster. He holds a B.Eng. degree in Communication Engineering from Beijing University of Technology, China. He has a M.Sc. degree in Project Management from the New Jersey Institute of Technology, US.

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

- Mark Atkins
  IBM Boulder
- Robert Dandar

- Joseph Demczar
- Chulho Kim
- John Lewars
- John Robb
- Hanhong Xue
- Gary Mincher
- Dave Wootton
- Paula Trimble
- William Lepera
- Joan McComb
- Bruce Potter
- Linda Mellor
- Alison White
- Richard Rosenthal
- Gordon McPheeters
- Ray Longi
- Alan Benner
- Lissa Valleta
- John Lemek
- Doug Szerdi
- David Lerma

IBM Poughkeepsie

- Ettore Tiotto

IBM Toronto, Canada

- Wei QQ Qu

IBM China

- Phil Sanders

IBM Rochester

- Richard Conway
- David Bennin

International Technical Support Organization, Poughkeepsie Center

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

http://www.ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  `http://www.ibm.com/redbooks`

- Send your comments in an email to:

  `redbooks@us.ibm.com`

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  `http://www.facebook.com/IBMRedbooks`

- Follow us on Twitter:

  `http://twitter.com/ibmredbooks`

- Look for us on LinkedIn:

  `http://www.linkedin.com/groups?home=&gid=2130806`

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  `https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm`

- Stay current on recent Redbooks publications with RSS Feeds:

  `http://www.redbooks.ibm.com/rss.html`

1

Understanding the IBM Power Systems 775 Cluster

In this book, we describe the new IBM Power Systems 775 Cluster hardware and software. The chapters provide an overview of the general features of the Power 775 and its hardware and software components. This chapter helps you get a basic understanding and concept of this cluster.

Application integration and monitoring of a Power 775 cluster is also described in greater detail in this IBM Redbooks publication. LoadLeveler, GPFS, xCAT, and more are documented with some examples to get a better view on the complete cluster solution.

Problem determination is also discussed throughout this publication for different scenarios that include xCAT configuration issues, Integrated Switch Network Manager (ISNM), Host Fabric Interface (HFI), GPFS, and LoadLeveler. These scenarios show the flow of how to determine the cause of the error and how to solve the error. This knowledge compliments the information in Chapter 5, “Maintenance and serviceability” on page 265.

Some cluster management challenges might need intervention that requires service updates, xCAT shutdown/startup, node management, and Fail in Place tasks. Documents that are available are referenced in this book because not everything is shown in this publication.

This chapter includes the following topics:

- Overview of the IBM Power System 775 Supercomputer
- Advantages and new features of the IBM Power 775
- Hardware information
- Power, packaging, and cooling
- Disk enclosure
- Cluster management
- Connection scenario between EMS, HMC, and Frame
- High Performance Computing software stack

1.1 Overview of the IBM Power System 775 Supercomputer

For many years, IBM provided High Performance Computing (HPC) solutions that provide extreme performance. For example, highly scalable clusters by using AIX and Linux for demanding workloads, including weather forecasting and climate modeling.

The previous IBM Power 575 POWER6 water-cooled cluster showed impressive density and performance. With 32 processors, 32 GB to 256 GB of memory in one central electronic complex (CEC) enclosure or cage, and up to 14 CECs per Frame (water-cooled), 448 processors per frame was possible.
The InfiniBand interconnect provided the cluster with powerful communication channels for the workloads.

The new Power 775 Supercomputer from IBM takes the density to a new height. With 256 3.84 GHz POWER7® processors, 2 TB of memory per CEC, and up to 12 CECs per Frame, a total of 3072 processors and 24 TBs memory per Frame is possible. Highly scalable with the capability to cluster 2048 CEC drawers together makes up 524,288 POWER7 processors to do the work to solve the most challenging problems. A total of 7.86 TF per CEC and 94.4 TF per rack highlights the capabilities of this high-performance computing solution.

The hardware is only as good as the software that runs on it. IBM AIX, IBM FileNet Process Engine (PE) Runtime Edition, LoadLeveler, GPFS, and xCAT are a few of the supported software stacks for the solution. For more information, see 1.9, “High Performance Computing software stack” on page 62.

1.2 The IBM Power 775 cluster components

The IBM Power 775 can consist of the following components:

- Compute subsystem:
  - Diskless nodes dedicated to perform computational tasks
  - Customized operating system (OS) images
  - Applications
- Storage subsystem:
  - I/O node (diskless)
  - OS images for IO nodes
  - SAS adapters attached to the Disk Enclosures (DE)
  - General Parallel File System (GPFS)
- Management subsystem:
  - Executive Management Server (EMS)
  - Login Node
  - Utility Node
- Communication Subsystem:
  - Host Fabric Interface (HFI):
    - Busses from processor modules to the switching hub in an octant
    - Local links (LL-links) between octants
    - Local remote links (LR-links) between drawers in a SuperNode
    - Distance links (D-links) between SuperNodes
  - Operating system drivers
  - IBM User space protocol
  - AIX and Linux IP drivers

Octants, SuperNode, and other components are described in the other sections of this book.

- Node types

  The following node types have other partial functions available for the cluster. In the context of the 9125-F2C drawer, a node is an OSI image that is booted in an LPAR. There are three general designations for node types on the 9125-F2C. Often these functions are dedicated to a node, but a node can have multiple roles:

  - Compute nodes

    Compute nodes run parallel jobs and perform the computational functions. These nodes are diskless and booted across the HFI network from a Service Node. Most of the nodes are compute nodes.

  - IO nodes

    These nodes are attached to either the Disk Enclosure in the physical cluster or external storage. These nodes serve the file system to the rest of the cluster.

  - Utility Nodes

    A Utility node offers services to the cluster. These nodes often feature more resources, such as an external Ethernet, external, or internal storage. The following Utility nodes are required:

    - Service nodes: Runs xCAT to serve the operating system to local diskless nodes
    - Login nodes: Provides a centralized login to the cluster

  - Optional utility node:

    - Tape subsystem server

Important: xCAT stores all system definitions as node objects, including the required EMS console and the HMC console. However, the consoles are external to the 9125-F2C cluster and are not referred to as cluster nodes. The HMC and EMS consoles are physically running on specific, dedicated servers. The HMC runs on a System x® based machine (7042 or 7310) and the EMS runs on a POWER 750 Server. For more information, see 1.7.1, “Hardware Management Console” on page 53 and 1.7.2, “Executive Management Server” on page 53.

For more information, see this website:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/p7had/p7had_775x.pdf

1.3 Advantages and new features of the IBM Power 775

The IBM Power Systems 775 (9125-F2C) has several new features that make this system even more reliable, available, and serviceable.

Fully redundant power, cooling and management, dynamic processor de-allocation and memory chip & lane sparing, and concurrent maintenance are the main reliability, availability, and serviceability (RAS) features.

The system is water-cooled, which gives a 100% heat capture. Some components are cooled by small fans, but the Rear Door Heat exchanger captures this heat.

Because most of the nodes are diskless nodes, the service nodes provide the operating system to the diskless nodes. The HFI network also is used to boot the diskless utility nodes.

The Power 775 Availability Plus (A+) feature allows processors, switching hubs, and HFI cables immediate failure-recovery because more resources are available in the system. These resources fail in place and no hardware must be replaced until a specified threshold is reached. For more information, see 5.4, “Power 775 Availability Plus” on page 297.

The IBM Power 775 cluster solution provides High Performance Computing clients with the following benefits:

- Sustained performance and low energy consumption for climate modeling and forecasting
- Massive scalability for cell and organism process analysis in life sciences
- Memory capacity for high-resolution simulations in nuclear resource management
- Space and energy efficient for risk analytics and real-time trading in financial services

1.4 Hardware information

This section provides detailed information about the hardware components of the IBM Power 775. Within this section, there are links to IBM manuals and external sources for more information.

1.4.1 POWER7 chip

The IBM Power System 775 implements the POWER7 processor technology. The PowerPC® Architecture POWER7 processor is designed for use in servers that provide solutions with large clustered systems, as shown in Figure 1-1 on page 5.

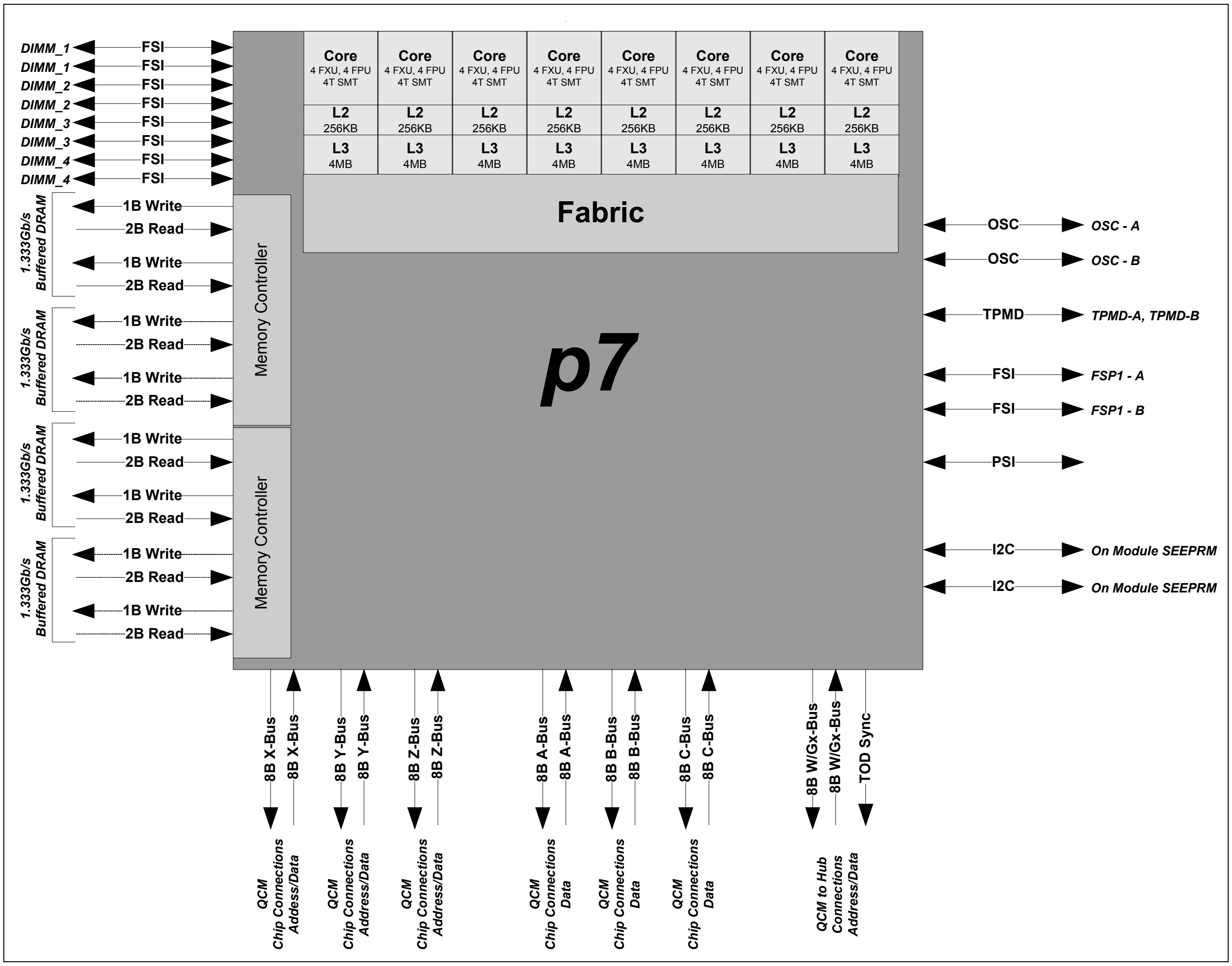


Figure 1-1 POWER7 chip block diagram

IBM POWER7 characteristics

This section provides a description of the following characteristics of the IBM POWER7 chip, as shown in Figure 1-1:

- 240 GFLOPs:
  - Up to eight cores per chip
  - Four Floating Point Units (FPU) per core
  - Two FLOPS/Cycle (Fused Operation)
  - 246 GFLOPs = 8 cores x 3.84 GHz x 4 FPU x 2)
- 32 KBs instruction and 32 KBs data caches per core
- 256 KB L2 cache per core
- 4 MB L3 cache per core
- Eight Channels of SuperNova buffered DIMMs:
  - Two memory controllers per chip
  - Four memory busses per memory controller (1 B wide Write, 2 B wide Read each)
- CMOS 12S SOI 11 level metal
- Die size: 567 mm2

Architecture

- PowerPC architecture
- IEEE New P754 floating point compliant
- Big endian, little endian, strong byte ordering support extension
- 46-bit real addressing, 68-bit virtual addressing
- Off-chip bandwidth: 336 GBps:
  - Local + remote interconnect)
- Memory capacity: Up to 128 GBs per chip
- Memory bandwidth: 128 GBps peak per chip

C1 core and cache

- 8 C1 processor cores per chip
- 2 FX, 2 LS, 4 DPFP, 1 BR, 1 CR, 1 VMX, 1 DFP
- 4 SMT, OoO
- 112x2 GPR and 172x2 VMX/VSX/FPR renames

PowerBus On-Chip Intraconnect

- 1.9 GHz Frequency
- (8) 16 B data bus, 2 address snoop, 21 on/off ramps
- Asynchronous interface to chiplets and off-chip interconnect

Differential memory controllers (2)

- 6.4-GHz Interface to Super Nova (SN)
- DDR3 support max 1067 Mhz
- Minimum Memory 2 channels, 1 SN/channel
- Maximum Memory 8 channels X 1 SN/channel
- 2 Ports/Super Nova
- 8 Ranks/Port
- X8b and X4b devices supported

PowerBus Off-Chip Interconnect

- 1.5 to 2.9 Gbps single ended EI-3
- 2 spare bits/bus
- Max 256-way SMP
- 32-way optimal scaling
- Four 8-B Intranode Buses (W, X, Y, or Z)
- All buses run at the same bit rate
- All capable of running as a single 4B interface; the location of the 4B interface within the 8 B is fixed
- Hub chip attaches via W, X, Y or Z
- Three 8-B Internode Buses (A, B,C)
- C-bus multiplex with GX Only operates as an aggregate data bus (for example, address and command traffic is not supported)

Buses

Table 1-1 describes the POWER7 busses.

Table 1-1 POWER7 busses

Bus name	Width (speed)	Connects	Function
W, X, Y, Z	8B+8B with 2 extra bits per bus (3 Gbps)	Intranode processors & hub	Used for address and data
A,B	8B+8B with 2 extra bits per bus (3 Gbps)	Other nodes within drawer	Data only
C	8B+8B with 2 extra bits per bus (3 Gb/p)	Other nodes within drawer	Data only, Multiplex with Gx
Mem1-Mem8	2B Read + 1B Write with 2 extra bits per bus (2.9 GHz)	Processor to memory	

WXYZABC Busses

The off-chip PowerBus supports up to seven coherent SMP links (WXYZABC) by using Elastic Interface 3 (EI-3) interface signaling that uses up to 3 Gbps. The intranode WXYZ links up to four processor chips to make a 32way and connect a Hub chip to each processor. The WXYZ links carry coherency traffic and data and are interchangeable as intranode processor links or Hub links. The internode AB links connect up to two nodes per processor chip. The AB links carry coherency traffic and data and are interchangeable with each other. The AB links also are configured as aggregate data-only links. The C link is configured only as a data-only link.

All seven coherent SMP links (WXYZABC) are configured as 8Bytes or 4Bytes in width.

The XYZABC Busses include the following features:

- Four (WXYZ) 8-B or 4-B EI-3 Intranode Links
- Two (AB) 8-B or 4-B EI-3 Internode Links or two (AB) 8-B or 4-B EI-3 data-only Links
- One (C) 8-B or 4-B EI-3 data-only Link

PowerBus

The PowerBus is responsible for coherent and non-coherent memory access, IO operations, interrupt communication, and system controller communication. The PowerBus provides all of the interfaces, buffering, and sequencing of command and data operations within the storage subsystem. The POWER7 chip has up to seven PowerBus links that are used to connect to other POWER7 chips, as shown in Figure 1-2 on page 8.

The PowerBus link is an 8-Byte-wide (or optional 4-Byte-wide), split-transaction, multiplexed, command and data bus that supports up to 32 POWER7 chips. The bus topology is a multitier, fully connected topology to reduce latency, increase redundancy, and improve concurrent maintenance. Reliability is improved with ECC on the external I/Os.

Data transactions are always sent along a unique point-to-point path. A route tag travels with the data to help routing decisions along the way. Multiple data links are supported between chips that are used to increase data bandwidth.

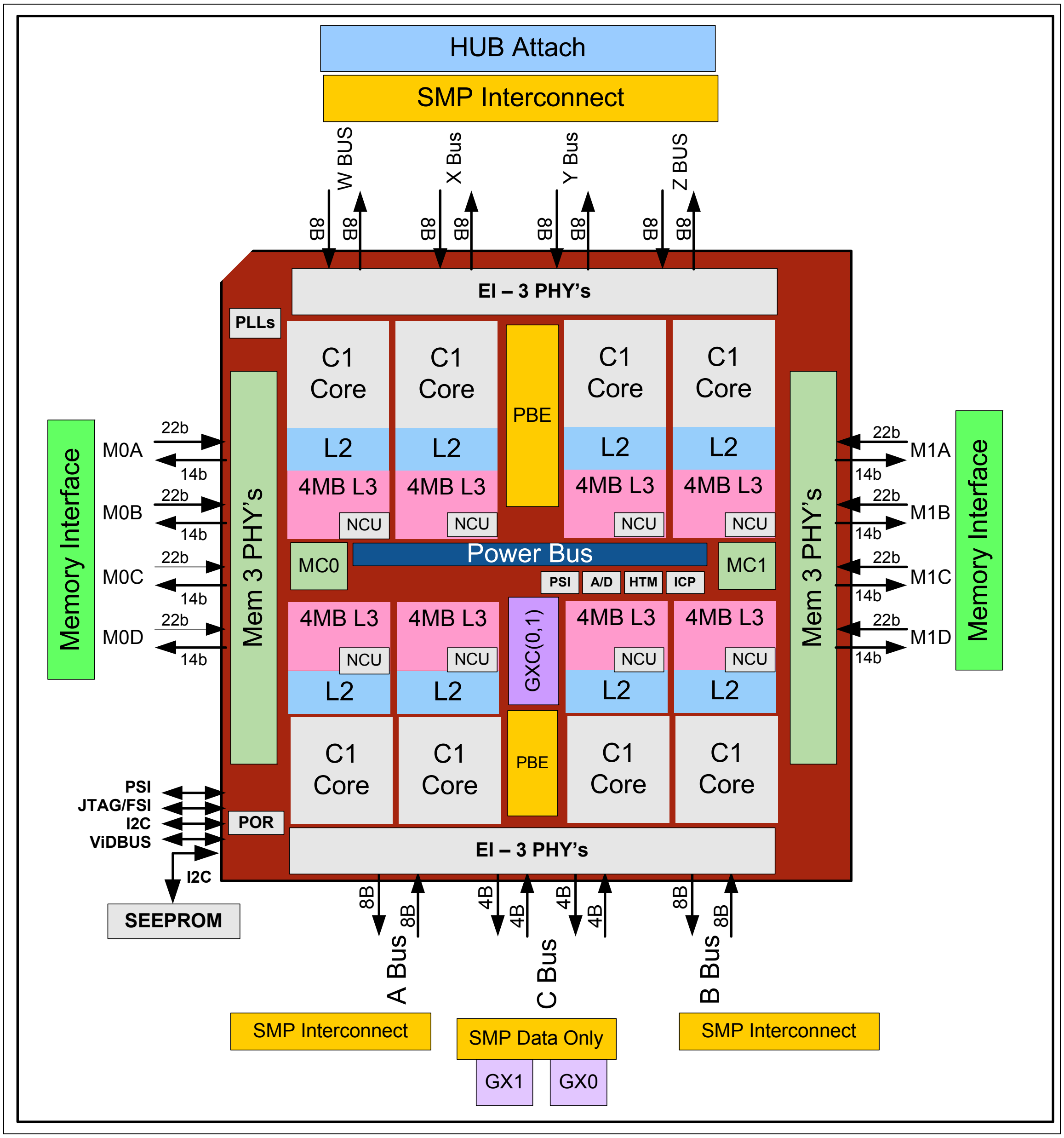


Figure 1-2 POWER7 chip layout

Figure 1-3 on page 9 shows the POWER7 core structure.

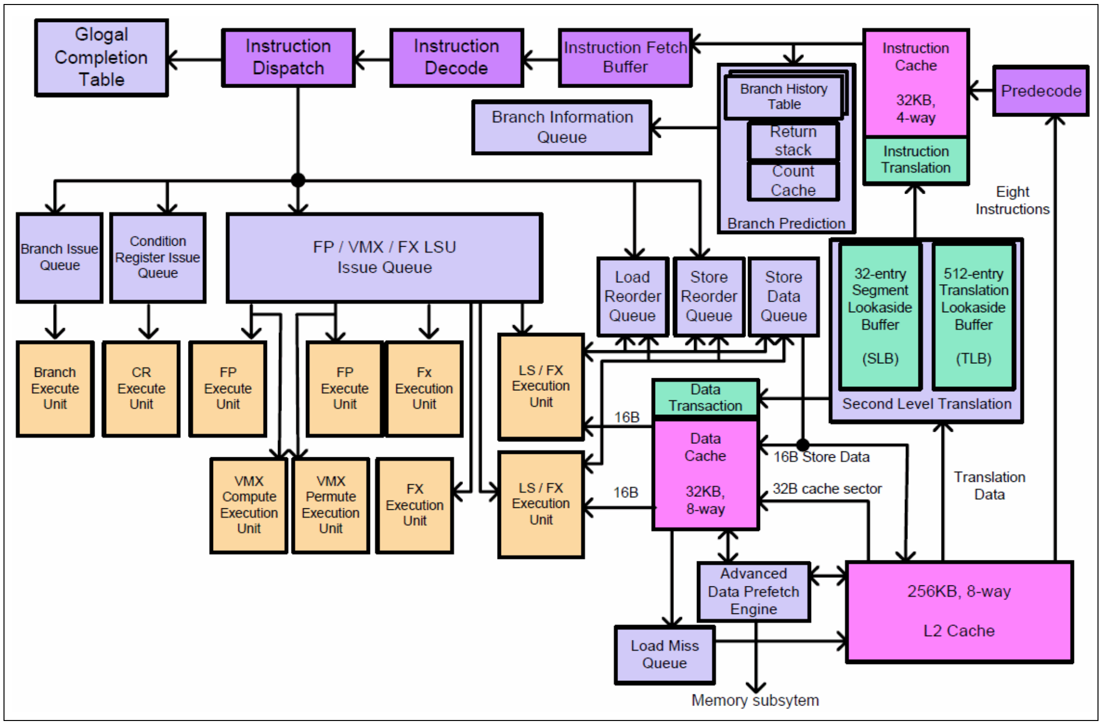


Figure 1-3 Microprocessor core structural diagram

Reliability, availability, and serviceability features

The microprocessor core includes the following reliability, availability, and serviceability (RAS) features:

- POWER7 core:
  - Instruction retry for soft core logic errors
  - Alternate processor recovery for hard core errors detected
  - Processor limited checkstop for other errors
  - Protection key support for AIX
- L1 I/D Cache Error Recovery and Handling:
  - Instruction retry for soft errors
  - Alternate processor recovery for hard errors
  - Guarding of core for core and L1/L2 cache errors
- L2 Cache:
  - ECC on L2 and directory tags
  - Line delete for L2 and directory tags (seven lines)
  - L2 UE handling includes purge and refetch of unmodified data
  - Predictive dynamic guarding of associated cores
- L3 Cache:
  - ECC on data
  - Line delete mechanism for data (seven lines)
  - L3UE handling includes purges and refetch of unmodified data
  - Predictive dynamic guarding of associated cores for CEs in L3 not managed by the line deletion

1.4.2 I/O hub chip

This section provides information about the IBM Power 775 I/O hub chip (or torrent chip), as shown in Figure 1-4.

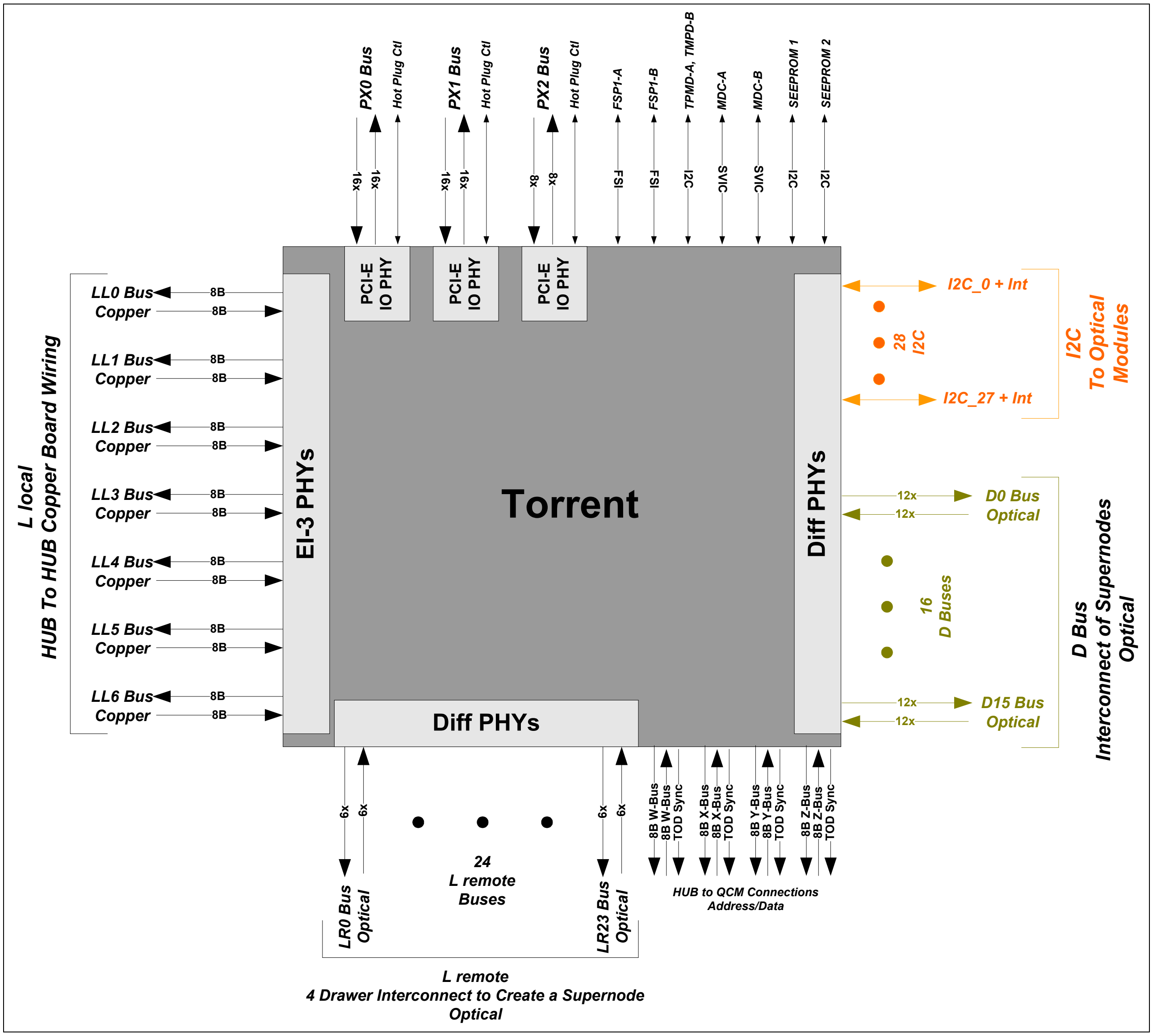


Figure 1-4 Hub chip (Torrent)

Host fabric interface

The host fabric interface (HFI) provides a non-coherent interface between a quad-chip module (QCM), which is composed of four POWER7, and the clustered network.

Figure 1-5 on page 11 shows two instances of HFI in a hub chip. The HFI chips also attach to the Collective Acceleration Unit (CAU).

Each HFI has one PowerBus command and four PowerBus data interfaces, which feature the following configuration:

1. The PowerBus directly connects to the processors and memory controllers of four POWER7 chips via the WXYZ links.

2. The PowerBus also indirectly coherently connects to other POWER7 chips within a 256-way drawer via the LL links. Although fully supported by the HFI hardware, this path provides reduced performance.
3. Each HFI has four ports to the Integrated Switch Router (ISR). The ISR connects to other hub chips through the D, LL, and LR links.
4. ISRs and D, LL, and LR links that interconnect the hub chips form the cluster network.

POWER7 chips: The set of four POWER7 chips (QCM), its associated memory, and a hub chip form the building block for cluster systems. A Power 775 systems consists of multiple building blocks that are connected to each other via the cluster network.

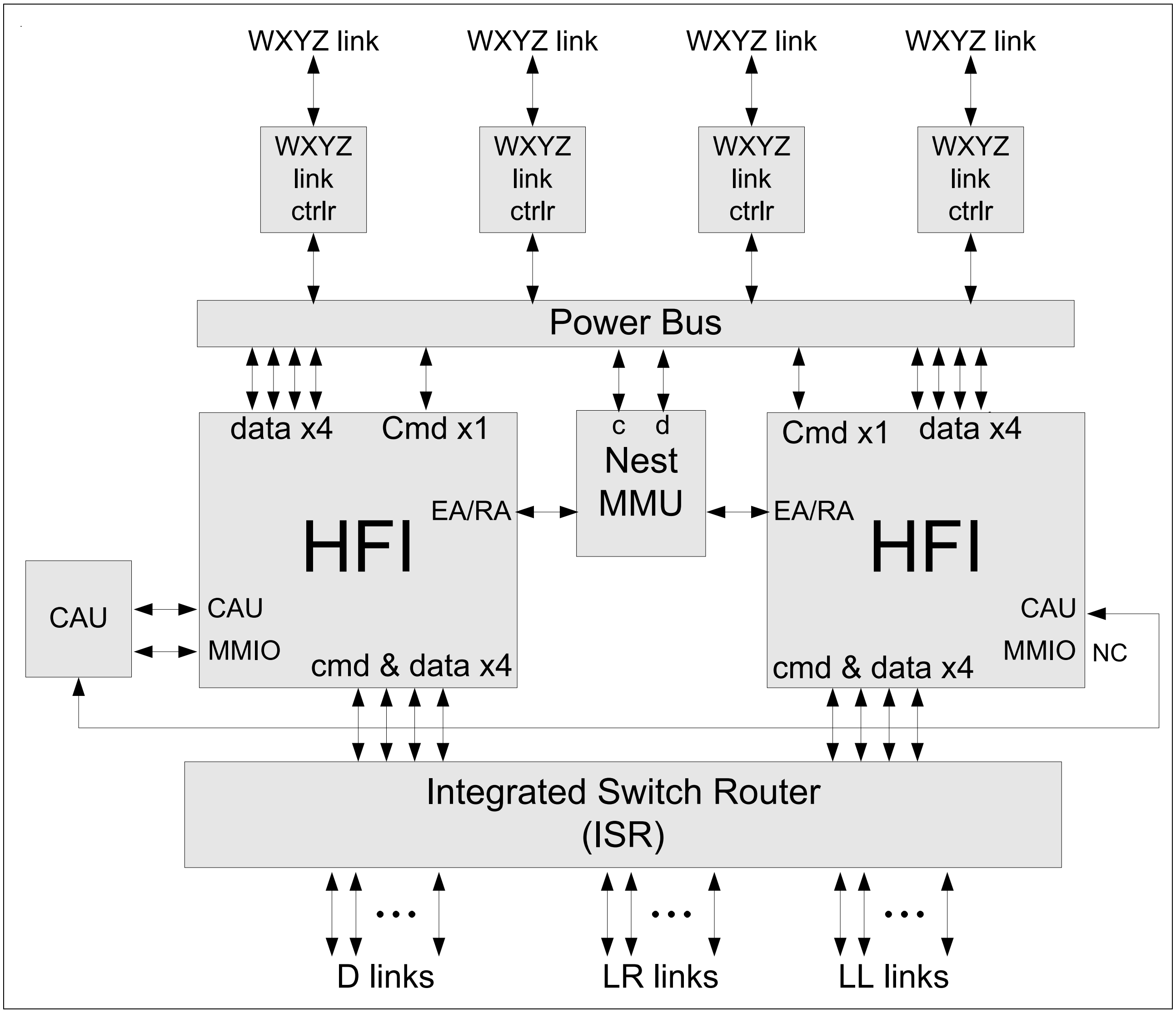


Figure 1-5 HFI attachment scheme

Packet processing

The HFI is the interface between the POWER7 chip quads and the cluster network, and is responsible for moving data between the PowerBus and the ISR. The data is in various formats, but packets are processed in the following manner:

- **Send**
  - Pulls or receives data from PowerBus-attached devices in a POWER7 chip
  - Translates data into network packets
  - Injects network packets into the cluster network via the ISR
- **Receive**
  - Receives network packets from the cluster network via the ISR
  - Translates them into transactions
  - Pushes the transactions to PowerBus-attached devices in a POWER7 chip
- **Packet ordering**
  - The HFIs and cluster network provide no ordering guarantees among packets. Packets that are sent from the same source window and node to the same destination window and node might reach the destination in a different order.

Figure 1-6 shows two HFIs cooperating to move data from devices that are attached to one PowerBus to devices attached to another PowerBus through the Cluster Network.

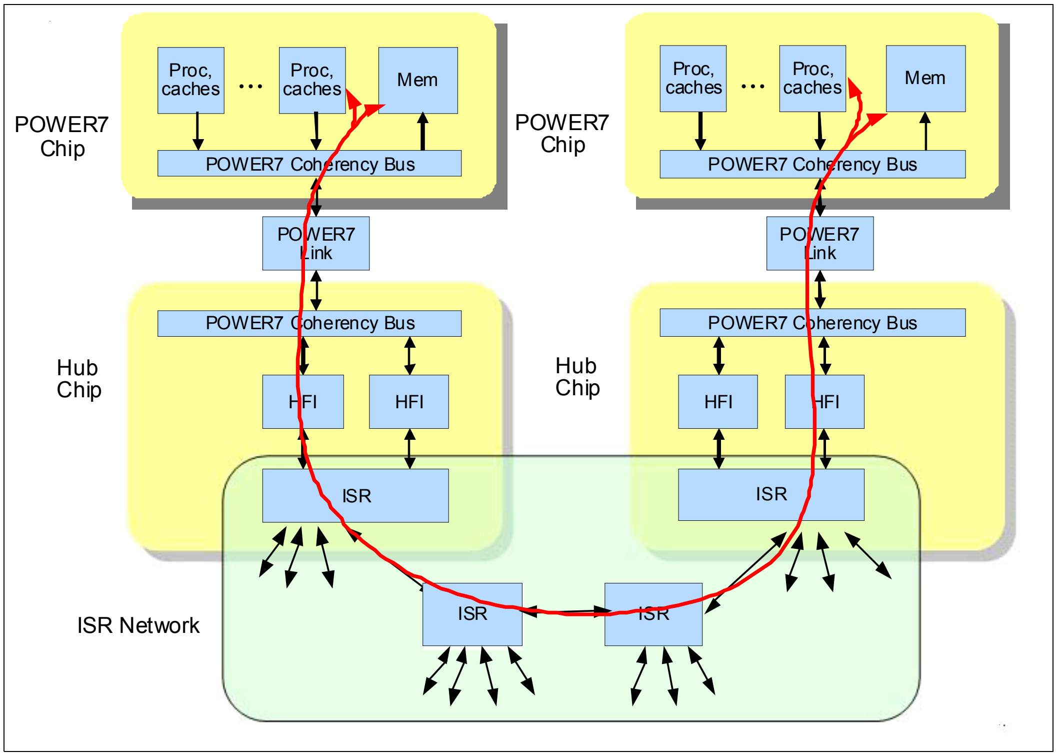


Figure 1-6 HFI moving data from one quad to another quad

HFI paths: The path between any two HFIs might be indirect, thus requiring multiple hops through intermediate ISRs.

1.4.3 Collective acceleration unit

The hub chip provides specialized hardware that is called the *Collective Acceleration Unit* (CAU) to accelerate frequently used collective operations.

Collective operations

Collective operations are distributed operations that operate across a tree. Many HPC applications perform collective operations with the application that make forward progress after every compute node that completed its contribution and after the results of the collective operation are delivered back to every compute node (for example, barrier synchronization, and global sum).

A specialized arithmetic-logic unit (ALU) within the collective CAU implements reduction, barrier, and reduction operations. For reduce operations, the ALU supports the following operations and data types:

- Fixed point: NOP, SUM, MIN, MAX, OR, ANDS, signed and unsigned XOR
- Floating point: MIN, MAX, SUM, single and double precision PROD

There is one CAU in each hub chip, which is one CAU per four POWER7 chips, or one CAU per 32 C1 cores.

Software organizes the CAUs in the system collective trees. The arrival of an input on one link causes its forwarding on all other links when there is a broadcast operation. For reduce operation, arrivals on all but one link causes the reduction result to forward to the remaining links.

A link in the CAU tree maps to a path composed of more than one link in the network. The system supports many trees simultaneously and each CAYU supports 64 independent trees.

The usage of sequence numbers and a retransmission protocol enables reliability and pipelining. Each tree has only one participating HFI window on any involved node. The order in which the reduction operation is evaluated is preserved from one run to another, which benefits programming models that allow programmers to require that collective operations are executed in a particular order, such as MPI.

Package propagation

As shown Figure 1-7 on page 14, a CAU receive packets from the following sources:

- The memory of a remote node is inserted into the cluster network by the HFI of the remote node
- The memory of a local node is inserted into the cluster network by the HFI of the local node
- A remote CAU

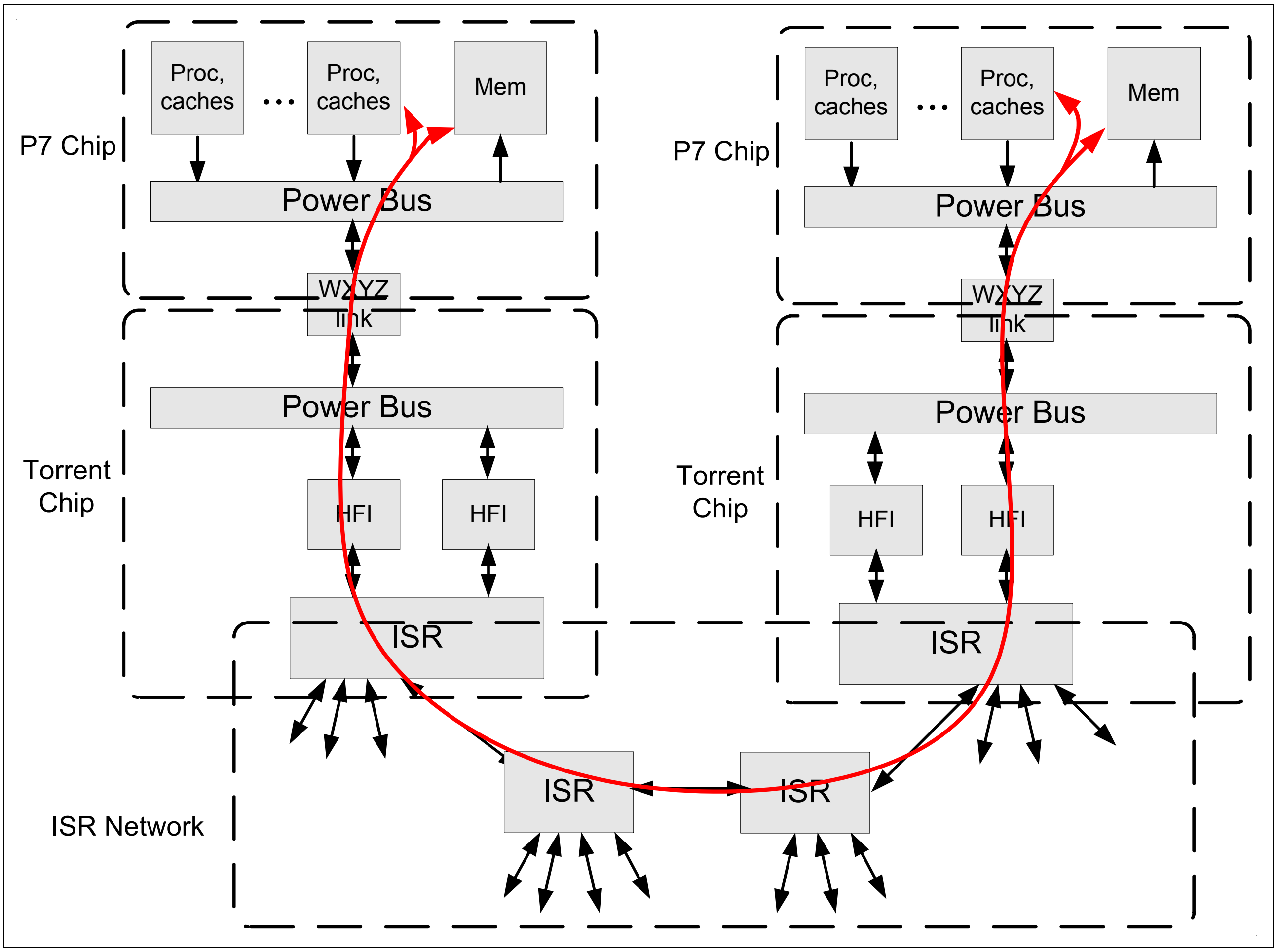


Figure 1-7 CAU packets received by CAU

As shown in Figure 1-8 on page 15, a CAU sends packets to the following locations:

- The memory of a remote node that is written to memory by the HFI of the remote node.
- The memory of a local node that is written to memory by the HFI of the local node.
- A remote CAU.

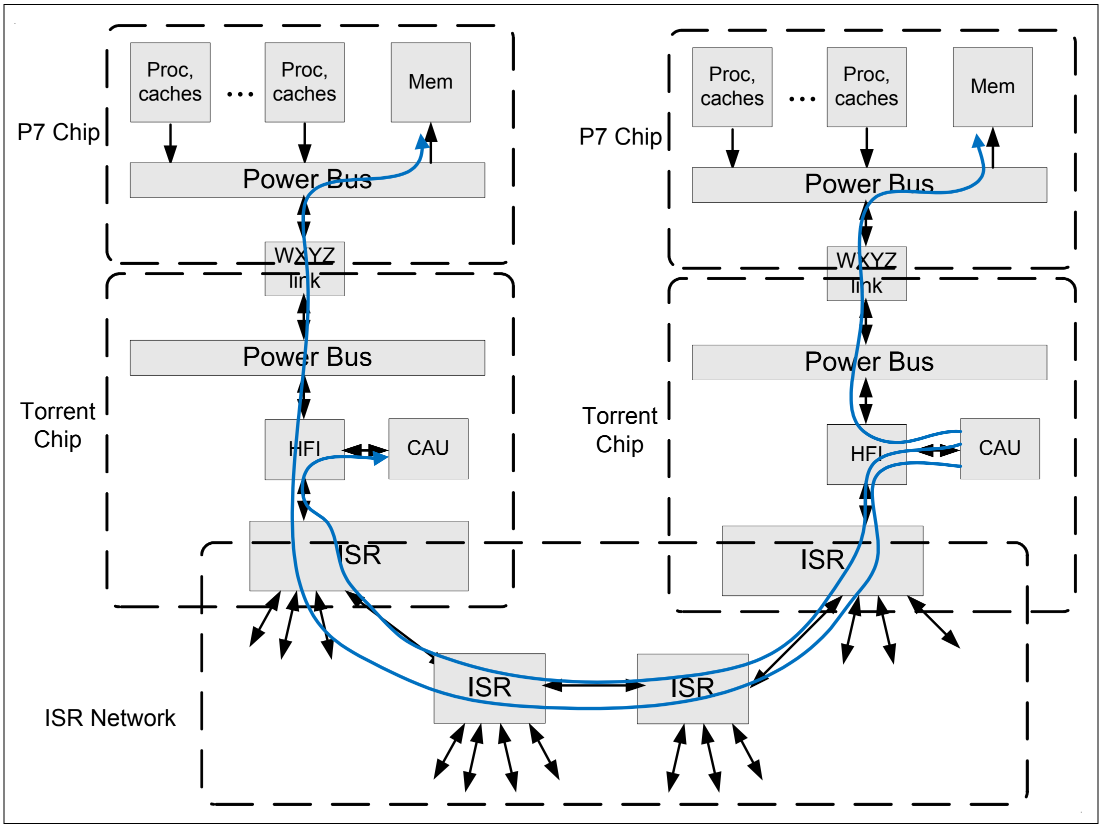


Figure 1-8 CAU packets sent by CAU

1.4.4 Nest memory management unit

The Nest Memory Management Unit (NMMU) that is in the hub check facilitates user-level code to operate on the address space of processes that executes on other compute nodes. The NMMU enables user-level code to create a global address space from which the NMMU performs operations. This facility is called *global shared memory*.

A process that executes on a compute node registers its address space, thus permitting interconnect packets to manipulate the registered shared region directly. The NMMU references a page table that maps effective addresses to real memory. The hub chip also maintains a cache of the mappings and maps the entire real memory of most installations.

Incoming interconnect packets that reference memory, such as RDMA packets and packets that perform atomic operations, contain an effective address and information that pinpoints the context in which to translate the effective address. This feature greatly facilitates global-address space languages, such as Unified Parallel C (UPC), co-array Fortran, and X10, by permitting such packets to contain easy-to-use effective addresses.

1.4.5 Integrated switch router

The integrated switch router (ISR) replaces the external switching and routing functions that are used in prior networks. The ISR is designed to dramatically reduce cost and improve performance in bandwidth and latency.

A direct graph network topology connects up to 65,536 POWER7 eight-core processor chips with two-level routing hierarchy of L and D busses.

Each hub chip ISR connects to four POWER7 chips via the HFI controller and the W busses. The Torrent hub chip and its four POWER7 chips are called an *octant*. Each ISR octant is directly connected to seven other octants on a drawer via the wide on-planar L-Local busses and to 24 other octants in three more drawers via the optical L-Remote busses.

A *Supernode* is the fully interconnected collection of 32 octants in four drawers. Up to 512 Supernodes are fully connected via the 16 optical D busses per hub chip. The ISR is designed to support smaller systems with multiple D busses between Supernodes for higher bandwidth and performance.

The ISR logically contains input and output buffering, a full crossbar switch, hierarchical route tables, link protocol framers/controllers, interface controllers (HFI and PB data), Network Management registers and controllers, and extensive RAS logic that includes link replay buffers.

The Integrated Switch Router supports the following features:

- Target cycle time up to 3 GHz
- Target switch latency of 15 ns
- Target GUPS: ~21 K. ISR assisted GUPs handling at all intermediate hops (not software)
- Target switch crossbar bandwidth greater than 1 TB per second input and output:
  - 96 Gbps WXYZ-busses (4 @ 24 Gbps) from P7 chips (unidirectional)
  - 168 Gbps local L-busses (7 @ 24 Gbps) between octants in a drawer (unidirectional)
  - 144 Gbps optical L-busses (24 @ 6 Gbps) to other drawers (unidirectional)
  - 160 Gbps D-busses (16 @ 10 Gbps) to other Supernodes (unidirectional)
- Two-tiered full-graph network
- Virtual Channels for deadlock prevention
- Cut-through Wormhole routing
- Routing Options:
  - Full hardware routing
  - Software-controlled indirect routing by using hardware route tables
- Multiple indirect routes that are supported for data striping and failover
- Multiple direct routes by using LR and D-links supported for less than a full-up system
- Maximum packet size that supported is 2 KB. Packets size varies from 1 to 16 flits, each flit being 128 Bytes
- Routing Algorithms:
  - Round Robin: Direct and Indirect
  - Random: Indirect routes only
- IP Multicast with central buffer and route table and supports 256 Bytes or 2 KB packets
- Global Hardware Counter implementation and support and includes link latency counts
- LCRC on L and D busses with link-level retry support for handling transient errors and includes error thresholds.
- ECC on local L and W busses, internal arrays, and busses and includes Fault Isolation Registers and Control Checker support
- Performance Counters and Trace Debug support

1.4.6 SuperNOVA

SuperNOVA is the second member of the fourth generation of the IBM Synchronous Memory Interface ASIC. It connects host memory controllers to DDR3 memory devices.

SuperNOVA is used in a planar configuration to connect to Industry Standard (I/S) DDR3 RDIMMs. SuperNOVA also resides on a custom, fully buffered memory module that is called the SuperNOVA DIMM (SND). Fully buffered DIMMs use a logic device, such as SuperNOVA, to buffer all signals to and from the memory devices.

As shown in Figure 1-9, SuperNOVA provides the following features:

- Cascaded memory channel (up to seven SNs deep) that use 6.4-Gbps, differential ended (DE), unidirectional links.
- Two DDR3 SDRAM command and address ports.
- Two, 8 B DDR3 SDRAM data ports with a ninth byte for ECC and a tenth byte that is used as a locally selectable spare.
- 16 ranks of chip selects and CKE controls (eight per CMD port).
- Eight ODT (four per CMD port).
- Four differential memory clock pairs to support up to four DDR3 registered dual in-line memory modules (RDIMMs).

Data Flow Modes include the following features:

- Expansion memory channel daisy-chain
- 4:1 or 6:1 configurable data rate ratio between memory channel and SDRAM domain

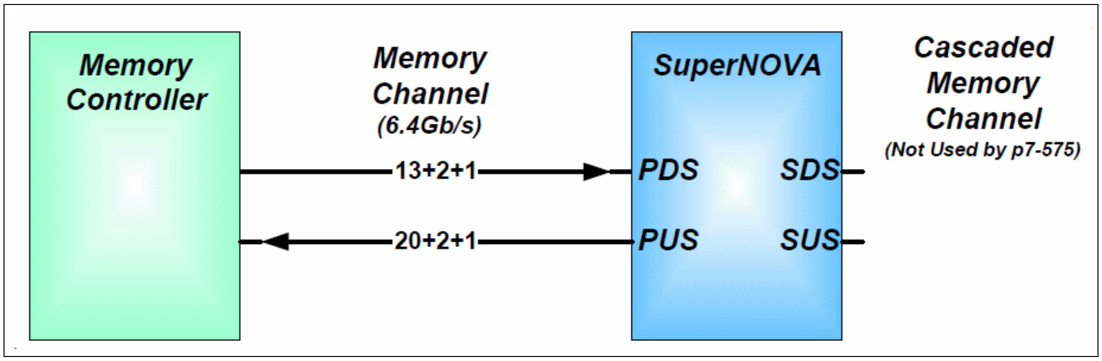


Figure 1-9 Memory channel

SuperNOVA uses a high speed, differential ended communications memory channel to link a host memory controller to the main memory storage devices through the SuperNOVA ASIC. The maximum memory channel transfer rate is 6.4 Gbps.

The SuperNOVA memory channel consists of two DE, unidirectional links. The downstream link transmits write data and commands away from the host (memory controller) to the SuperNOVA. The downstream includes 13 active logical signals (lanes), two more spare lanes, and a bus clock. The upstream (US), link transmits read data and responses from the SuperNOVA back to the host. The US includes 20 active logical signals, two more spare lanes, and a bus clock.

Although SuperNOVA supports a cascaded memory channel topology of multiple chips that use daisy chained memory channel links, Power 775 does not use this capability.

The links that are connected on the host side are called the *Primary Up Stream* (PUS) and *Primary Down Stream* (PDS) links. The links on the cascaded side are called the *Secondary Up Stream* (SUS) and *Secondary Down Stream* (SDS) links.

The SuperNOVA US and downstream links each include two dedicated spare lanes. One of these lanes is used to repair either a clock or data connection. The other lane is used only to repair data signal defects. Each segment (host to SuperNOVA or SuperNOVA to SuperNOVA connection) of a cascaded memory channel is independently deployed of their dedicated spares per link. This deployment maximizes the ability to survive multiple interconnect hard failures. The spare lanes are tested and aligned during initialization but are deactivated during normal runtime operation. The channel frame format, error detection, and protocols are the same before and after spare lane invocation. Spare lanes are selected by one of the following means:

- The spare lanes are selected during initialization by loading host and SuperNOVA configuration registers based on previously logged lane failure information.
- The spare lanes are selected dynamically by the hardware during runtime operation by an error recovery operation that performs the link reinitialization and repair procedure. This procedure is initiated by the host memory controller and supported by the SuperNOVAs in the memory channel. During the link repair operation, the memory controller holds back memory access requests. The procedure is designed to take less than 10 ms to prevent system performance problems, such as timeouts.
- The spare lanes are selected by system control software by loading host or SuperNOVA configuration registers that are based on the results of the memory channel lane shadowing diagnostic procedure.

1.4.7 Hub module

The Power 775 hub module provides all the connectivity that is needed to form a clustered system, as shown in Figure 1-10 on page 19.

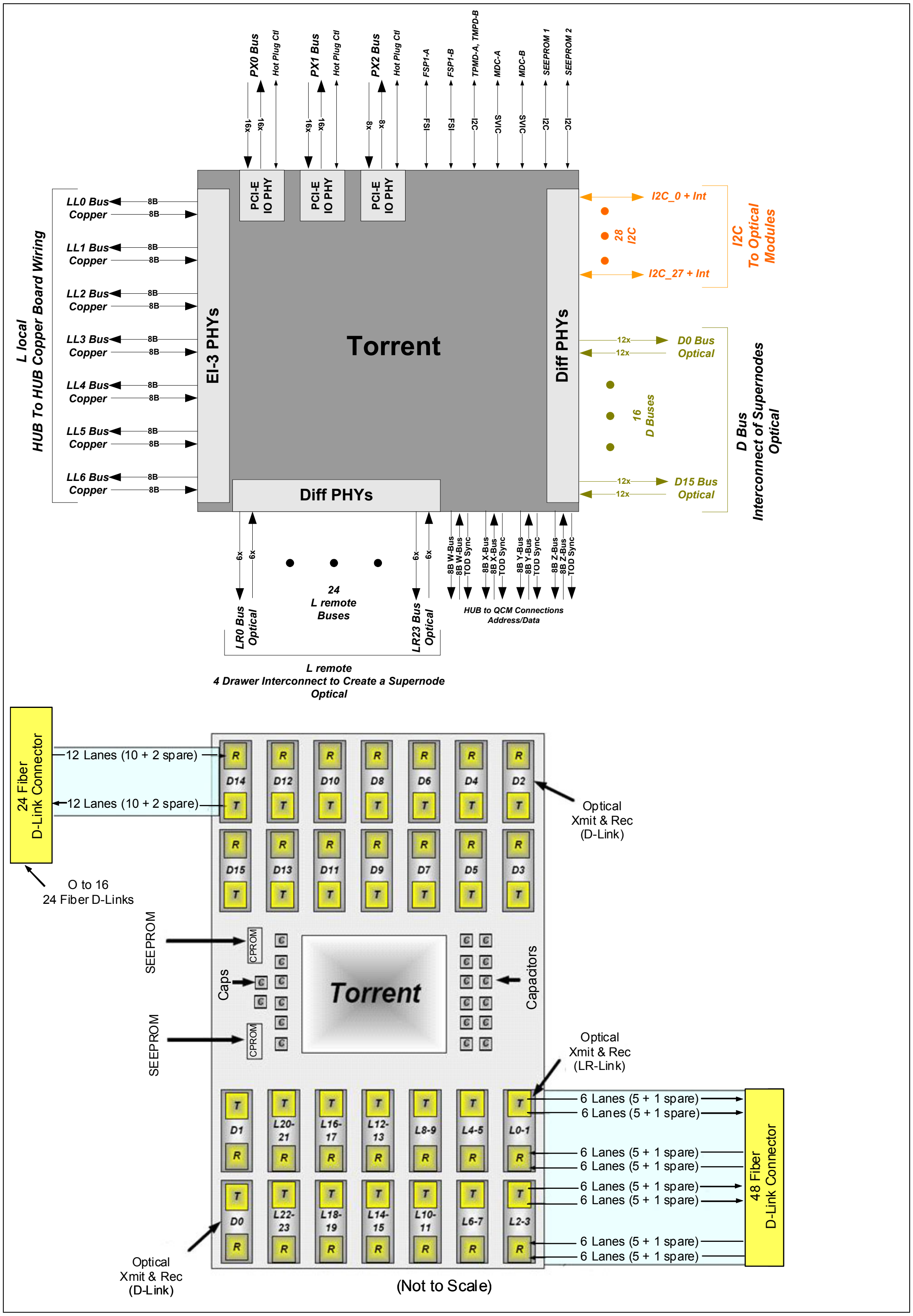


Figure 1-10 Hub module diagram

The hub features the following primary functions:

- Connects the QCM Processor/Memory subsystem to up to two high-performance 16x PCIe slots and one high-performance 8x PCI Express slot. This configuration provides general-purpose I/O and networking capability for the server node.

  POWER 775 drawer: In a Power 775 drawer (CEC), the Octant 0 has 3 PCIe, in which two PCIe are 16x and one PCIe is 8x (SRIOV Ethernet Adapter is given priority in the 8x slot.). Octants 1-7 have two PCI Express, which are 16x.

- Connects eight Processor QCMs together by using a low-latency, high-bandwidth, coherent copper fabric (L-Local buses) that includes the following features:
  - Enables a single hypervisor to run across 8 QCMs, which enables a single pair of redundant service processors to manage 8 QCMs
  - Directs the I/O slots that are attached to the eight hubs to the compute power of any of the eight QCMs that provide I/O capability where needed
  - Provides a message passing mechanism with high bandwidth and the lowest possible latency between eight QCMs (8.2 TFLOPs) of compute power
- Connects four Power 775 planars via the L-Remote optical connections to create a 33 TFLOP tightly connected compute building block (SuperNode). The bi-sectional exchange bandwidth between the four boards is 3 TBps, the same bandwidth as 1500 10 Gb Ethernet links.
- Connects up to 512 groups of four planers (SuperNodes) together via the D optical buses with ~3 TBs of exiting bandwidth per planer.

Optical links

The Hub modules that are on the node board house optical transceivers for up to 24 L-Remote links and 16 D-Links. Each optical transceiver includes a jumper cable that connects the transceiver to the node tailstock. The transceivers are included to facilitate cost optimization, depending on the application. The supported options are shown in Table 1-2.

Table 1-2 Supported optical link options

SuperNode type	L-Remote links	D-Links	Number of combinations
SuperNodes not enabled	0	0-16 in increments of 1	17
Full SuperNodes	24	0-16 in increments of 1	17
			34

Some customization options are available on the hub optics module, which allow some optic transceivers to remain unpopulated on the Torrent module if the wanted topology does not require all of transceivers. The number of actual offering options that are deployed is dependent on specific large customer bids.

Optics physical package

The optics physical package includes the following features:

- Individual transmit (Tx) and Receive (Rx) modules that are packaged in Tx+Rx pairs on glass ceramic substrate.
- Up to 28 Tx+Rx pairs per module.
- uLGA (Micro-Land Grid Array) at 0.7424 mm pitch interconnects optical modules to ceramic substrate.
- 12-fiber optical fiber ribbon on top of each Tx and each Rx module, which is coupled through Prizm reflecting and spheric-focusing 12-channel connectors.
- Copper saddle over each optical module and optical connector for uLGA actuation and heat spreading.
- Heat spreader with springs and thermal interface materials that provide uLGA actuation and heat removal separately for each optical module.
- South (rear) side of each glass ceramic module carries 12 Tx+Rx optical pairs that support 24 (6+6) fiber LR-links, and 2 Tx+Rx pairs that support 2 (12+12) fiber D-links.
- North (front) side of each glass ceramic module carries 14 Tx+Rx optical module pairs that support 14 (12+12) fiber D-links

Optics electrical interface

The optics electrical interface includes the following features:

- 12 differential pairs @ 10 GB per second (24 signals) for each TX optics module
- 12 differential pairs @ 10 GB per second (24 signals) for each RX module
- Three wire I2C/TWS (Serial Data & Address, Serial Clock, Interrupt): three signals

Cooling

Cooling includes the following features:

- Optics are water-cooled with Hub chip
- Cold plate on top of module, which is coupled to optics through heat reader and saddles, with thermal interface materials at each junction
- Recommended temperature range: 20C – 55C at top of optics modules

Optics drive/receive distances

Optics links might be up to 60 meters rack-to-rack (61.5 meters, including inside-drawer optical fiber ribbons).

Reliability assumed

The following reliability features are assumed:

- 10 FIT rate per lane.
- D-link redundancy. Each (12+12)-fiber D-link runs normally with 10 active lanes and two spares. Each D-link runs in degraded-Bandwidth mode with as few as eight lanes.
- LR-link redundancy: Each (6+6)-fiber D-link runs normally with six active lanes. Each LR-link (half of a Tx+Rx pair) runs in degraded-bandwidth mode with as few as four lanes out of six lanes.
- Overall redundancy: As many four lanes out of each 12 (two lanes of each six lanes) might fail without disabling any D-links or LR-links.
- Expect to allow one failed lane per 12 lanes in manufacturing.
- Bit Error Rate: Worst-case, end-of-life BER is 10^-12. Normal expected BER is 10^-18

1.4.8 Memory subsystem

The memory controller layout is shown in Figure 1-11.

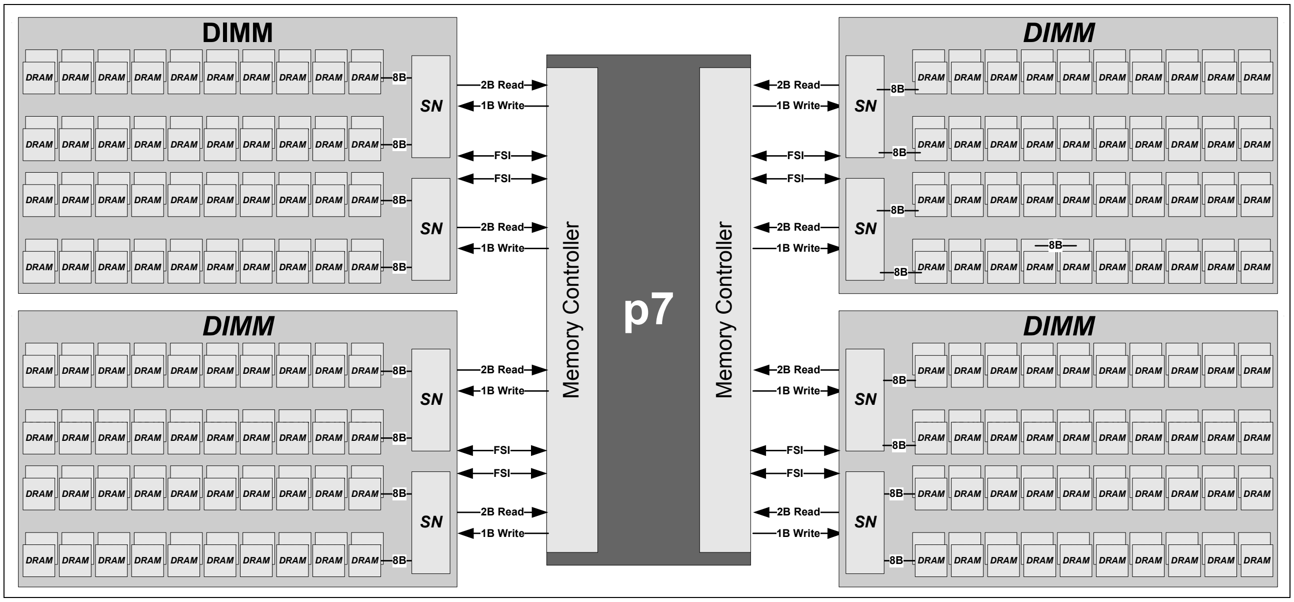


Figure 1-11 Memory controller layout

The memory cache sizes are shown in Table 1-3.

Table 1-3 Cache memory sizes

Cache level	Memory size (per core)
L1	32 KB Instruction, 32 KB Data
L2	256 KB
L3	4 MB eDRAM

The memory subsystem features the following characteristics:

- Memory capacity: Up to 128 GB/processor
- Memory bandwidth: 128 GB/s (peak)/processor
- Eight channels of SuperNOVA buffered DIMMs/processor
- Two memory controllers per processor:
  - Four memory busses per memory controller
  - Each buss is 1 B-wide Write, 2 B-wide Read

Memory per drawer

Each drawer features the following minimum and maximum memory ranges:

- Minimum:
  - 4 DIMMs per QCM x 8 QCM per drawer = 32 DIMMs per drawer
  - 32 DIMMs per drawer x 8 GB per DIMM = 256 GB per drawer
- Maximum:
  - 16 DIMMs per QCM x 8 QCM per drawer = 128 DIMMs per drawer
  - 128 DIMMs per drawer x 16 GB per DIMM = 2 TB per drawer

Memory DIMMs

Memory DIMMs include the following features:

- Two SuperNOVA chips each with a bus connected directly to the processor
- Two ports on the DIMM from each SuperNova
- Dual CFAM interfaces from the processor to each DIMM, wired to the primary SuperNOVA and dual chained to the secondary SuperNOVA on the DIMM
- Two VPD SEEPROMs on the DIMM interfaced to the primary SuperNOVA CFAM
- 80 DRAM sites - 2 x 10 (x8) DRAM ranks per SuperNova Port
- Water cooled jacketed design
- 50 watt max DIMM power
- Available in sizes: 8 GB, 16 GB, and 32 GB (RPQ)

For best performance, it is recommended that all 16 DIMM slots are plugged in each node. All DIMMs driven by a quad-chip module (QCM) must have the same size, speed, and voltage rating.

1.4.9 Quad chip module

The previous sections provided a brief introduction to the low-level components of the Power 775 system. We now look at the system on a modular level. This section discusses the quad-chip module or QCM, which contains four POWER7 chips that are connected in a ceramic module.

The standard Power 775 CEC drawer contains eight QCMs. Each QCM contains four, 8-core POWER7 processor chips and supports 16 DDR3 SuperNova buffered memory DIMMs.

Figure 1-12 on page 24 shows the POWER7 quad chip module which contains the following characteristics:

- 4x POWER7 cores
- 32 cores (4 x 8 = 32)
- 948 GFLOPs / QCM
- 474 GOPS (Integer) / QCM
- Off-chip bandwidth: 336 Gbps (peak):
  - local + remote interconnect

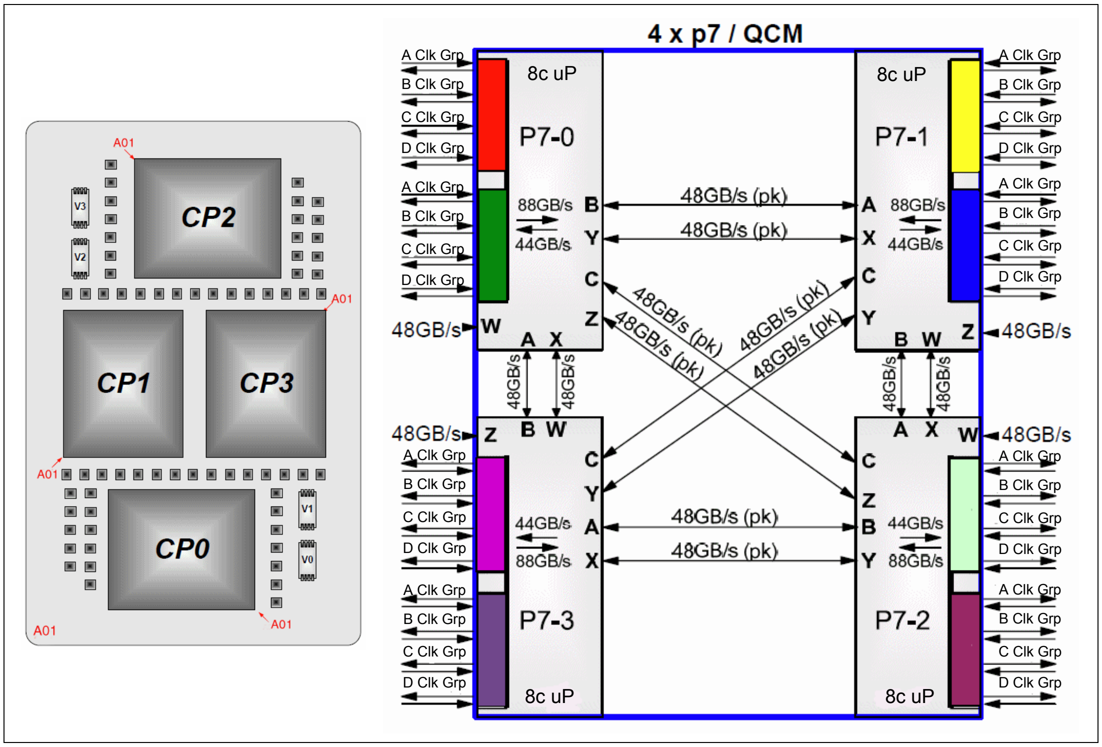


Figure 1-12 POWER7 quad chip module

1.4.10 Octant

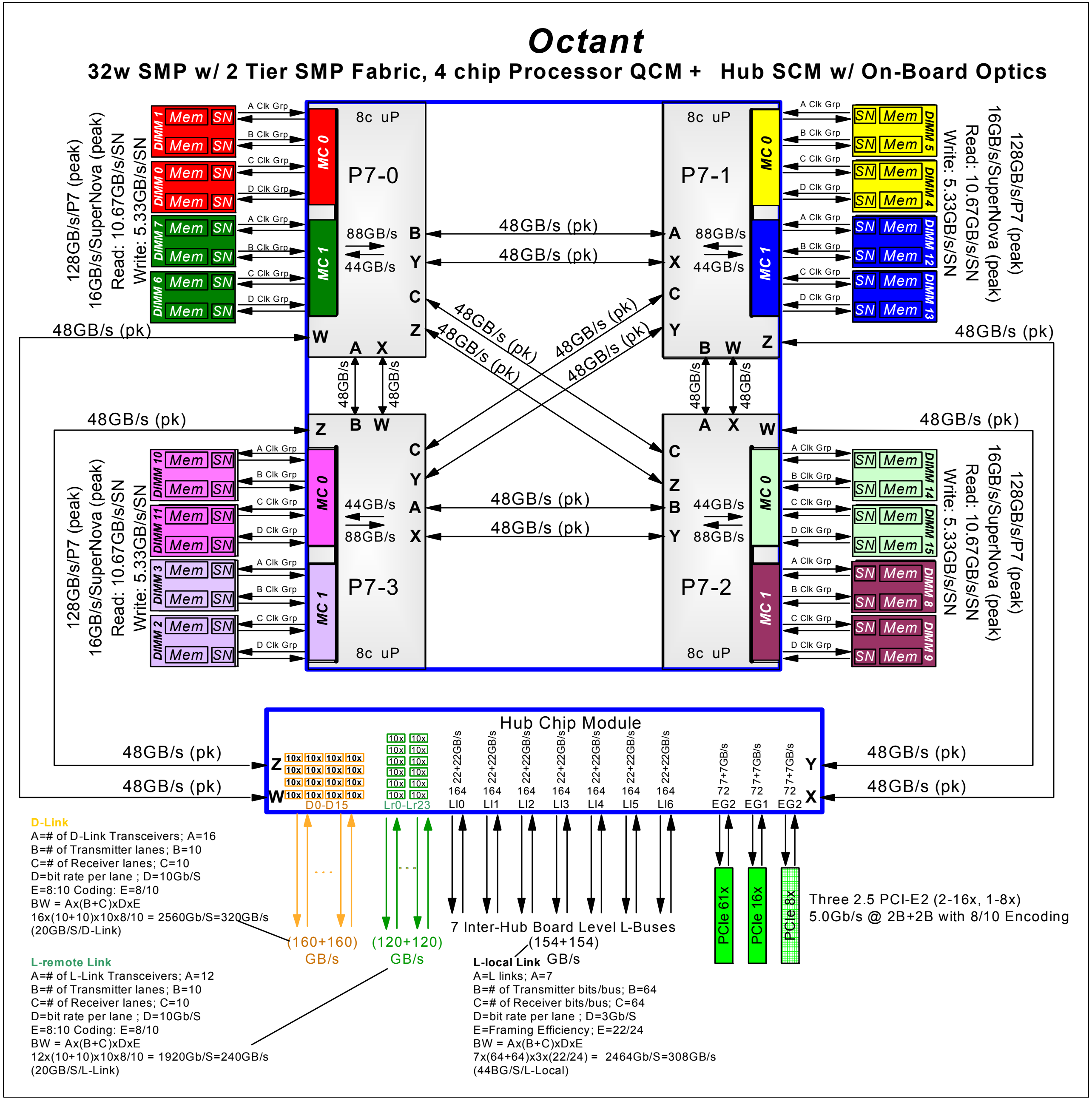
This section discusses the “octant” level of the system.

Figure 1-13 on page 25 shows a 32-way SMP with two-tier SMP fabric, four chip processor MCM + Hub SCM with onboard optics.

Each octant represents 1/8 of the CEC planar, which contains one QCM, one Hub module, and up to 16 associated memory modules.

Octant 0: Octant 0 controls another PCIe 8x slot that is used for an Ethernet adapter for cluster management.

Figure 1-13 Power 775 octant logic diagram


Octant
32w SMP w/ 2 Tier SMP Fabric, 4 chip Processor QCM + Hub SCM w/ On-Board Optics
128GB/s/P7 (peak)
16GB/s/SuperNova (peak)
Read: 10.67GB/s/SN
Write: 5.33GB/s/SN
DIMM 1
DIMM 0
DIMM 7
DIMM 6
DIMM 5
DIMM 4
DIMM 12
DIMM 13
DIMM 10
DIMM 11
DIMM 3
DIMM 2
DIMM 14
DIMM 15
DIMM 8
DIMM 9
Mem
SN
A Clk Grp
B Clk Grp
C Clk Grp
D Clk Grp
MC 0
MC 1
8c uP
P7-0
P7-1
P7-2
P7-3
88GB/s
44GB/s
48GB/s (pk)
48GB/s
A
B
C
W
X
Y
Z
Hub Chip Module
10x
D0-D15
Lr0-Lr23
22+22GB/s
164
LI0
LI1
LI2
LI3
LI4
LI5
LI6
7+7GB/s
72
EG2
EG1
EG2
PCIe 61x
PCIe 16x
PCIe 8x
Three 2.5 PCI-E2 (2-16x, 1-8x)
5.0Gb/s @ 2B+2B with 8/10 Encoding
(160+160)
GB/s
(120+120)
GB/s
7 Inter-Hub Board Level L-Buses
(154+154)
GB/s
D-Link
A=# of D-Link Transceivers; A=16
B=# of Transmitter lanes; B=10
C=# of Receiver lanes; C=10
D=bit rate per lane ; D=10Gb/S
E=8:10 Coding: E=8/10
BW = Ax(B+C)xDxE
16x(10+10)x10x8/10 = 2560Gb/S=320GB/s
(20GB/S/D-Link)
L-remote Link
A=# of L-Link Transceivers; A=12
B=# of Transmitter lanes; B=10
C=# of Receiver lanes; C=10
D=bit rate per lane ; D=10Gb/S
E=8:10 Coding: E=8/10
BW = Ax(B+C)xDxE
12x(10+10)x10x8/10 = 1920Gb/S=240GB/s
(20GB/S/L-Link)
L-local Link
A=L links; A=7
B=# of Transmitter bits/bus; B=64
C=# of Receiver bits/bus; C=64
D=bit rate per lane ; D=3Gb/S
E=Framing Efficiency; E=22/24
BW = Ax(B+C)xDxE
7x(64+64)x3x(22/24) = 2464Gb/S=308GB/s
(44BG/S/L-Local)

Each Power 775 planar consists of eight octants, as shown in Figure 1-14 on page 26. Seven of the octants are composed of 1x QCM, 1 x HUB, 2 x PCI Express 16x. The other octant contains 1x QCM, 1x HUB, 2x PCI Express 16x, 1x PCI Express 8x.

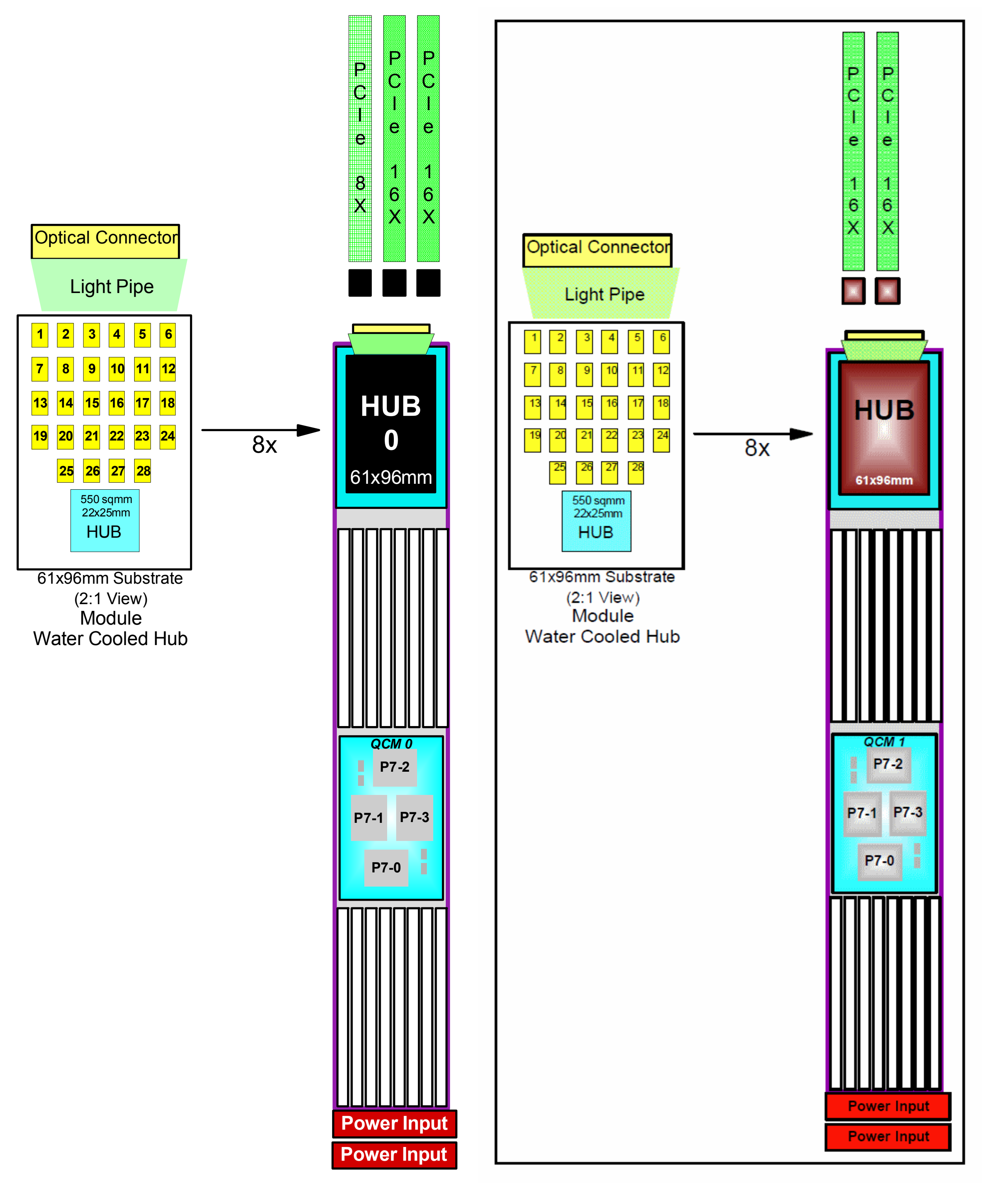


Figure 1-14 Octant layout differences

The Power 7755 includes the following features:

- Compute power: 1 TF, 32 cores, 128 threads (Flat Memory Design)
- Memory: 512 GB max capacity, ~512 Gbps peak memory BW (1/2 B/FLOP):
  - 1280 DRAM sites (80 Dram sites per DIMM, 1/2/4 Gb DRAMS over time)
  - 2 Processor buses and four 8B memory buses per DIMM (two Buffers)
  - Double stacked at the extreme with memory access throttling

    For more information, see 1.4.8, “Memory subsystem” on page 22.
- I/O: 1 TB/s BW:
  - 32 Gbps Generic I/O:
    - Two PCIe2 16x line rate capable busses
    - Expanded I/O slot count through PCIe2 expansion possible
  - 980 Gbps maximum Switch Fabric BW (12 optical lanes active)
- IBM Proprietary Fabric with On-board copper/Off-board Optical:
  - Excellent cost / performance (especially at mid-large scale)
  - Basic technology can be adjusted for low or high BW applications
- Packaging:
  - Water Cooled (>95% at level shown), distributed N+1 Point Of Load Power
  - High wire count board with small Vertical Interconnect Accesses (VIAs)
  - High pin count LGA module sockets
  - Hot Plug PCIe Air Cooled I/O Adapter Design
  - Fully Redundant Out-of-band Management Control

1.4.11 Interconnect levels

A functional Power 775 system consists of multiples nodes that are spread across several racks. This configuration means multiple octants are available in which every octant is connected to every other octant on the system. The following levels of interconnect are available on a system:

- First level

  This level connects the eight octants in a node together via the hub module by using copper board wiring. This interconnect level is referred to as “L” local (LL). Every octant in the node is connected to every other octant. For more information, see 1.4.12, “Node” on page 28.
- Second level

  This level connects four nodes together to create a Supernode. This interconnection is possible via the hub module optical links. This interconnection level is referred to as “L” distant (LD). Every octant in a node must connect to every other octant in the other three nodes that form the Supernode. Every octant features 24 connections, but the total number of connections across the four nodes in a Supernode is 384. For more information, see 1.4.13, “Supernodes” on page 30.
- Third level

  This level connects every Supernode to every other Supernode in a system. This interconnection is possible via the hub module optical links. This interconnect level is referred to as D-link. Each Supernode has up to 512 D-links. It is possible to scale up this level to 512 Supernodes. Every Supernode has a minimum of one hop D-link to every other Supernode. For more information, see 1.4.14, “Power 775 system” on page 32.

1.4.12 Node

This section discusses the node level of the Power 775 physically represented by the drawer, also commonly referred as CEC. A node is composed of eight octants and their local interconnect. Figure 1-15 shows the CEC drawer from the front.

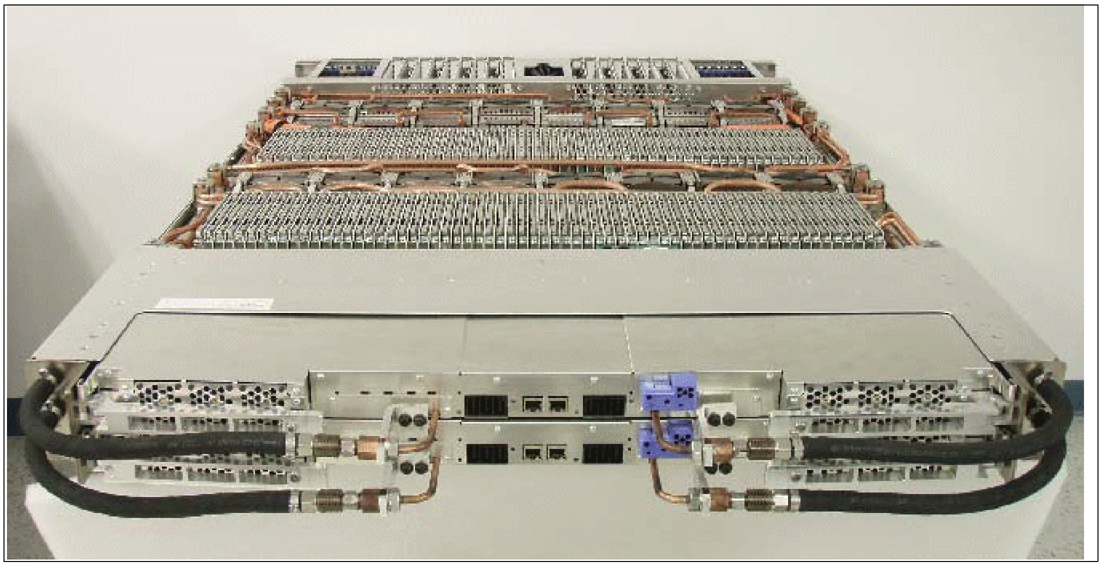


Figure 1-15 CEC drawer front view

Figure 1-16 shows the CEC drawer rear view.

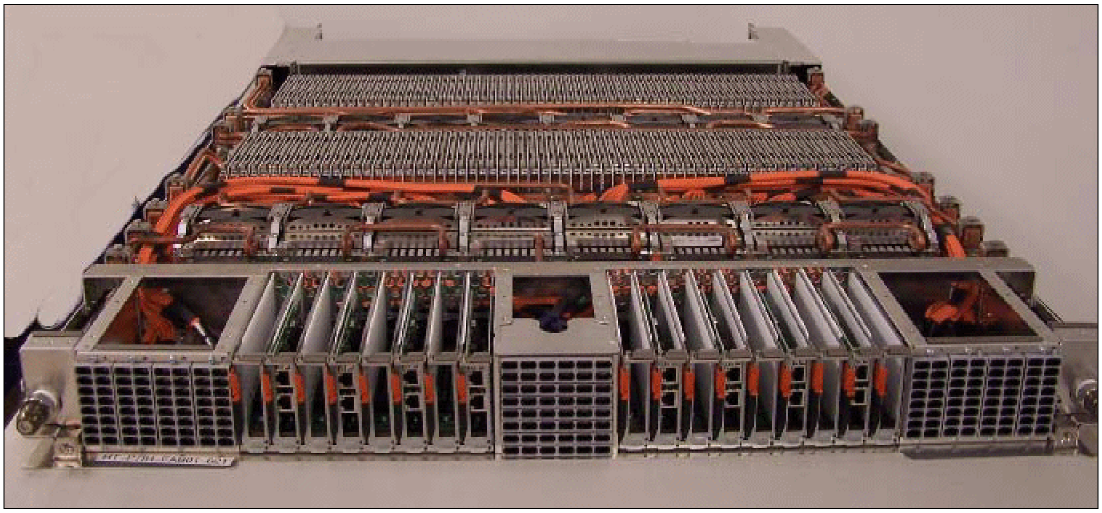


Figure 1-16 CEC drawer rear view

First level interconnect: L Local

L Local (LL) connects the eight octants in the CEC drawer together via the HUB module by using copper board wiring. Every octant in the node is connected to every other octant, as shown in Figure 1-17.

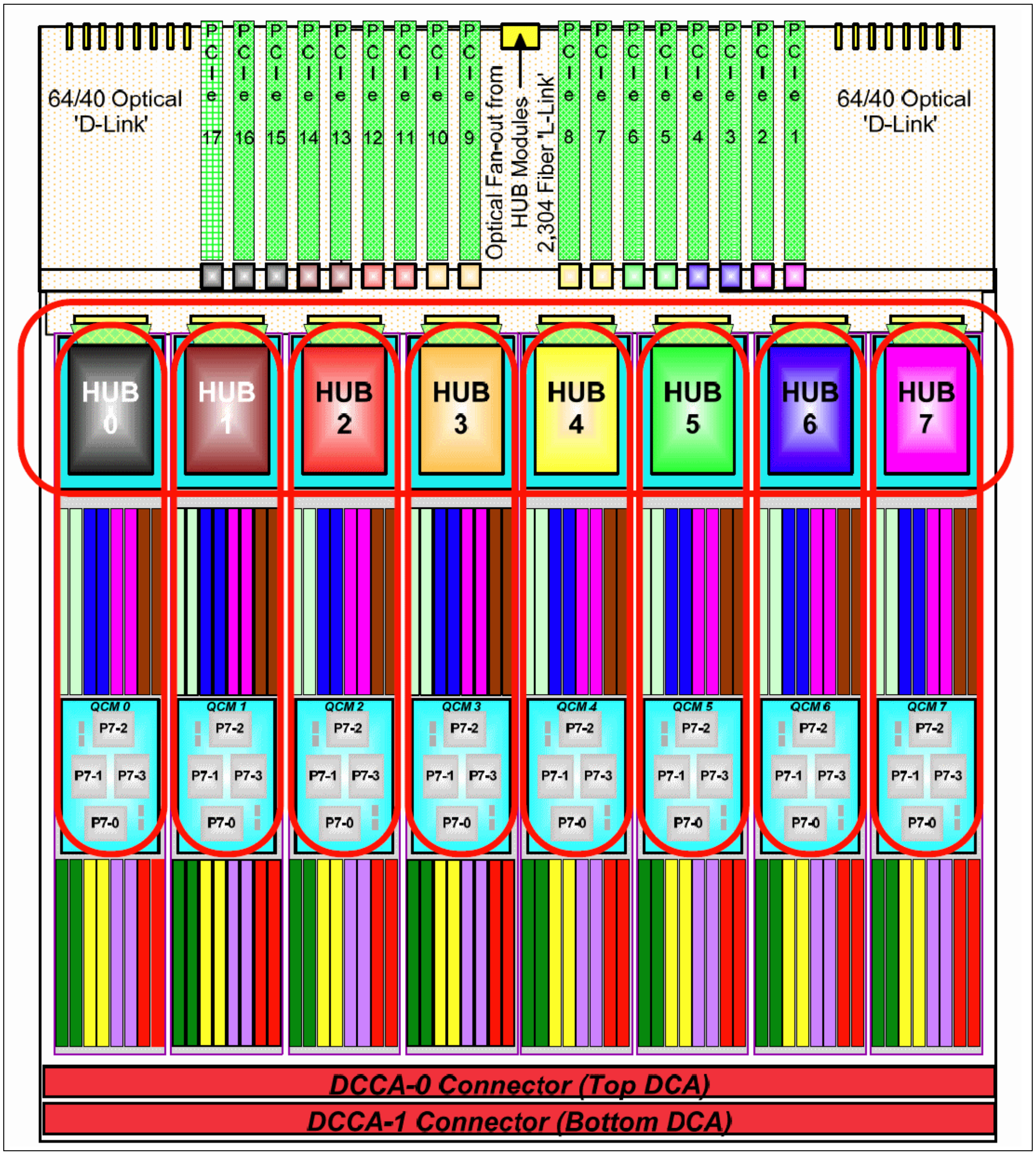


Figure 1-17 First level local interconnect (256 cores)

System planar board

This section provides details about the following system planar board characteristics:

- Approximately 2U x 85 cm wide x 131cm deep overall node package in 30 EIA frame
- Eight octants and each octant features one QCM, one hub, and 4 - 16 DIMMs
- 128 memory slots
- 17 PCI adapter slots (octant 0 has three PCI slots, octant 1 - 7 each have two PCI slots)
- Regulators on the bottom side of planar directly under modules to reduce loss and decoupling capacitance
- Water-cooled stiffener to cool regulators on bottom side of planar and memory DIMMs on top of board
- Connectors on rear of board to optional PCI cards (17x)
- Connectors on front of board to redundant 2N DCCAs
- Optical fiber D Link interface cables from HUB modules to left and right of rear tail stock
- 128 total = 16 links x 8 Hub Modules
- Optical fiber L-remote interface cables from Hub modules to center of rear tail stock
- 96 total = 24 links x 8 Hub Modules
- Clock distribution & out-of-band control distribution from DCCA.

Redundant service processor

The redundant service processor features the following characteristics:

- The clocking source follows the topology of the service processor.
- N+1 redundancy of the service processor and clock source logic use two inputs on each processor and HUB chip.
- Out-of-band signal distribution for memory subsystem (SuperNOVA chips) and PCI express slots are consolidated to the standby powered pervasive unit on the processor and HUB chips. PCI express is managed on the Hub and Bridge chips.

1.4.13 Supernodes

This section describes the concept of Supernodes.

Supernode configurations

The following supported Supernode configurations are used in a Power 775 system. The usage of each type is based on cluster size or application requirements:

- Four-drawer Supernode

  The four-drawer Supernode is the most common Supernode configuration. This configuration is formed by four CEC drawers (32 octants) connected via Hub optical links.

- Single-drawer Supernode

  In this configuration, each CEC drawer in the system is a Supernode.

Second level interconnect: L Remote

This level connects four CEC drawers (32 octants) together to create a Supernode via Hub module optical links. Every octant in a node connects to every other octant in the other three nodes in the Supernode. There are 384 connections in this level, as shown in Figure 1-18 on page 31.

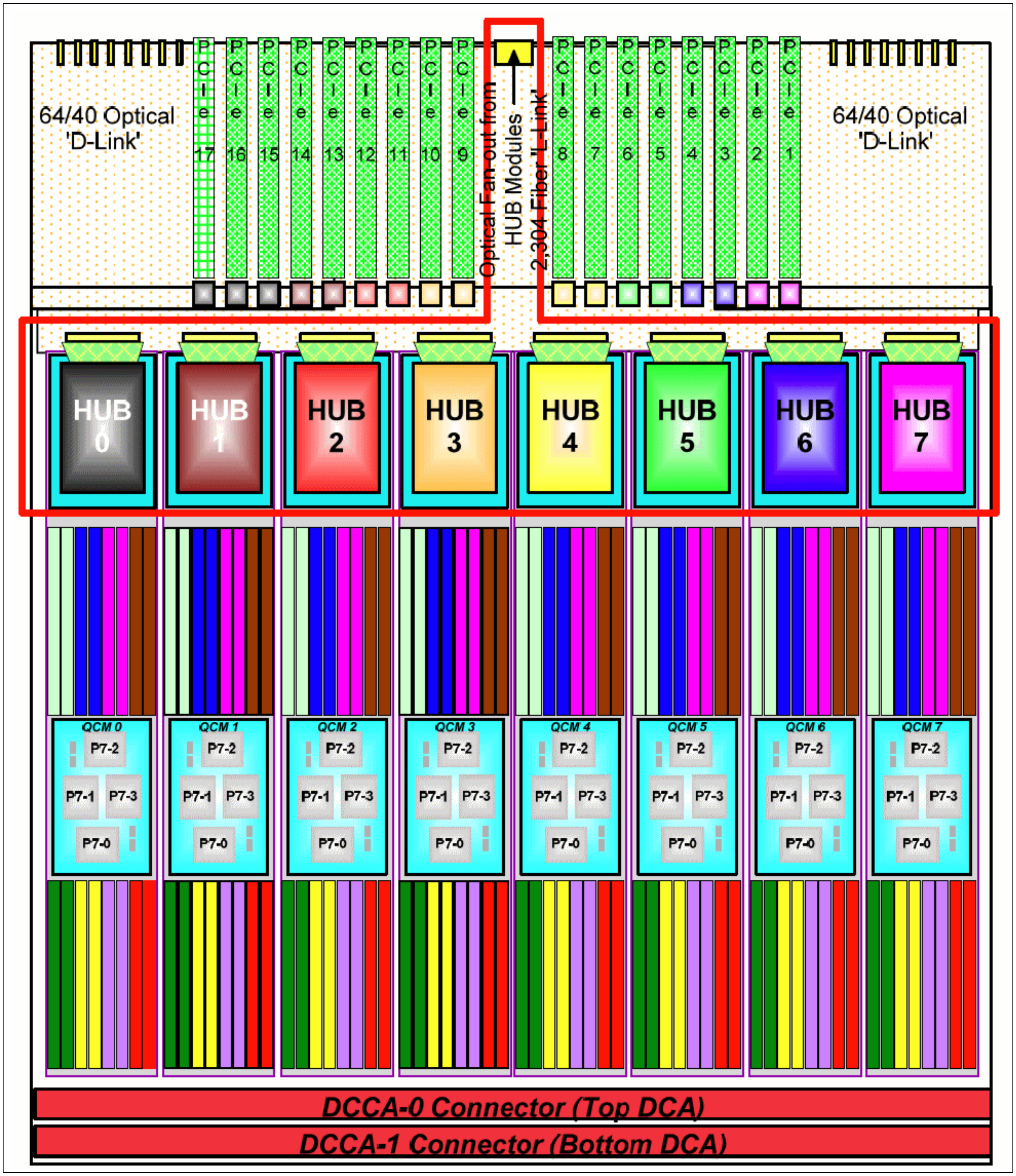
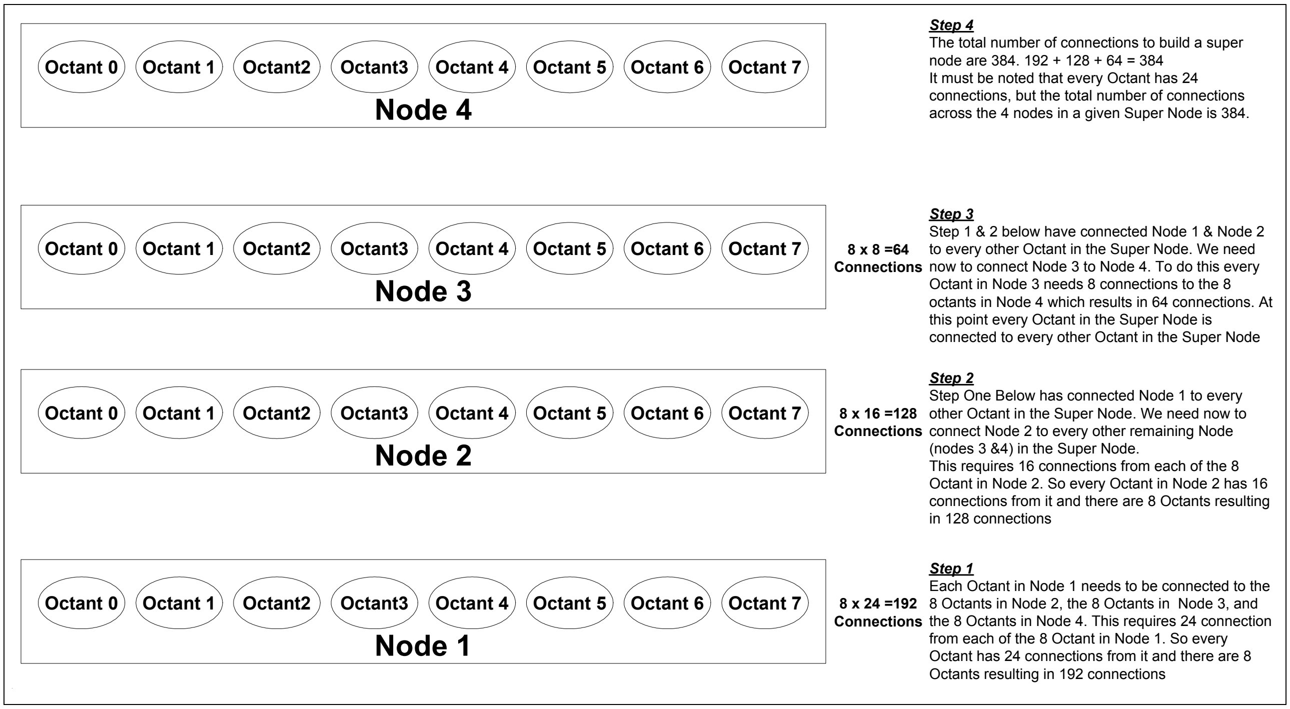


Figure 1-18 Board 2nd level interconnect (1,024 cores)

The second level wiring connector count is shown in Figure 1-19 on page 32.



Step 4
The total number of connections to build a super node are 384. 192 + 128 + 64 = 384
It must be noted that every Octant has 24 connections, but the total number of connections across the 4 nodes in a given Super Node is 384.

Step 3
Step 1 & 2 below have connected Node 1 & Node 2 to every other Octant in the Super Node. We need now to connect Node 3 to Node 4. To do this every Octant in Node 3 needs 8 connections to the 8 octants in Node 4 which results in 64 connections. At this point every Octant in the Super Node is connected to every other Octant in the Super Node

Step 2
Step One Below has connected Node 1 to every other Octant in the Super Node. We need now to connect Node 2 to every other remaining Node (nodes 3 &4) in the Super Node.
This requires 16 connections from each of the 8 Octant in Node 2. So every Octant in Node 2 has 16 connections from it and there are 8 Octants resulting in 128 connections

Step 1
Each Octant in Node 1 needs to be connected to the 8 Octants in Node 2, the 8 Octants in Node 3, and the 8 Octants in Node 4. This requires 24 connection from each of the 8 Octant in Node 1. So every Octant has 24 connections from it and there are 8 Octants resulting in 192 connections

Figure 1-19 Second level wiring connector count

1.4.14 Power 775 system

This section describes the Power 775 system and provides details about the third level of interconnect.

Third level interconnect: Distance

This level connects every Supernode to every other Supernode in a system by using Hub module optical links. Each Supernode includes up to 512 D-links, which allows for system that contains up to 512 Supernodes. Every Supernode features a minimum of one hop D-link to every other Supernode, and there are multiple two hop connections, as shown in Figure 1-20 on page 33.

Each HUB contains 16 Optical D-Links. The Physical node (board) contains eight HUBs; therefore, a physical node (board) contains 16 x 8 = 128 Optical D-Links. A Super Node is four Physical Nodes, which result in 16 x 8 x 4 = 512 Optical D-Links per Super node. This configuration allows up to 2048 CEC connected drawers.

In smaller configurations, in which the system features less than 512 Super Nodes, more than one optical D-Link per node is possible. Multiple connections between Supernodes are used for redundancy and higher bandwidth solutions.

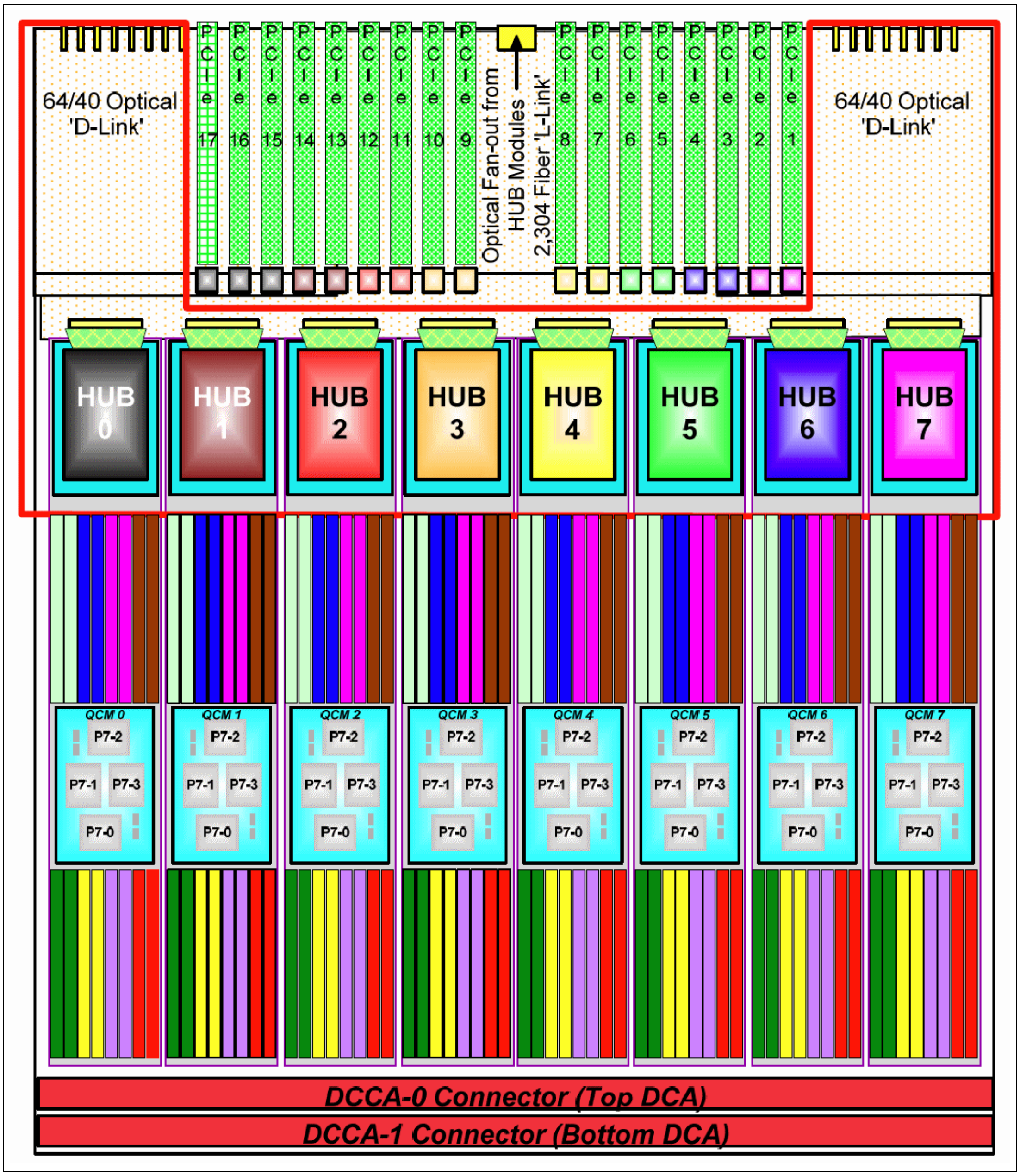


Figure 1-20 System third level interconnect

Integrated cluster fabric interconnect

A complete Power 775 system configuration is achieved by configuring server nodes into a tight cluster by using a fully integrated switch fabric.

The fabric is a multitier, hierarchical implementation that connects eight logical nodes (octants) together in the physical node (server drawer or CEC) by using copper L-local links. Four physical nodes are connected with structured optical cabling into a Supernode by using optical L-remote links. Up to 512 super nodes are connected by using optical D-links. Figure 1-21 on page 34 shows a logical representation of a Power 775 cluster.

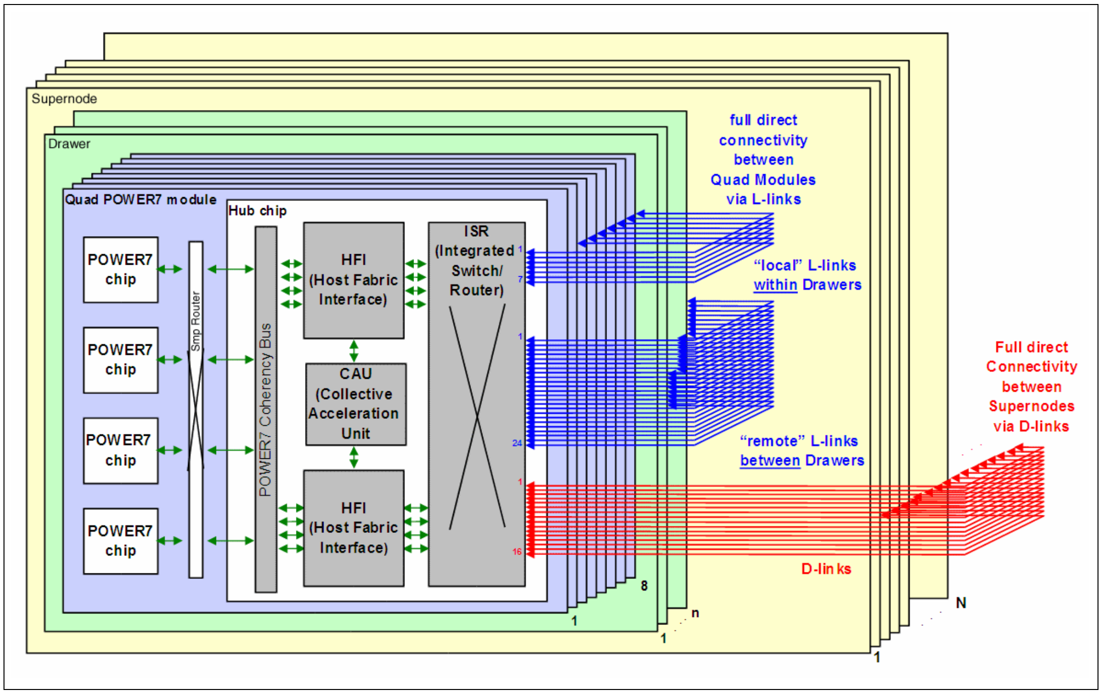

Supernode
Drawer
Quad POWER7 module
POWER7 chip
POWER7 chip
POWER7 chip
POWER7 chip
Smp Router
Hub chip
POWER7 Coherency Bus
HFI (Host Fabric Interface)
CAU (Collective Acceleration Unit
HFI (Host Fabric Interface)
ISR (Integrated Switch/ Router)
1
7
1
24
1
16
8
1
n
1
N
1
full direct connectivity between Quad Modules via L-links
"local" L-links within Drawers
"remote" L-links between Drawers
Full direct Connectivity between Supernodes via D-links
D-links

Figure 1-21 Logical view of a Power 775 system

Figure 1-22 shows an example configuration of a 242 TFLOP Power 775 cluster that uses eight Supernodes and direct graph interconnect. In this configuration, there are 28 D-Link cable paths to route and 1-64 12-lane 10 Gb D-Link cables per cable path.

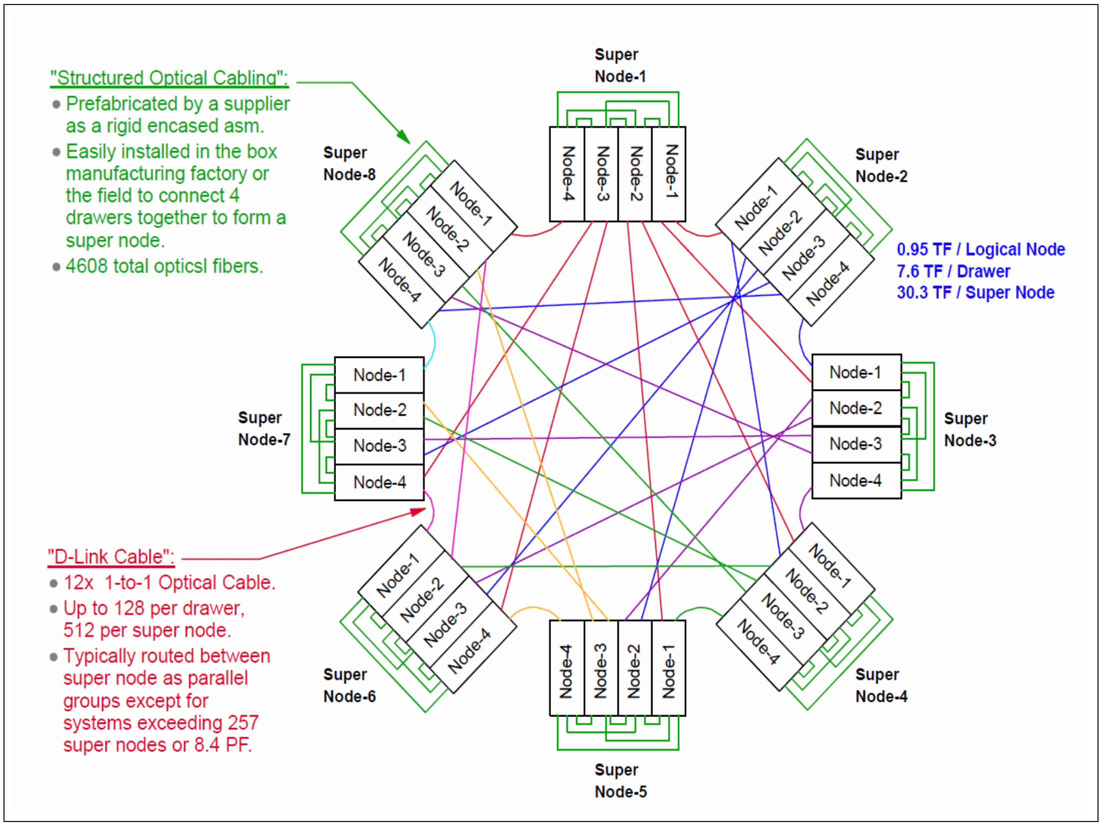
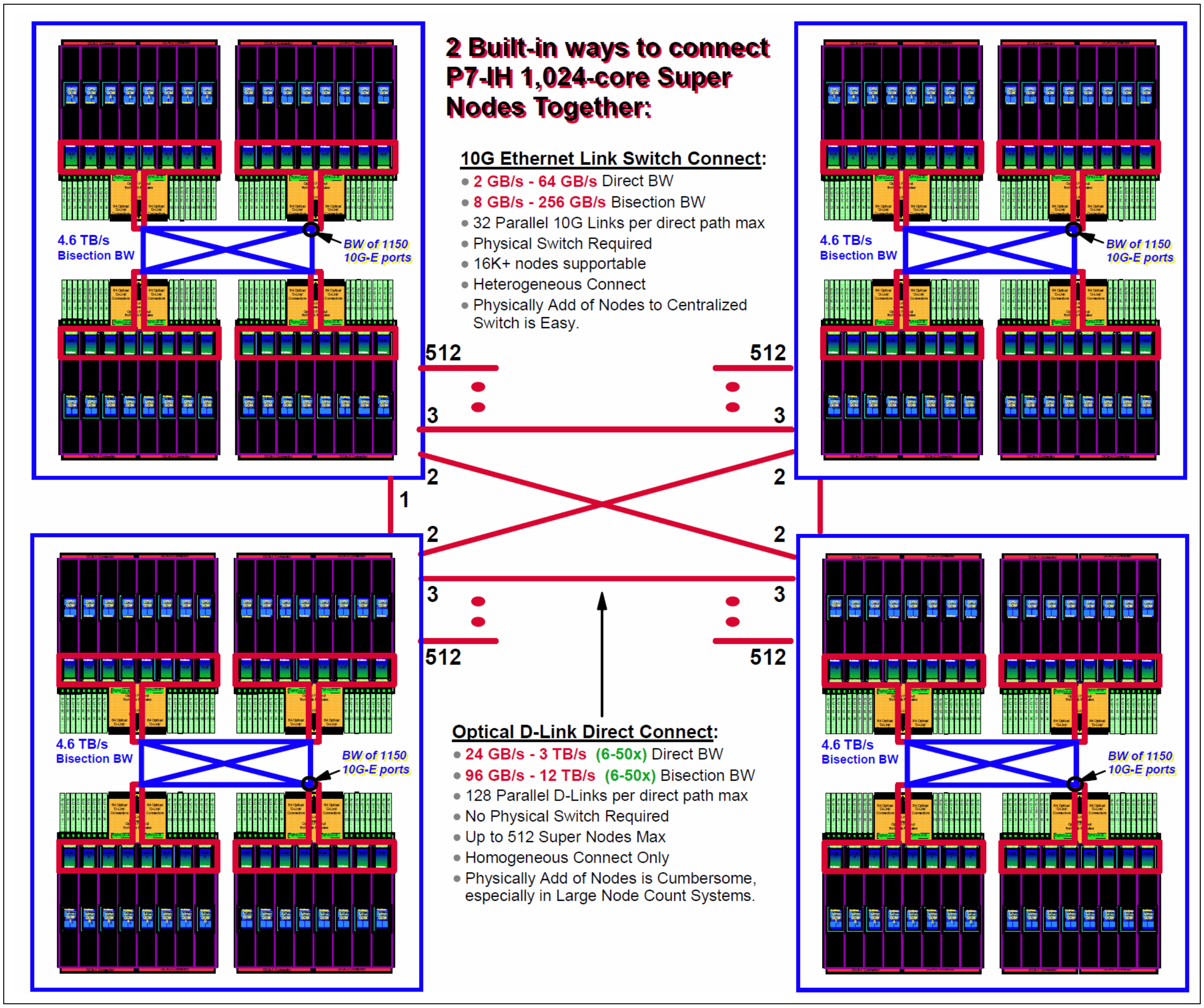

"Structured Optical Cabling":
● Prefabricated by a supplier as a rigid encased asm.
● Easily installed in the box manufacturing factory or the field to connect 4 drawers together to form a super node.
● 4608 total opticsl fibers.
"D-Link Cable":
● 12x 1-to-1 Optical Cable.
● Up to 128 per drawer, 512 per super node.
● Typically routed between super node as parallel groups except for systems exceeding 257 super nodes or 8.4 PF.
0.95 TF / Logical Node
7.6 TF / Drawer
30.3 TF / Super Node
Super Node-1
Super Node-2
Super Node-3
Super Node-4
Super Node-5
Super Node-6
Super Node-7
Super Node-8
Node-1
Node-2
Node-3
Node-4

Figure 1-22 Direct graph interconnect example

A 4,096 core (131 TF), fully interconnected system is shown in Figure 1-23.



2 Built-in ways to connect P7-IH 1,024-core Super Nodes Together:

10G Ethernet Link Switch Connect:

- 2 GB/s - 64 GB/s Direct BW
- 8 GB/s - 256 GB/s Bisection BW
- 32 Parallel 10G Links per direct path max
- Physical Switch Required
- 16K+ nodes supportable
- Heterogeneous Connect
- Physically Add of Nodes to Centralized Switch is Easy.

Optical D-Link Direct Connect:

- 24 GB/s - 3 TB/s (6-50x) Direct BW
- 96 GB/s - 12 TB/s (6-50x) Bisection BW
- 128 Parallel D-Links per direct path max
- No Physical Switch Required
- Up to 512 Super Nodes Max
- Homogeneous Connect Only
- Physically Add of Nodes is Cumbersome, especially in Large Node Count Systems.

Figure 1-23 Fully interconnected system example

Network topology

Optical D-links connect Supernodes in different connection patterns. Figure 1-24 on page 36 shows an example of 32 D-links between each pair of supernodes. Topology is 32D, a connection pattern that supports up to 16 supernodes.

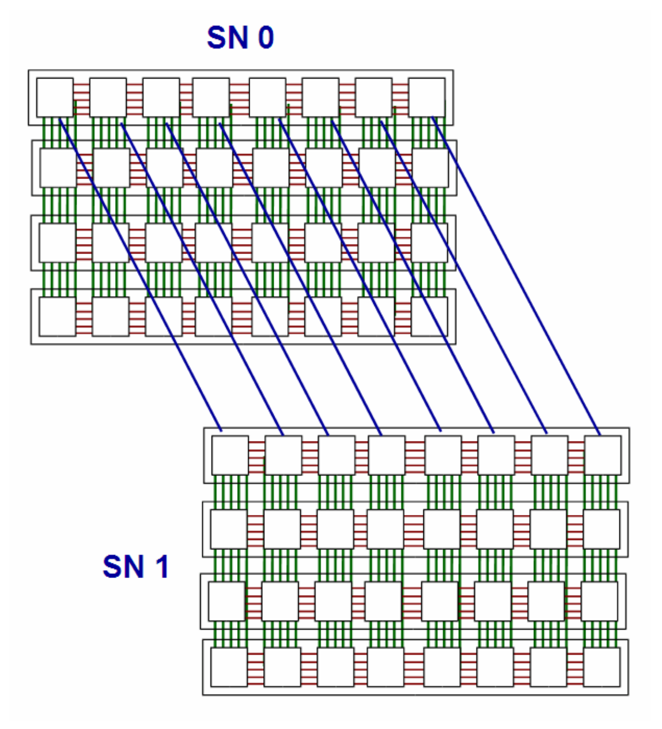


Figure 1-24 Supernode connection using 32D topology

Figure 1-25 shows another example in which there is one D-link between supernode pairs, which supports up to 512 supernodes in a 1D topology.

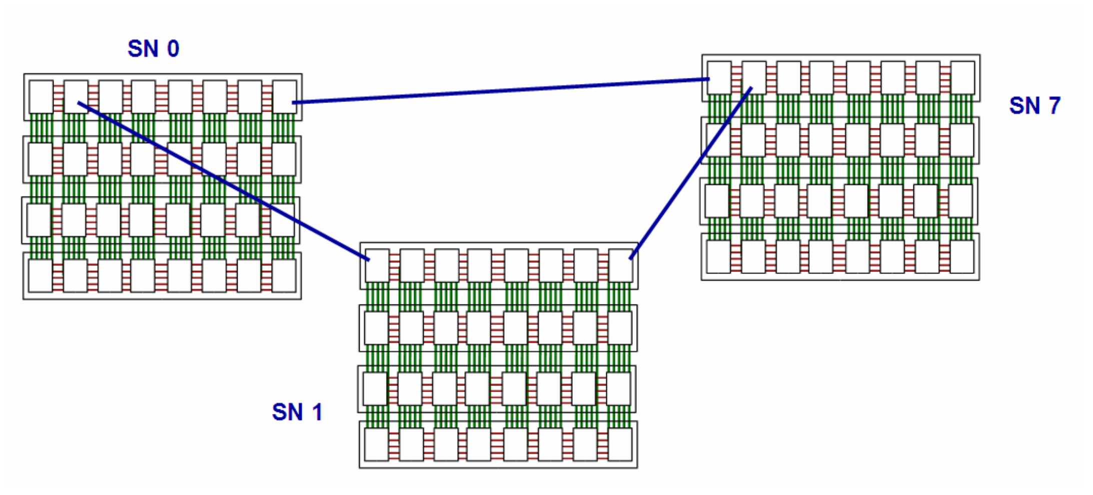


Figure 1-25 1D network topology

The network topology is specified during the installation. A topology specifier is set up in the cluster database. In the cluster DB site table, topology=<specifier>. Table 1-4 shows the supported four-drawer and single-drawer Supernode topologies.

Table 1-4 Supported four-drawer and single-drawer Supernode topologies

Topology	Maximum number of supernodes
256D	3
128D	5
64D	8
32D	16
16D	32
8D	64
4D	128
2D	256
1D	512
Single-Drawer Supernode topologies	
2D_SDSN	48
8D_SDSN	12

ISR network routes

Each ISR includes a set of hardware route tables. The Local Network Management Controller (LNMC) routing code generates and maintains the routes with help from Central Network Manage (CNM), as shown in Figure 1-26 on page 38. These route tables are set up during system initialization and are dynamically adjusted as links go down or come up during operation. Packets are injected into the network with a destination identifier and the route mode. The route information is picked up from the route tables along the route path that is based on this information. Packets that is injected into the interconnect by the HFI employ source route tables with the route partially determined. Per-port route tables are used to route packets along each hop in the network. Separate route tables are used for intersupernode and intrasupernode routes.

Routes are classified as *direct* or *indirect*. A direct route uses the shortest path between any two compute nodes in a system. There are multiple direct routes between a set of compute nodes because a pair of supernodes are connected by more than one D-link.

The network topology features two levels and therefore the longest direct route has three hops (no more than two L hops and at most one D hop). This configuration is called an L-D-L route.

The following conditions exist when source and destination hubs are within a drawer:

- The route is one L-hop (assuming all of the links are good).
- LNMC needs to know only the local link status in this CEC.

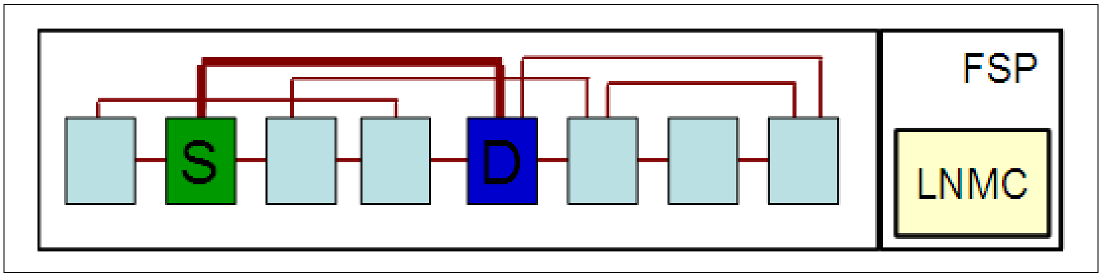


Figure 1-26 Routing within a single CEC

The following conditions exist when source and destination hubs lie within a supernode, as shown in Figure 1-27:

- Route is one L-hop (every hub within a supernode is directly connected via Lremote link to every other hub in the supernode).
- LNMC needs to know only the local link status in this CEC.

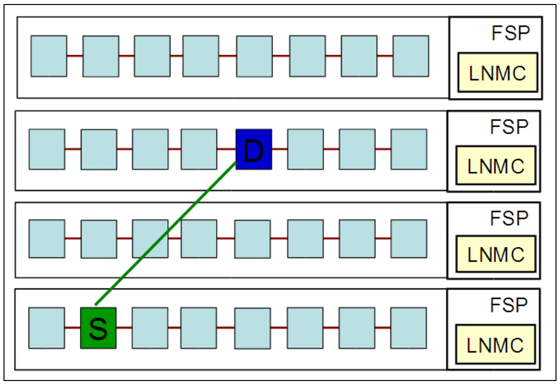


Figure 1-27 L-hop

If an L-remote link is faulty, the route requires two hops. However, only the link status local to the CEC is needed to construct routes, as shown in Figure 1-28.

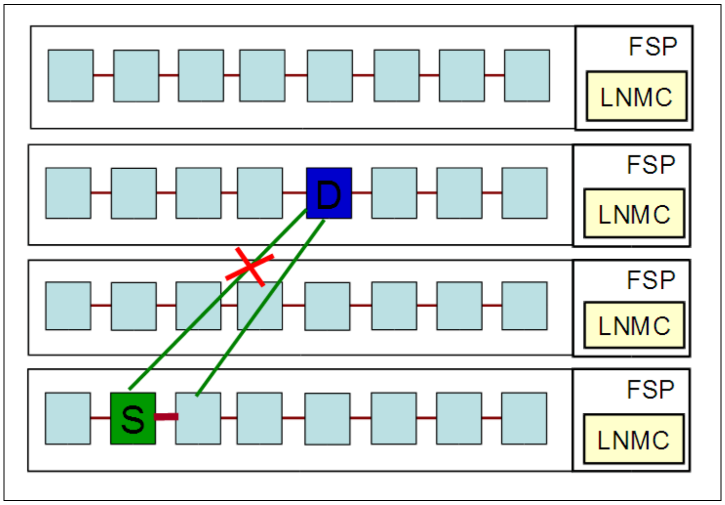


Figure 1-28 Route representation in event of a faulty Lremote link

When source and destination hubs lie in different supernodes, as shown in Figure 1-29, the following conditions exist:

- Route possibilities: one D-hop, or L-D (L-D-L routes also are used)
- LNMC needs non-local link status to construct L-D routes

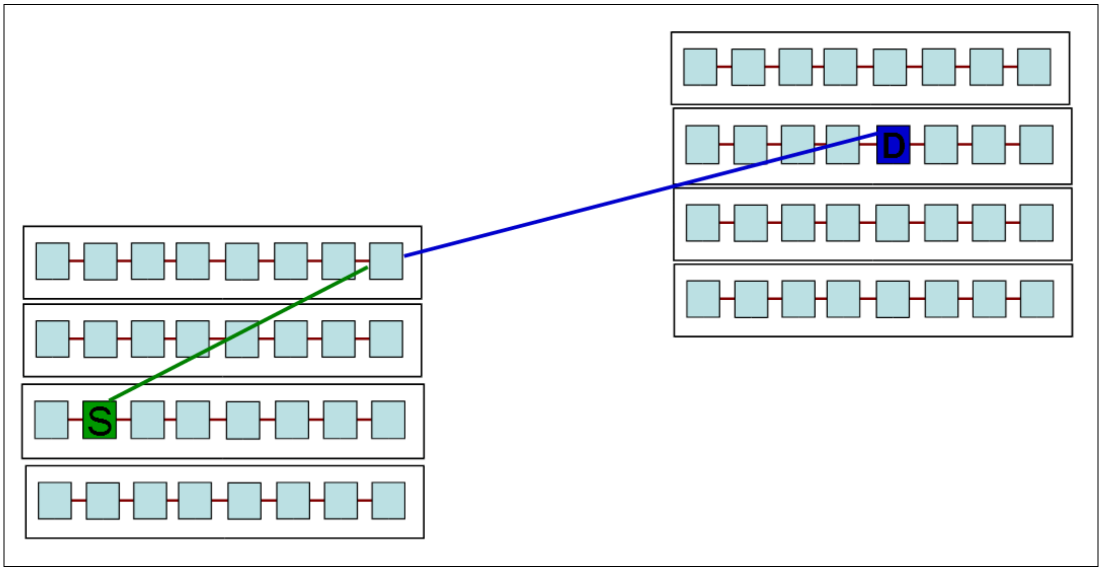


Figure 1-29 L-D route example

The ISR also supports indirect routes to provide increased bandwidth and to prevent hot spots in the interconnect. An indirect route is a route that has an intermediate compute node in the route that is on a different supernode, not the same supernode in which source and compute nodes reside. An indirect route must employ the shortest path from the source compute node to the intermediate node, and the shortest path from the intermediate compute node to the destination compute node. Although the longest indirect route has five hops at most, no more than three hops are L hops and two hops (at most) are D hops. This configuration often is represented as an L-D-L-D-L route.

The following methods are used to select a route is when multiples routes exist:

- Software specifies the intermediate supernode, but the hardware determines how to route to and then route from the intermediate supernode.
- The hardware selects among the multiple routes in a round-robin manner for both direct and indirect routes.
- The hub chip provides support for route randomization in which the hardware selects one route between a source–destination pair. Hardware-directed randomized route selection is available only for indirect routes.

These routing modes are specified on a per-packet basis.

The correct choice between the use of direct- versus indirect-route modes depends on the communication pattern that us used by the applications. Direct routing is suitable for communication patterns in which each node must communicate with many other nodes by using spectral methods. Communication patterns that involve small numbers of compute nodes benefit from the extra bandwidth that is offered by the multiple routes with indirect routing.

1.5 Power, packaging, and cooling

This section provides information about the IBM Power Systems 775 power, packaging, and cooling features.

1.5.1 Frame

The front view of an IBM Power Systems 775 frame is shown in Figure 1-30.

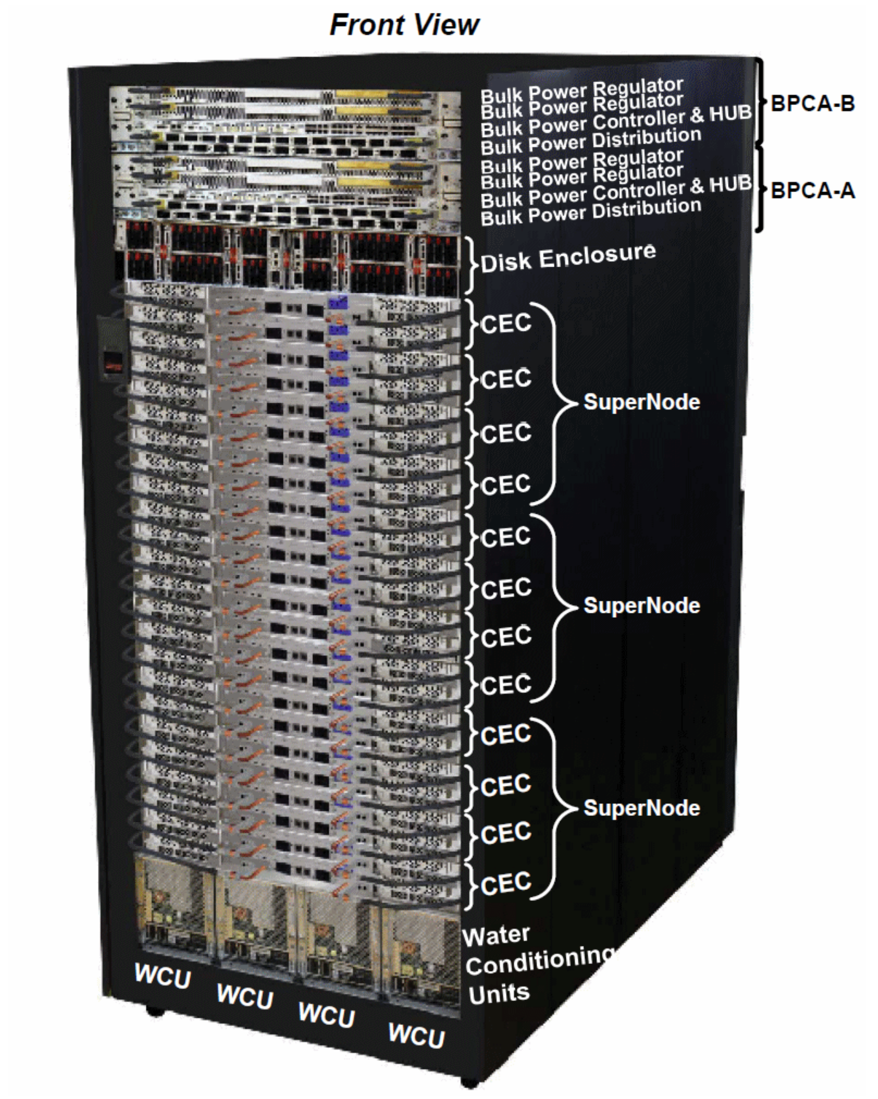


Figure 1-30 Power 775 frame

The Power 775 frame front view is shown in Figure 1-31 on page 41.

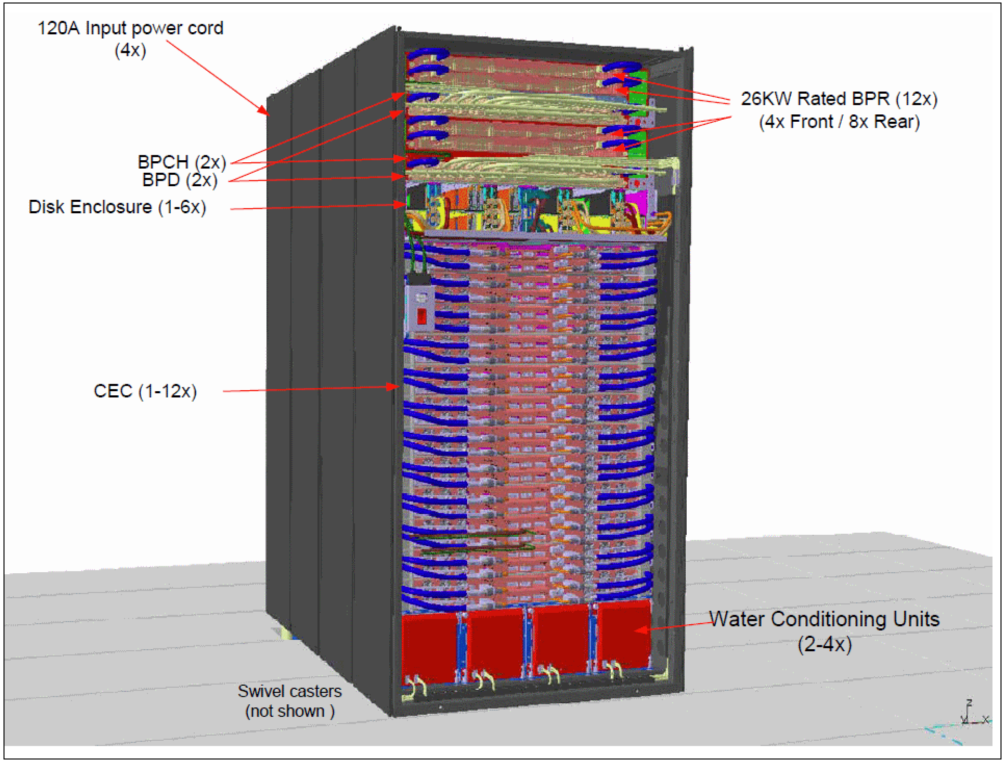


Figure 1-31 Frame front view

The rear view of the Power 775 frame is shown in Figure 1-32 on page 42.

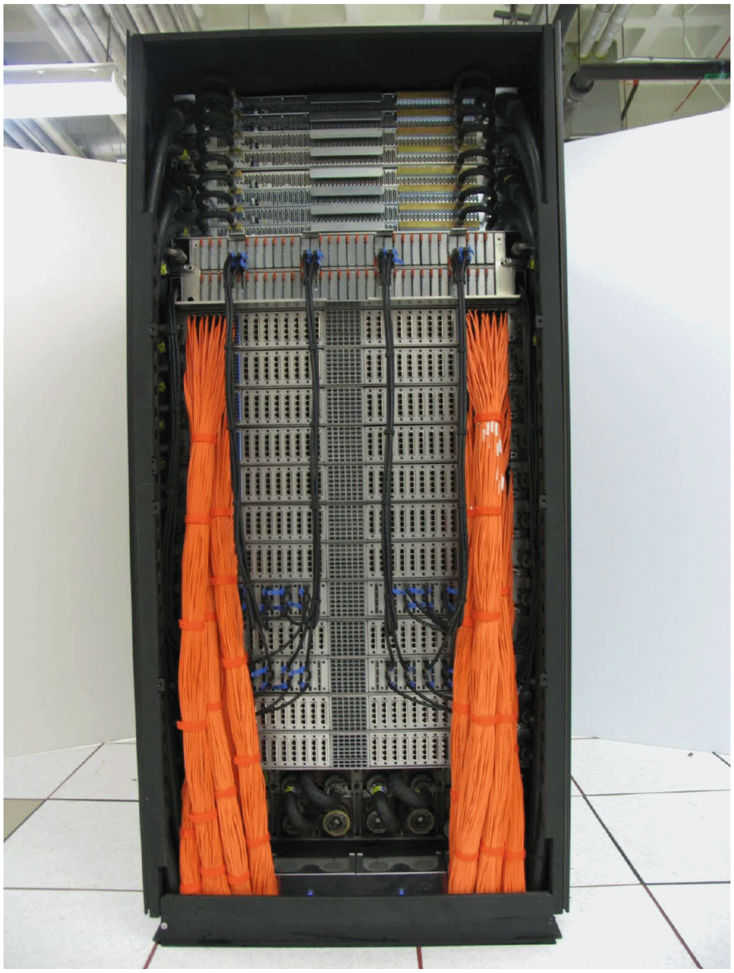


Figure 1-32 Frame rear photo

1.5.2 Bulk Power and Control Assembly

Each Bulk Power and Control Assembly (BPCA) is a modular unit that includes the following features:

- Bulk Power and Control Enclosure (BPCE)

  Contains two 125A 3-phase AC couplers, six BPR bays, one BPCH and one BPD bay.

- Bulk Power Regulators (BPR), each rated at 27 KW@-360 VDC

  One to six BPRs are populated in the BPCE depending on CEC drawers and storage enclosures in frame, and the type of power cord redundancy that is wanted.

- Bulk Power Control and Communications Hub (BPCH)

  This unit provides rack-level control, storage enclosures, and water conditioning units (WCUs), and concentration of the communications interfaces to the unit level controllers that are in each server node, storage enclosure, and WCU.

- Bulk Power Distribution (BPD)

  This unit distributes 360 VDC to server nodes and disk enclosures.

- Power Cords

One per BPCE for populations of one to three BPRs/BPCE and two per BPCE for four to six BPRs/BPCE.

The front and rear views of the BPCA are shown in Figure 1-33.

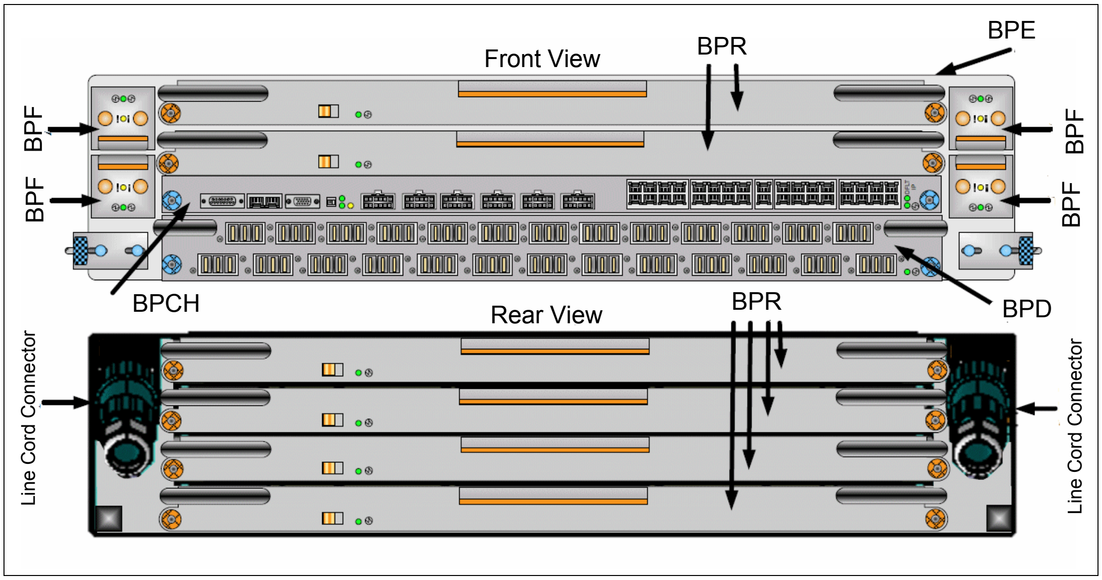


Figure 1-33 BPCA

The minimum configuration per BPCE is one x BPCH, one x BPD, one x BPR and one line core. There are always two BPCAs above one another in the top of the cabinet.

BPRs are added uniformly to each BPCE depending on the power load in the rack.

A single fully configured BPCE provides 27 KW x 6 = 162 KW of Bulk Power, which equates to aggregate system power cord power of approximately 170 KW. Up to this power level, bulk power is in a 2N arrangement, where a single BPCE is removed entirely for maintenance concurrently with the rack that remains fully operational. If the rack bulk power demand exceeds 162 KW, the bulk power provides an N+1 configuration of up to 27 KW x 9 = 243 KW, which equates to aggregate system power cord power of approximately 260 KW. N+1 Bulk Power mode means one of the four power cords are disconnected and the cabinet continues to operate normally. BPCE concurrent maintenance is not conducted in N+1 bulk power mode, unless the rack bulk power load is reduced to less than 162 KW by invoking Power Efficient Mode on the server nodes. This mode reduces the peak power demand at the expense of reducing performance.

Cooling

The BPCA is nearly entirely water cooled. Each BPR and the BPD has two quick connects to enable connection to the supply and return water cooling manifolds in the cabinet. All of the components that dissipate any significant power in the BPR and BPD are heat sunk to a water-cooled cold plate in these units.

To assure the ambient air temperature internal to the BP, BPCH, and BPD enclosures is kept low, two hot pluggable blowers are installed in the rear of each BPCA in an N+1 speed controlled arrangement. These blowers flush the units to keep the temperature internal at approximately system inlet air temperature, which is 40 degrees-C maximum. A fan is replaced concurrently.

Management

The BPCH provides the mechanism to connect the cabinet to the management server via a 1 Gb Ethernet out-of-band network. Each BPCH has two management server-facing 1 Gb Ethernet ports or buses so that the BPCH connects to a fully redundant network.

1.5.3 Bulk Power Control and Communications Hub

The front view of the bulk power control and communications hub (BPCH) is shown in Figure 1-34.

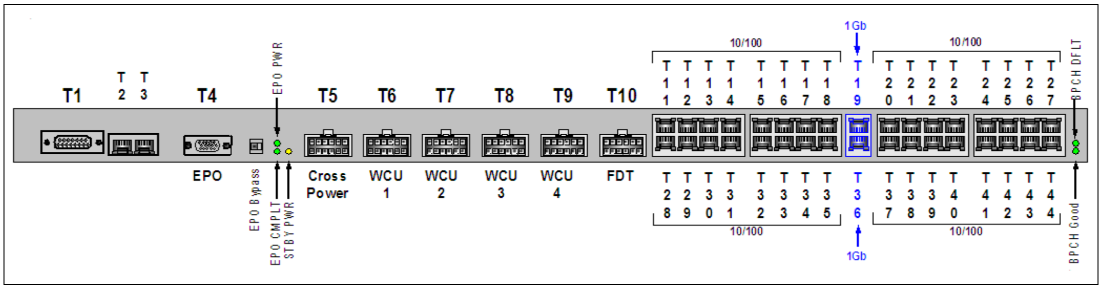


Figure 1-34 BPCH front view

The following connections are shown in Figure 1-34:

- T2: 10/100 Mb Ethernet from HMC1.
- T3: 10/100 Mb Ethernet from HMC2.
- T4: EPO.
- T5: Cross Power.
- T6-T9: RS422 UPIC for WCUs.
- T10: RS422 UPIC port for connection of the Fill and Drain Tool.
- T19/T36: 1 Gb Ethernet for HMC connectors (T19, T36).
- 2x – 10/100 Mb Ethernet port to plug in a notebook while the frame is serviced.
- 2x – 10/100 Mb spare (connectors contain both eNet and ½ Duplex RS422).
- T11-T19, T20-T35, T37-T44: 10/100 Mb Ethernet ports or buses to the management processors in CEC drawers and storage enclosures. Max configuration supports 12 CEC drawers and one storage enclosure in the frame (connectors contain both eNet and ½ Duplex RS422).

1.5.4 Bulk Power Regulator

This section describes the bulk power regulator (BPR).

Input voltage requirements

The BPR supports DC and AC input power for the Power 775. A single design accommodates both the AC and DC range with different power cords for various voltage options.

DC requirements

The BPR features the following DC requirements:

- The Bulk Power Assembly (BPA) is capable of operating over a range of 300 to 600 VDC
- Nominal operating DC points are 375 VDC and 575 VDC

AC requirements

Table 1-5 shows the AC electrical requirements.

Table 1-5 AC electrical requirements

Input configuration	Three phase and GND (no neutral)	Three phase and GND (no neutral)
Rated nominal voltage and frequency	200 to 240 Vac @ 50 to 60Hz	380 to 480 Vac @50 to 60Hz
Rated current (amps per phase)	125A	100A
Acceptable voltage tolerance @ machine power cord	180 - 259Vac	333 - 508Vac
Acceptable frequency tolerance @ machine power cord	47 - 63Hz	47 - 63Hz

1.5.5 Water conditioning unit

The Power 755 WCU system is shown in Figure 1-35.

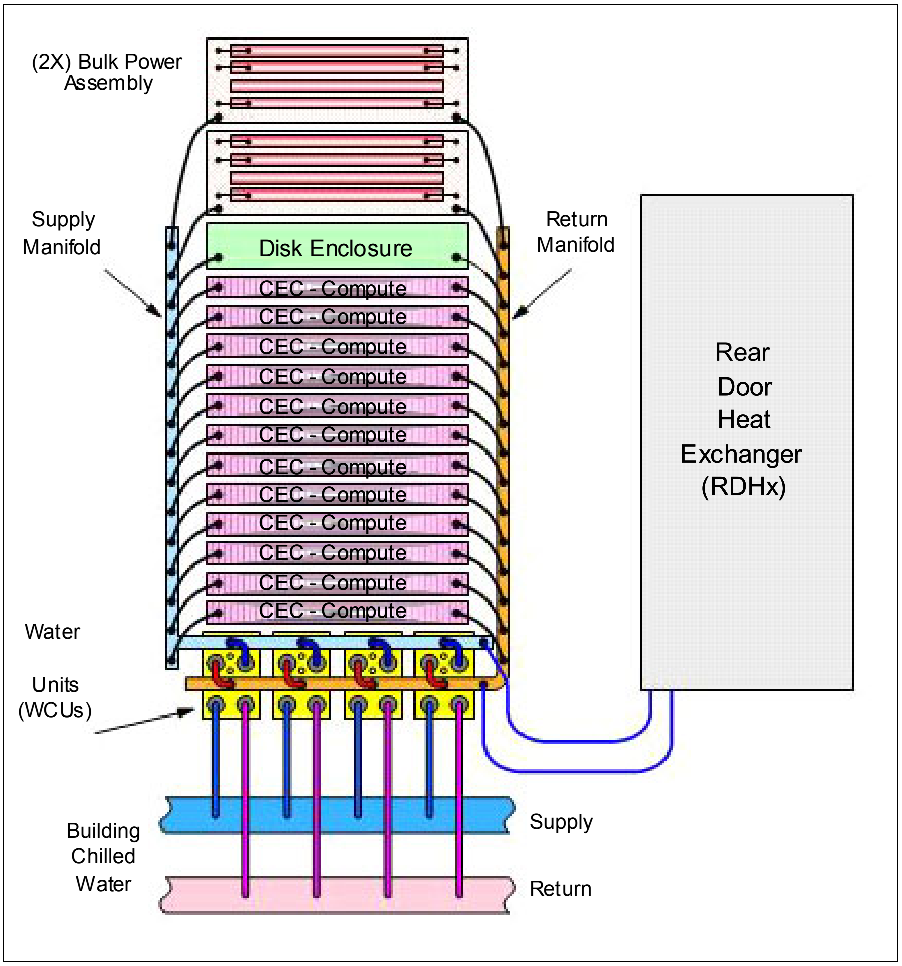


Figure 1-35 Power 755 water conditioning unit system

The hose and manifolds assemblies and WCUs are shown in Figure 1-36.

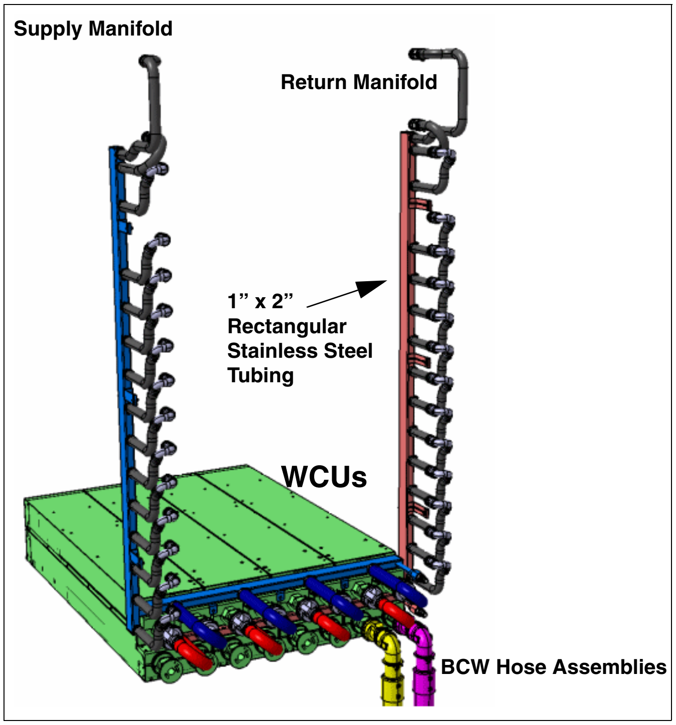


Figure 1-36 Hose and manifold assemblies

The components of the WCU are shown in Figure 1-37 on page 47.

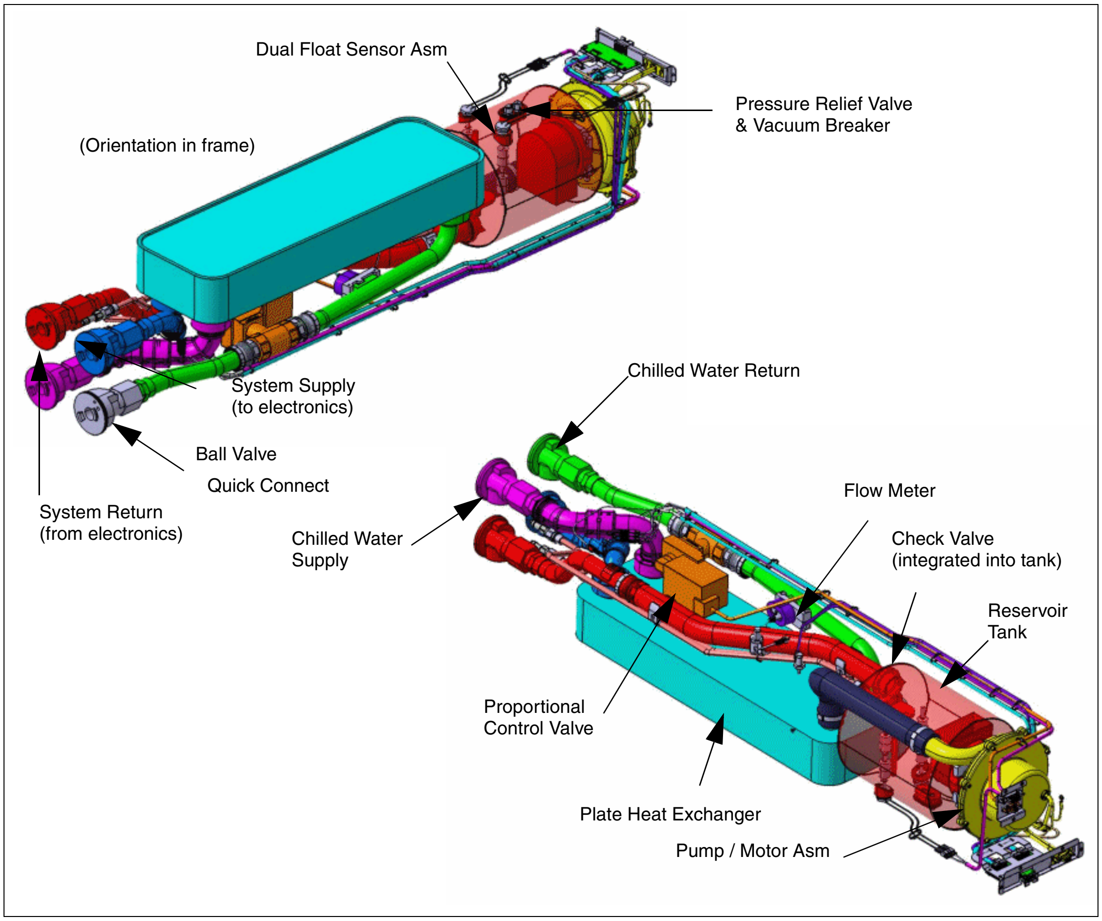


Figure 1-37 WCU components

The WCU schematics are shown in Figure 1-38.

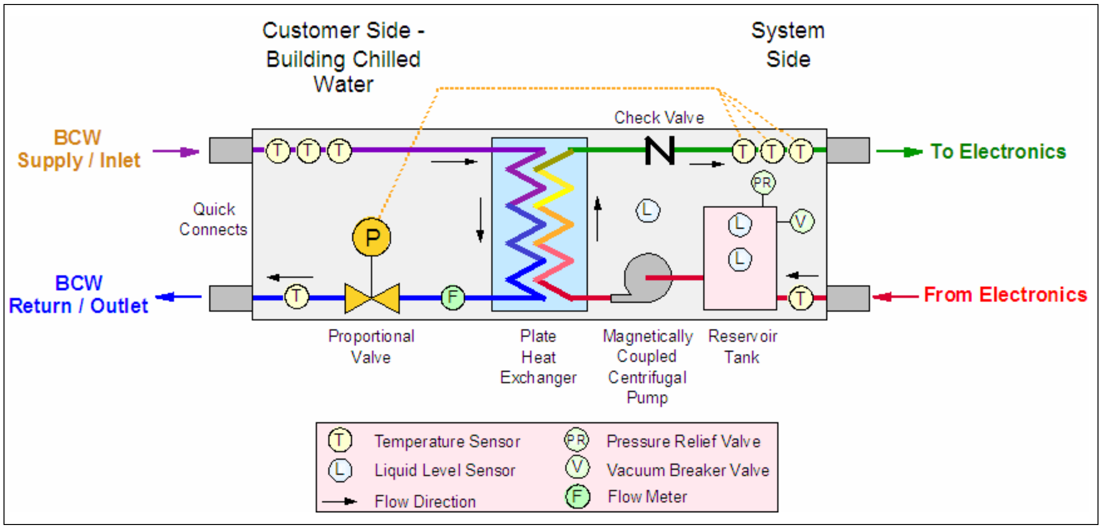


Figure 1-38 WCU schematics

1.6 Disk enclosure

This section describes the storage disk enclosure for the Power 775 system.

1.6.1 Overview

The Power 775 system features the following disk enclosures:

- SAS Expander Chip (SEC):
  - # PHYs: 38
  - Each PHY capable of SAS SDR or DDR
- 384 SFF DASD drives:
  - 96 carriers with four drives each
  - 8 Storage Groups (STOR 1-8) with 48 drives each:
    - 12 carriers per STOR
    - Two Port Cards per STOR each with 3 SECs
  - 32 SAS x4 ports (four lanes each) on 16 Port Cards.
- Data Rates:
  - Serial Attach SCSI (SAS) SDR = 3.0 Gbps per lane (SEC to Drive)
  - Serial Attach SCSI (SAS) DDR= 6.0 Gbps per lane (SAS Adapter in Node to SEC)
- The drawer supports 10 K/15 K rpm drives in 300 Gb or 600 Gb sizes.
- A Joint Test Action Group (JTAG) interface is provided from the DC converter assemblies (DCAs) to each SEC for error diagnostics and boundary scan.

> **Important:** STOR is the short name for storage group (it is not an acronym).

The front view of the disk enclosure is shown in Figure 1-39 on page 49.

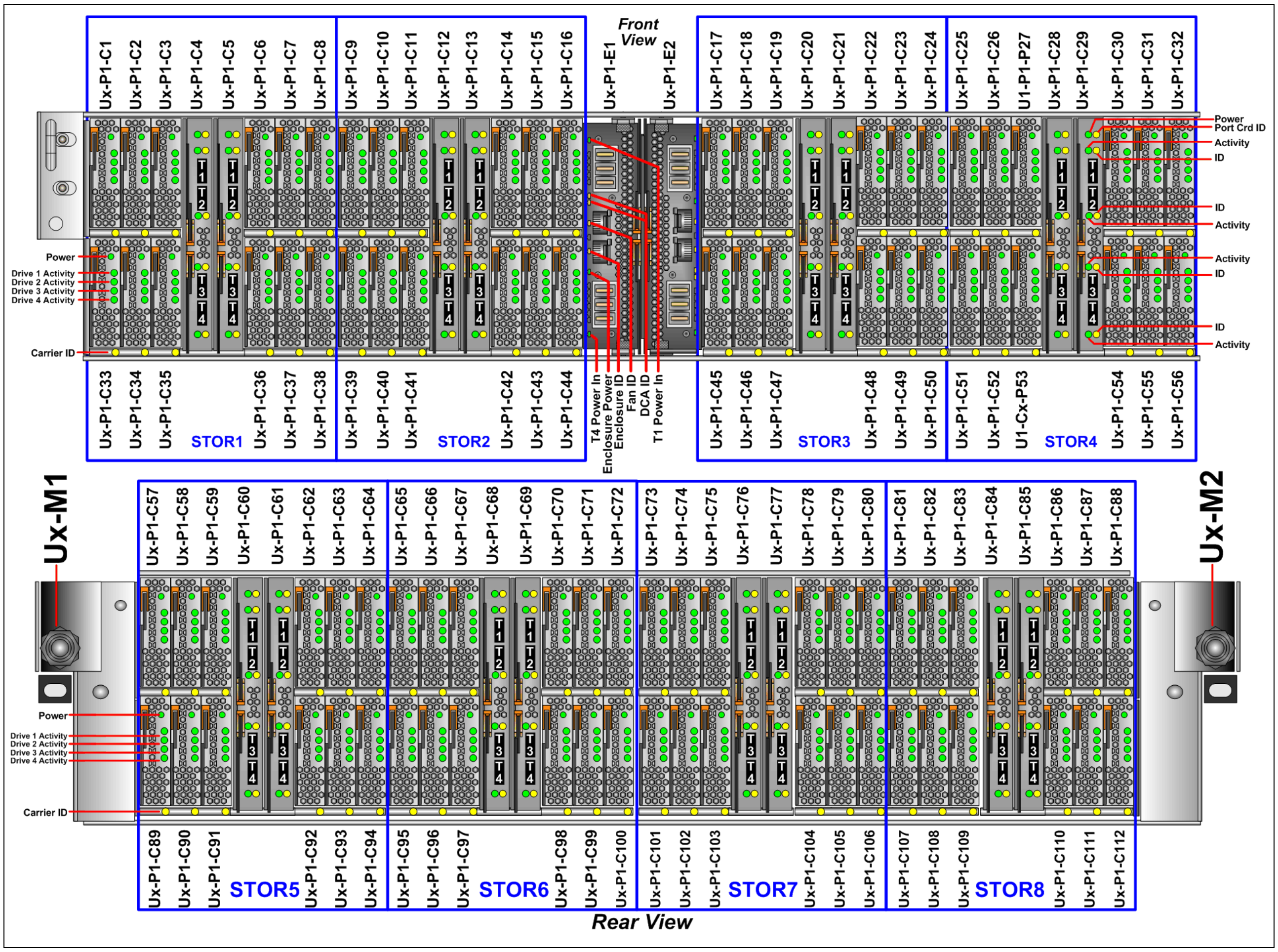


Figure 1-39   Disk enclosure front view

1.6.2 High-level description

Figure 1-40 on page 50 represents the top view of a disk enclosure and highlights the front view of a STOR. Each STOR includes 12 carrier cards (six at the top of the drawer and six at the bottom of the drawer) and two port cards.

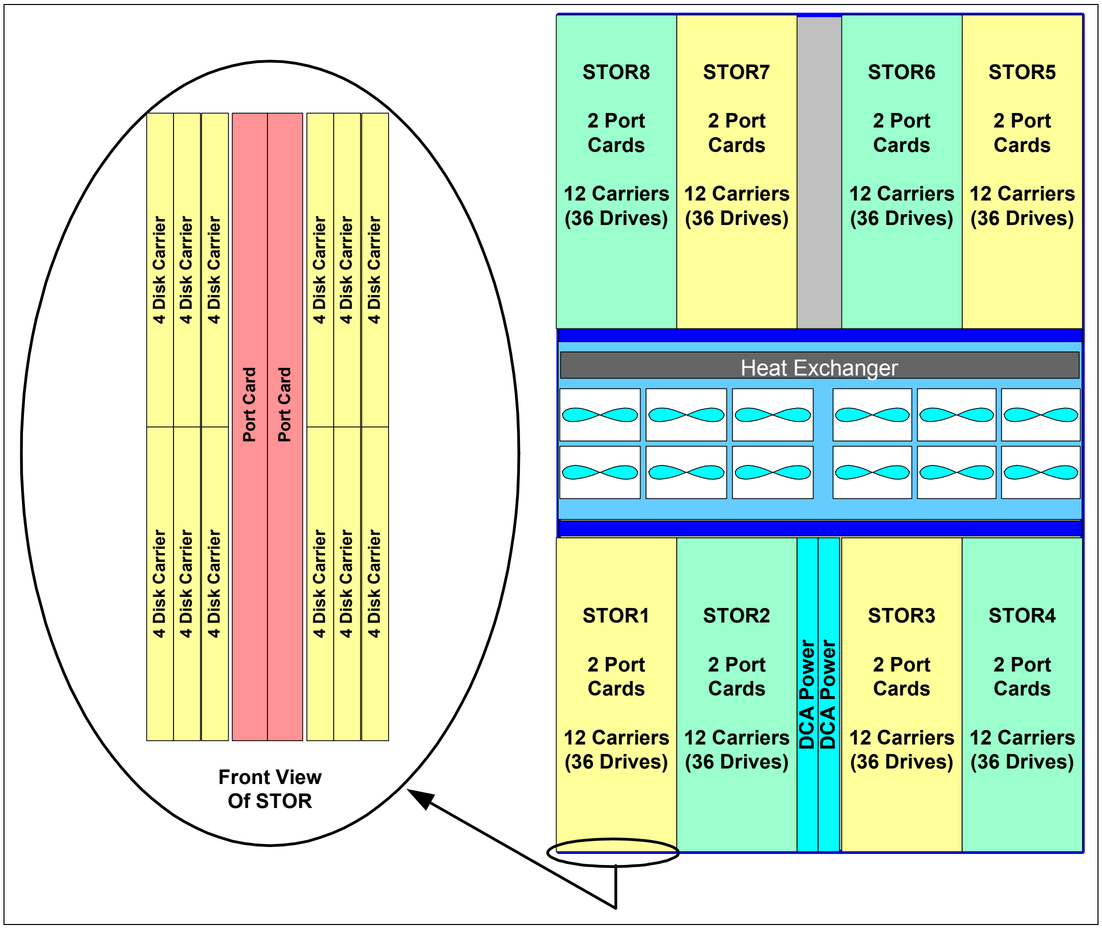


Figure 1-40 Storage drawer top view

The disk enclosure is a SAS storage drawer that is specially designed for the IBM Power 775 system. The maximum storage capacity of the drawer is 230.4 TB, distributed over 384 SFF DASD drives logically organized in eight groups of 48.

The disk enclosure feature two mid-plane boards that comprise the inner core assembly. The disk drive carriers, port cards, and power supplies plug into the mid-plane boards. There are four Air Moving Devices (AMD) in the center of the drawer. Each AMD consists of three counter-rotating fans.

Each carrier contains connectors for four disk drives. The carrier features a solenoid latch that is released only through a console command to prevent accidental unseating. The disk carriers also feature LEDs close to each drive and a gold capacitor circuit so that drives are identified for replacement after the carrier is removed for service.

Each port card includes four SAS DDR 4x ports (four lanes at 6 Gbps/lane). These incoming SAS lanes connect to the input SEC, which directs the SAS traffic to the drives. Each drive is connected to one of the output SECs on the port card with SAS SDR 1x (one lane @ 6 Gbps). There are two port cards per STOR. The first Port card connects to the A ports of all 48 drives in the STOR. The second Port card connects to the B ports of all 48 drives in the STOR. The port cards include soft switches for all 48 drives in the STOR (5 V and 12 V soft switches connect and interrupt and monitor power). The soft switch is controlled by I2C from the SAS Expander Chip (SEC) on the port card.

A fully cabled drawer includes 36 cables: four UPIC power cables and 32 SAS cables from SAS adapters in the CEC. During service to replace a power supply, two UPIC cables manage the current and power control of the entire drawer. During service of a port card, the second port card in the STOR remains cabled to the CEC so that the STOR remains operational. A customer minimum configuration is two SAS cables per STOR and four UPIC power cables per drawer to ensure proper redundancy.

1.6.3 Configuration

A disk enclosure must reside in the same frame as the CEC to which it is cabled. A frame might contain up to six Disk Enclosures. The disk enclosure front view is shown in Figure 1-41.

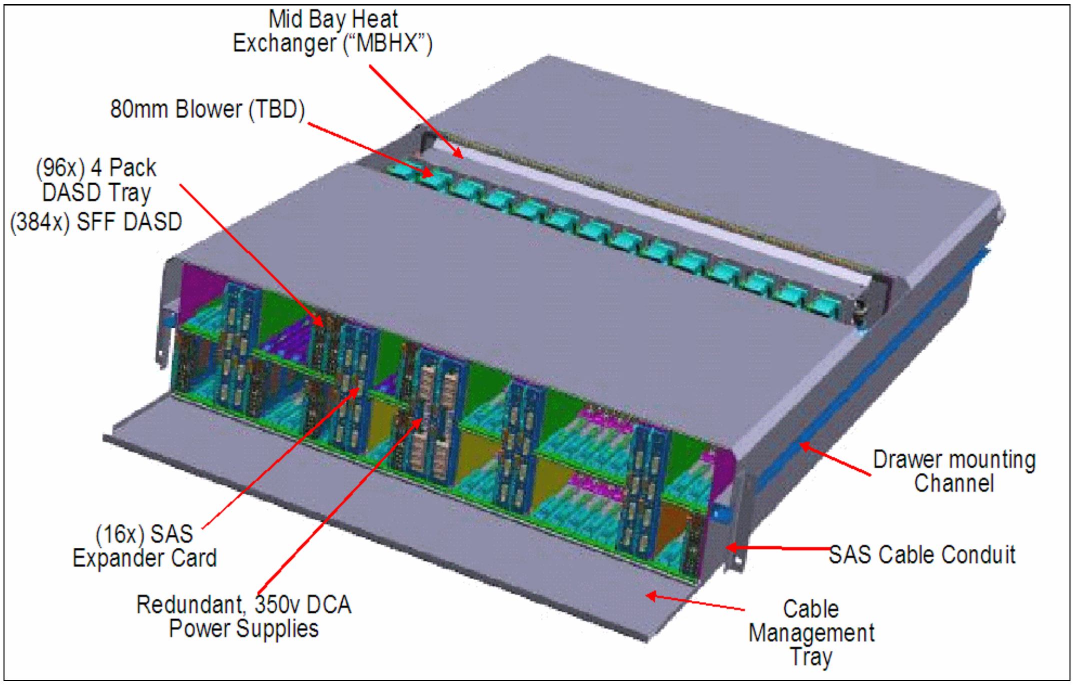


Figure 1-41 Disk Enclosure front view

The disk enclosure internal view is shown in Figure 1-42 on page 52.

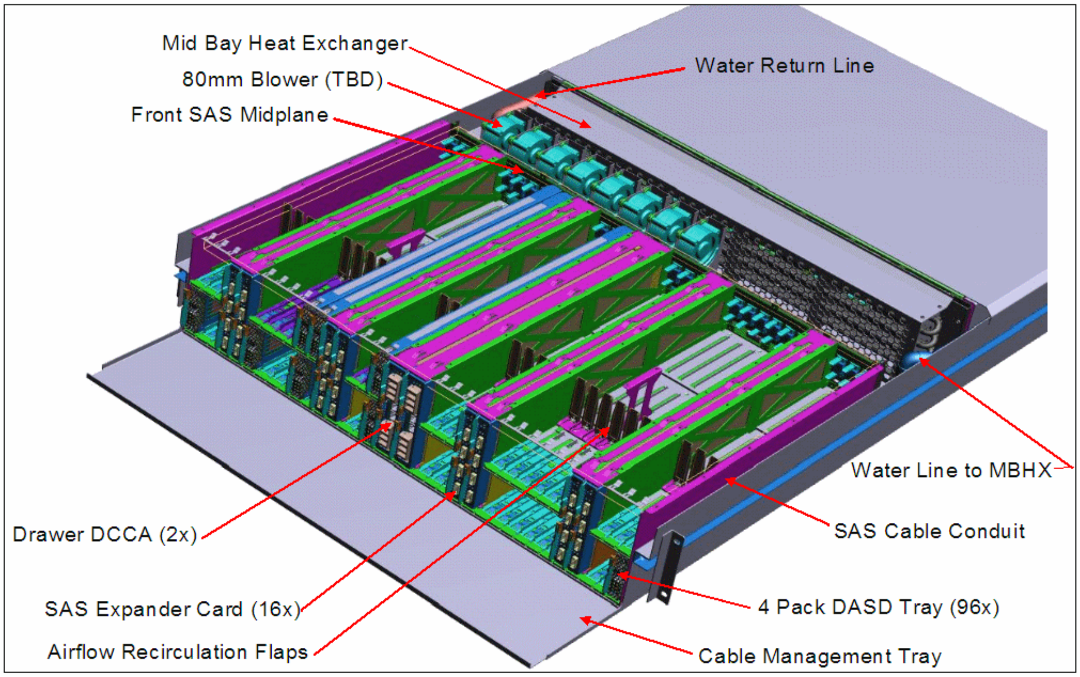


Figure 1-42 Disk Enclosure internal view

The disk carrier is shown in Figure 1-43.

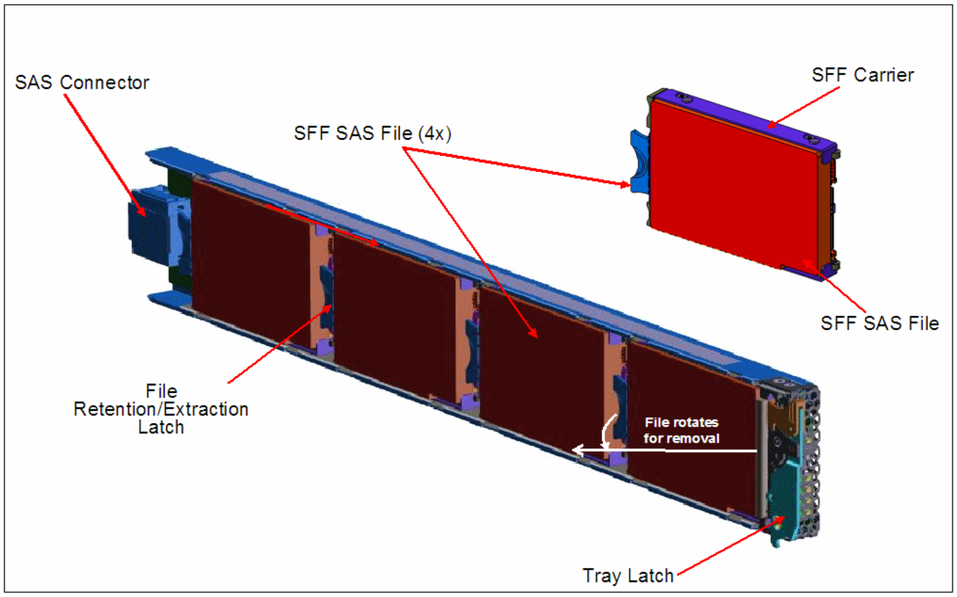


Figure 1-43 Disk carrier

The disk enclosure includes the following features:

- A disk enclosure is one quarter, one half, three quarters, or fully populated with HDDs and eight SSDs. The disk enclosure always is populated with eight SSDs.
- Disk enclosure contains two GPFS recovery groups (RGs). The carriers that hold the disks of the RGs are distributed throughout all of the STOR domains in the drawer.
- A GPFS recovery group consists of four SSDs and one to four declustered arrays (DAs) of 47 disks each.
- Each DA contains distributed spare space that is two disks in size.
- Every DA in a GPFS system must be the same size.
- The granularity of capacity and throughput is an entire DA.
- RGs in the GPFS system do not need to be the same size.

1.7 Cluster management

The cluster management hardware that supports the Cluster is placed in 42 U, 19-inch racks. The cluster management requires Hardware Management Consoles (HMCs), redundant Executive Management Servers (EMS), and the associated Ethernet network switches.

1.7.1 Hardware Management Console

The HMC runs on a single server and is used to help manage the Power 775 servers. The traditional HMC functions for configuring and controlling the servers are done via xCAT. For more information, see 1.9.3, “Extreme Cluster Administration Toolkit” on page 72.

The HMC is often used for the following tasks:

- During installation
- For reporting hardware serviceable events, especially through Electronic Service Agent™ (ESA), which is also commonly known as call-home
- By service personal to perform guided service actions

An HMC is required for every 36 CECs (1152 LPARs) and all Power 775 system have redundant HMCs. For every group of 10 HMCs, a spare HMC is in place. For example, if a cluster requires four HMCs, five HMCs are present. If a cluster requires 16 HMCs, the cluster has two HMCs to serve as spares.

1.7.2 Executive Management Server

The EMS is a standard 4U POWER7 entry-level server responsible for cluster management activities. EMSs often are redundant; however, a simplex configuration is supported in smaller Power 775 deployments.

At the cluster level, a pair of EMSs provide the following maximum management support:

- 512 frames
- 512 supernodes
- 2560 disk enclosures

The EMS is the central coordinator of the cluster from a system management perspective. The EMS is connected to, and manages, all cluster components: the frames and CECs, HFI/ISR interconnect, I/O nodes, service nodes, and compute nodes. The EMS manages these components through the entire lifecycle, including discovery, configuration, deployment, monitoring, and updating via private network Ethernet connections. The cluster administrator uses the EMS as their primary management cluster control point. The service nodes, HMCs, and Flexible Service Processors (FSPs) are mostly transparent to the system administrator, and therefore the cluster appears to be a single, flat cluster, despite the hierarchical management infrastructure to be deployed by using xCAT.

1.7.3 Service node

Systems management throughout the cluster is a hierarchical structure (see Figure 1-44) to achieve the scaling and performance necessary for a large cluster size. All the compute and I/O nodes in a building block are initially booted via the HFI and managed by a dedicated server that is called a *service node* (SN) in the utility CECs.

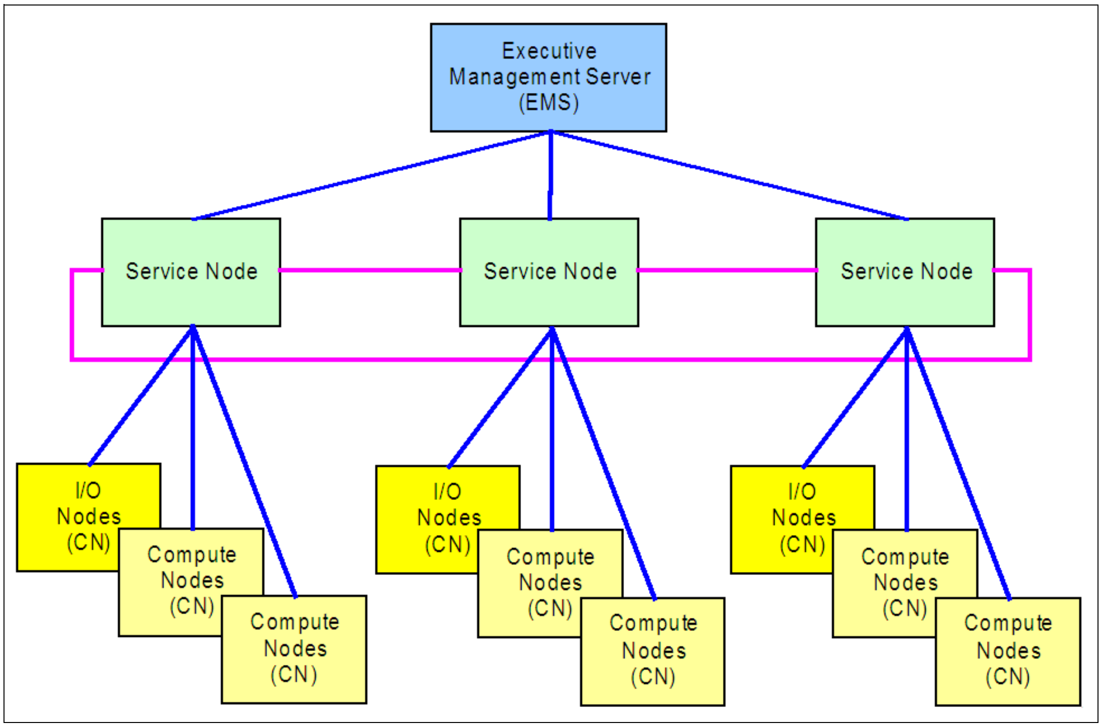


Figure 1-44 EMS hierarchy

Two service nodes (one for redundancy) per 36 CECs/Drawers (1 - 36) are required for all Power 775 clusters.

The two service nodes must reside in different frames, except under the following conditions:

- If there is only one frame, the nodes must reside in different super nodes in the frame.
- If there is only one super node in the frame, the nodes must reside in different CECs in the super node.
- If there are only two or three CEC drawers, the nodes must reside in different CEC drawers.
- If there is only one CEC drawer, the two Service nodes must reside in different octants.

The service node provides diskless boot and an interface to the management network. The service node requires that a PCIe SAS adapter and two 600 GB HDD PCIe form factor (in RAID1 for redundancy) must be installed to support diskless boot. The recommended location is shown on Figure 1-45. The SAS PCIe must reside in PCIe slot 16 and the HDDs in slots 15 and 14.

The service node also contains a 1 Gb Enet PCIe card that is in PCIe slot 17.

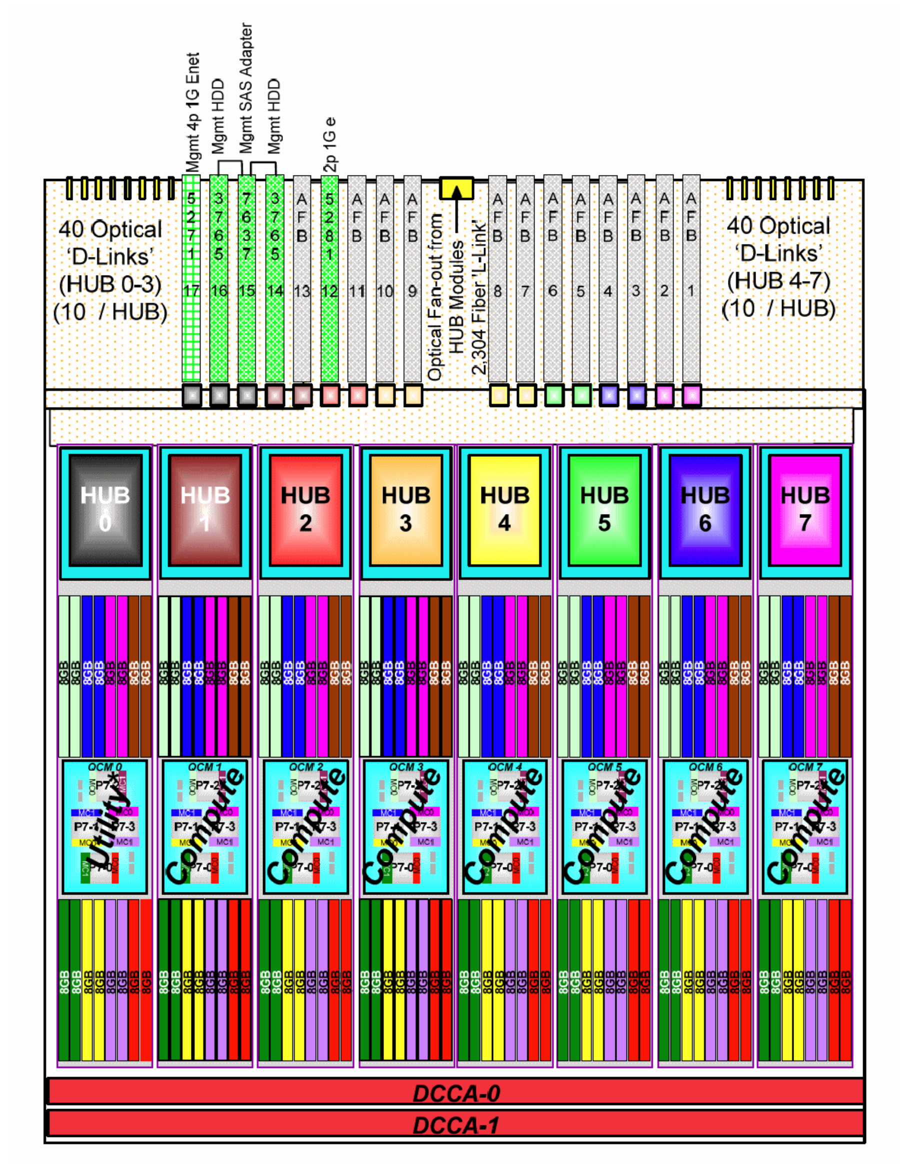


Figure 1-45 Service node

1.7.4 Server and management networks

Figure 1-46 on page 56 shows the logical structure of the two Ethernet networks for the cluster that is known as the *service* network and the *management* network. In Figure 1-46 on page 56, the black nets designate the service network and the red nets designate the management network.

The service network is a private, out-of-band network that is dedicated to managing the Power 775 clusters hardware. This network provides Ethernet based connectivity between the FSP of the CEC, the frame control BPA, the EMS, and the associated HMCs. Two identical network switches (ENET A and ENET B in the figure) are deployed to ensure high availability of these networks.

The management network is primarily responsible for booting all nodes, the designated service nodes, compute nodes, and I/O nodes, and monitoring their OS image loads. This management network connects the dual EMSs running the system management software with the various Power 775 servers of the cluster. Both the service and management networks must be considered private and not routed into the public network of the enterprise for security reasons.

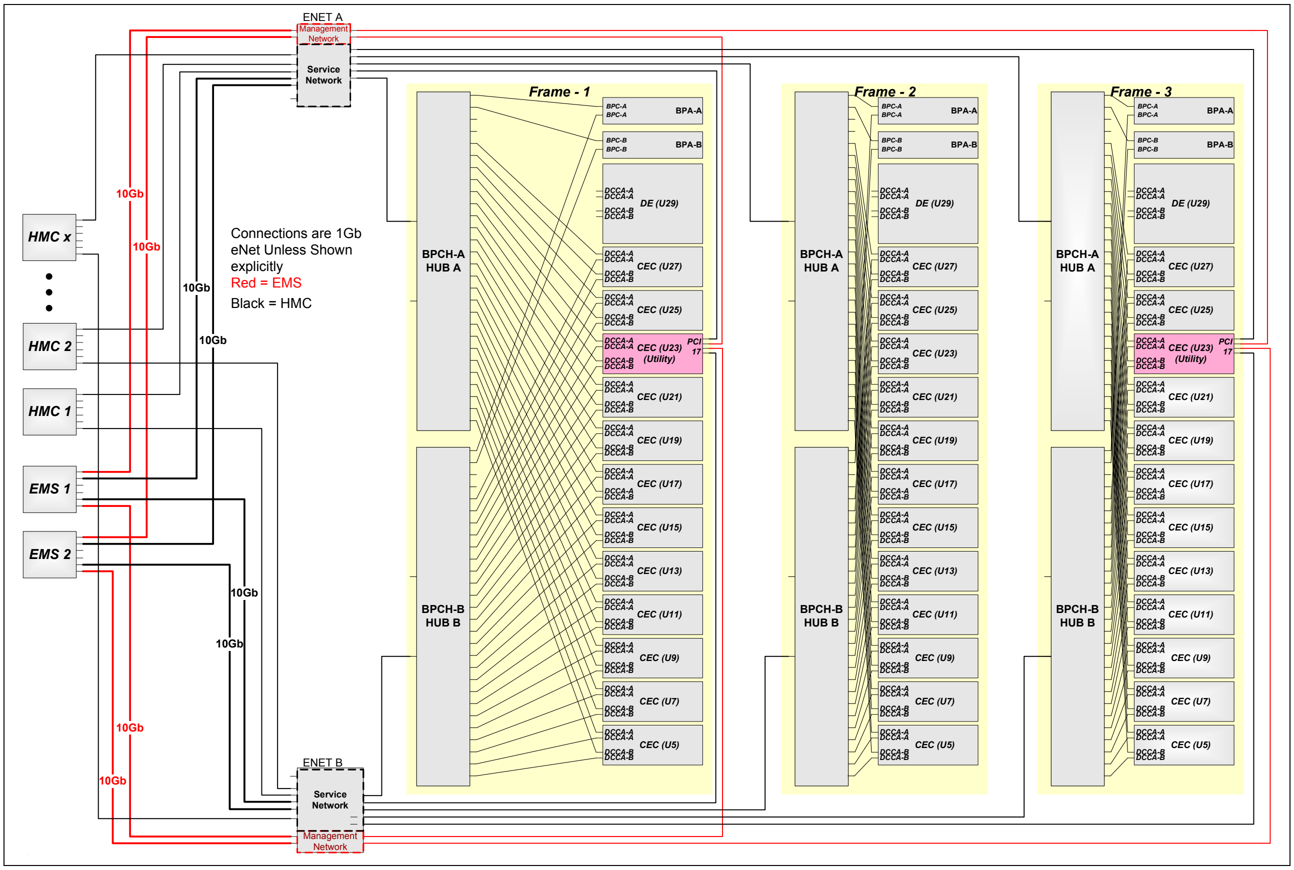


Figure 1-46 Logical structure of service and management networks

1.7.5 Data flow

This section provides a high-level description of the data flow on the cluster service network and cluster management operations.

After discovery of the hardware components, their definitions are stored in the xCAT database on the EMS. HMCs, CECs, and frames are discovered via Service Location Protocol (SLP) by xCAT. The discovery information includes model and serial numbers, IP addresses, and so on. The Ethernet switch of the service LAN also is queried to determine which switch port is connected to each component. This discovery is run again when the system is up, if wanted. The HFI/ISR cabling also is tested by the CNM daemon on the EMS. The disk enclosures and their disks are discovered by GPFS services on these dedicated nodes when they are booted up.

The hardware is configured and managed via the service LAN, which connects the EMS to the HMCs, BCAs, and FSPs.

Management is hierarchical with the EMS at the top, followed by the service nodes, then all the nodes in their building blocks. Management operations from the EMS to the nodes are also distributed out through the service nodes. Compute nodes are deployed by using a service node as the diskless image server.

Monitoring information comes from the sources (frames/CECs, nodes, HFI/ISR fabric, and so on), flows through the service LAN and cluster LAN back to the EMS, and is logged in the xCAT database.

1.7.6 LPARs

The minimum hardware requirement for an LPAR is one POWER7 chip with memory attached to its memory controller. If an LPAR is assigned to one POWER7, that chip must have memory on either of its memory controllers. If an LPAR is assigned to two, three, or four POWER7 chips, any one or more of the POWER7 chips must have memory that is attached to them.

A maximum of one LPAR per POWER7 chip supported. A single LPAR resides on one, two, three, or four POWER7 chips. This configuration results in an Octant with the capability to have one, two, three, or four LPARs. An LPAR cannot reside in two Octants. With this configuration, the number of LPARs per CEC (eight Octants) ranges 8 - 32 (4 x 8). Therefore, 1 - 4 LPARs per Octant and 8 - 32 LPARs per CEC.

The following LPAR assignments are supported in an Octant:

- LPAR with all processors and memory that is allocated to that LPAR
- LPARs with 75% of processor and memory resources that are allocated to the first LPAR and 25% to the second
- LPARs with 50% of processor and memory resources that are allocated to each LPAR
- LPARs with 50% of processor and memory resources that are allocated to the first LPAR and 25% to each of the remaining two LPARs
- LPARs with 25% of processor and memory resources that are allocated to each LPAR

Recall that for an LPAR to be assigned, a POWER7 chip and memory that is attached to its memory controller is required. If either one of the two requirements is not met, that POWER7 is skipped and the LPAR is assigned to the next valid POWER7 in the order.

1.7.7 Utility nodes

This section defines the utility node for all Power 775 frame configurations.

A CEC is defined as a Utility CEC (node) when it has the Management server (Service node) as an LPAR. Each frame configuration is addressed individually. A single Utility LPAR supports a maximum of 1536 LPARs, one of which is an LPAR (one Utility LPAR and 1535 other LPARs). Recall that a node contains four POWER7 chips and a single POWER7 contains a maximum of one LPAR; therefore, a CEC contains 8 x 4 = 32 POWER7 chips. This configuration results in up to 32 LPARs per CEC.

This result of 1536 LPARs translates to the following figures:

- 1536 POWER7 chips
- 384 Octants (1536 / 4 = 384)
- 48 CECs (a CEC can contain up to 32 LPARS; therefore, 1536 / 32 = 48)

There are always redundant utility nodes that reside in different frames when possible. If there is only one frame and multiple SuperNodes, the utility node resides in different SuperNodes. If there is only one SuperNode, the two utility nodes reside in different CECs. If there is only one CEC, the two utility LPARs reside in different Octants in the CEC.

The following defined utility CEC is used in the four-frame, three-frame, two-frame, and single-frame with 4, 8, and 12 CEC configurations. The single frame with 1 - 3 CECs uses a different utility CEC definition. These utilities CEC definitions are defined in their respective frame definition sections.

The utility LPAR resides in Octant 0. The LPAR is assigned only to a single POWER7. Figure 1-47 on page 59 shows the eight Octant CEC and the location of the Management LPAR. The two Octant and the four Octant CEC might be used as a utility CEC and follows the same rules as the eight Octant CEC.

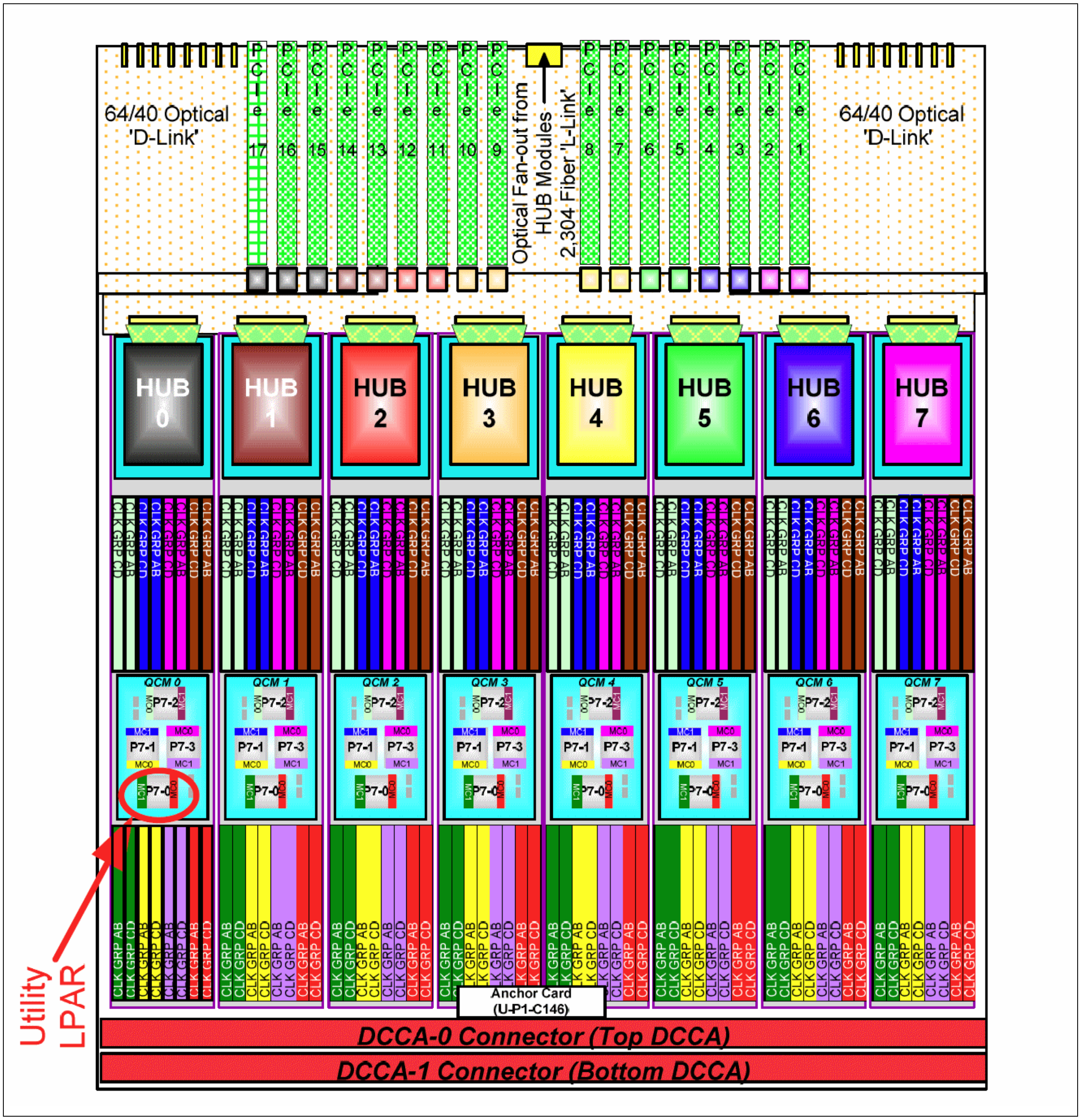


Figure 1-47 Eight octant utility node definition

1.7.8 GPFS I/O nodes

Figure 1-48 shows the GPFS Network Shared Disk (NSD) node in Octant 0.

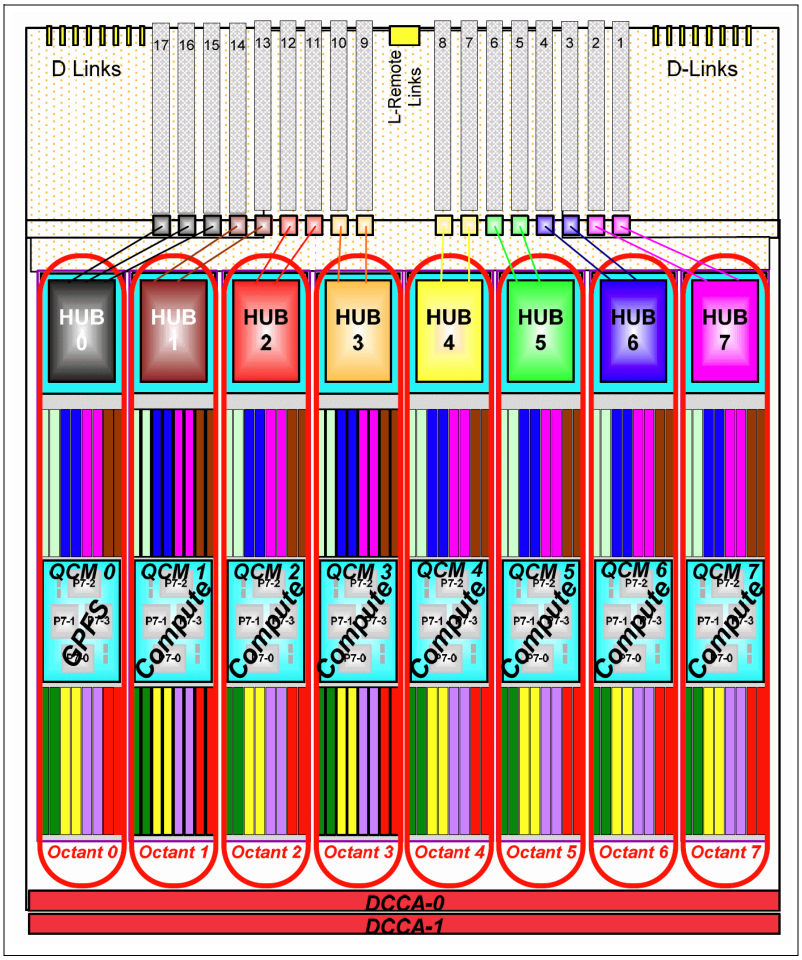


Figure 1-48 GPFS NSD node on octant 0

1.8 Connection scenario between EMS, HMC, and Frame

The network interconnect between the different system components (EMS server, HMC, Frame) requires the managing, running, maintaining, configuring, and monitoring of the cluster. The management rack for a POWER 775 Cluster houses the different components, such as the EMS servers (IBM POWER 750), HMCs, network switches, I/O drawers for the EMS data disks, keyboard, and mouse. The different networks that are used in such an environment are the management network and the service network (as shown in Figure 1-49 on page 61). The customer network is connected to some components, but for the actual cluster, only the management and service networks are essential. For more information about the server and management networks, see 1.7.4, “Server and management networks” on page 55.

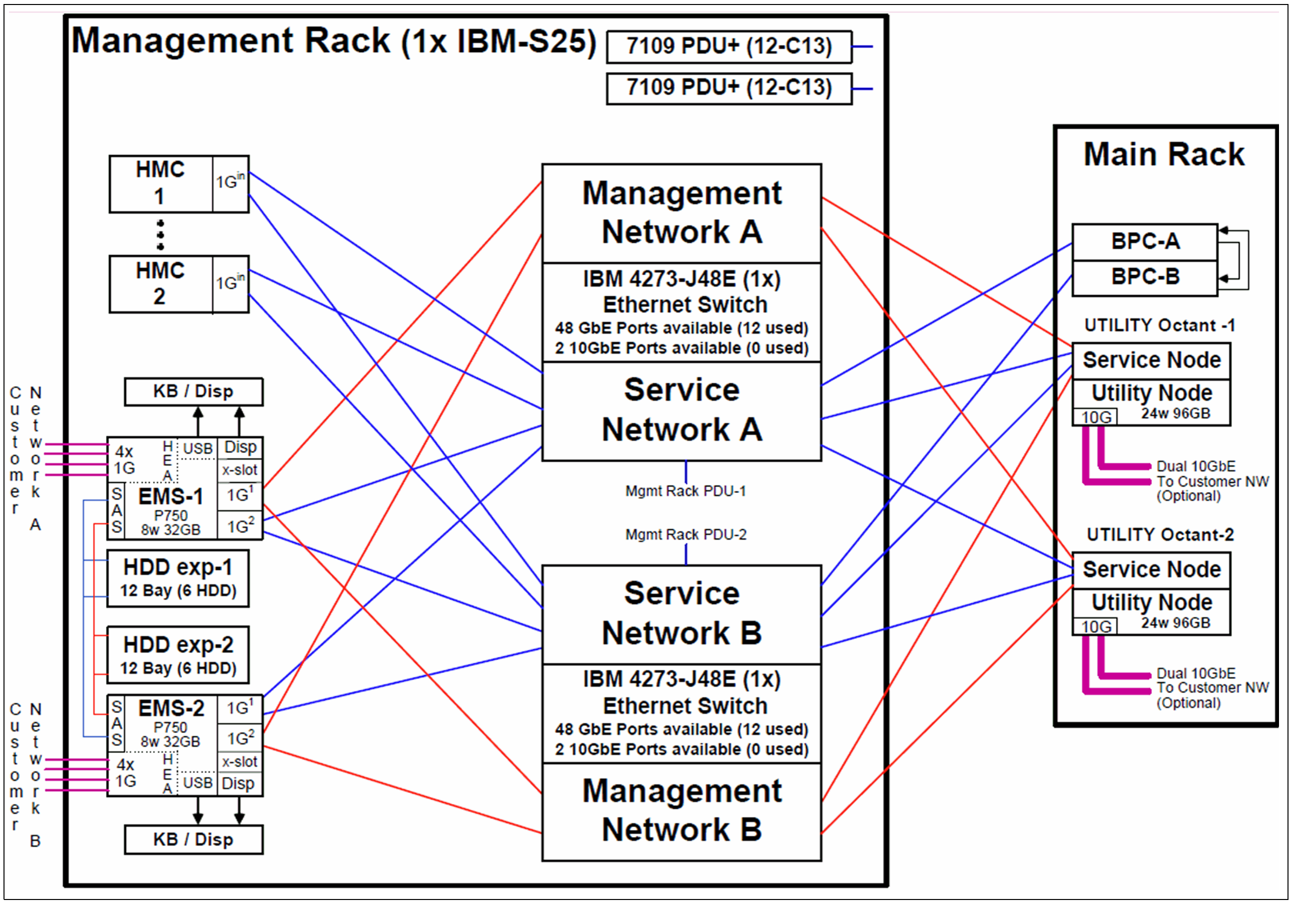


Figure 1-49 Typical cabling scenario for the HMC, the EMS, and the frame

In Figure 1-49, you see the different networks and cabling. Each Frame has two Ethernet ports on the BPCH to connect the Service Network A and B.

The I/O drawers in which the disks are installed for the EMS Servers also are interconnected. Therefore, the data is secured with RAID6 and the I/O drawers also are software mirrored. This means that when one EMS server goes down for any reason, the other EMS server accesses the data. The EMS servers are redundant from in this scenario, but there is no automated high-availability process for recovery of a failed EMS server.

All actions to activate the second EMS server must be performed manually. There also is no plan to automate this process. A cluster continues running without the EMS servers (in case both servers failed). No node fails because of a server failure or an HMC error. When multiple problems rise simultaneously, there might be a greater need for more intervention, but often this intervention does not occur under normal circumstances.

1.9 High Performance Computing software stack

Figure 1-50 shows the IBM Power Systems HPC software stack.

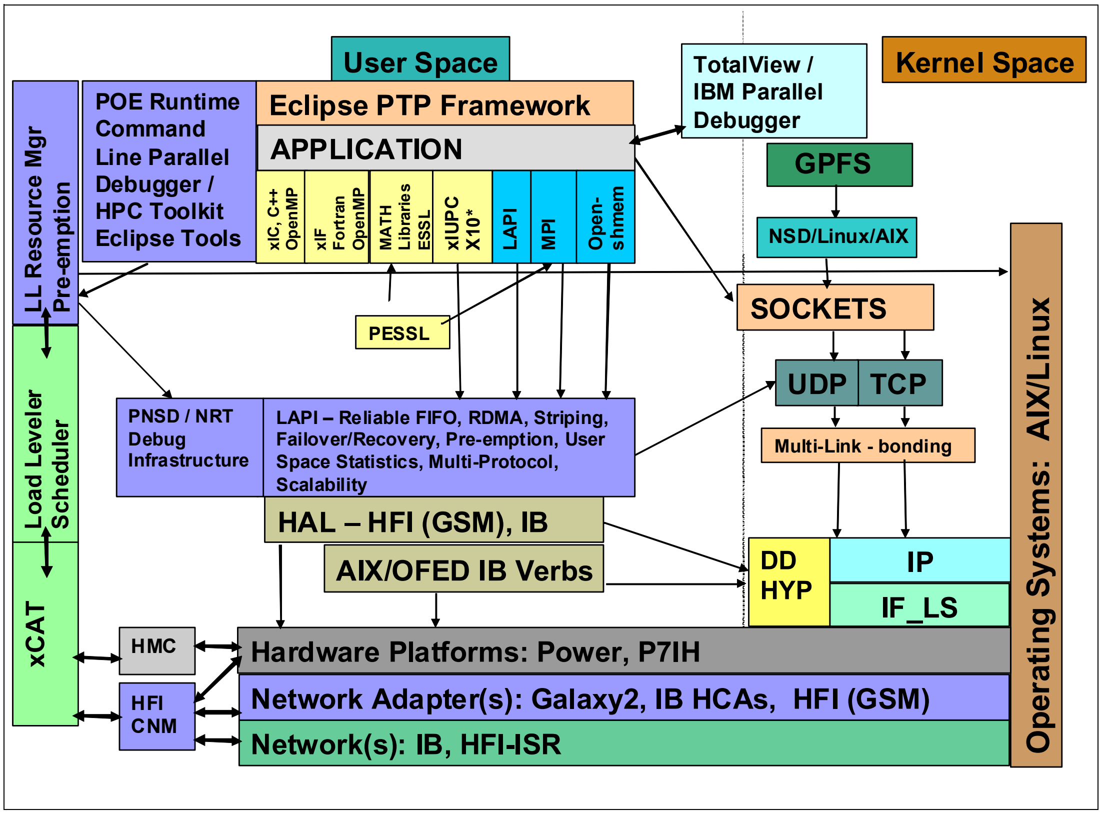

User Space
Kernel Space
LL Resource Mgr
Pre-emption
Load Leveler
Scheduler
xCAT
POE Runtime Command Line Parallel Debugger / HPC Toolkit Eclipse Tools
Eclipse PTP Framework
APPLICATION
xlC, C++ OpenMP
xlF Fortran OpenMP
MATH Libraries ESSL
xlUPC X10*
LAPI
MPI
Open-shmem
PESSL
TotalView / IBM Parallel Debugger
GPFS
NSD/Linux/AIX
SOCKETS
UDP
TCP
Multi-Link - bonding
PNSD / NRT Debug Infrastructure
LAPI – Reliable FIFO, RDMA, Striping, Failover/Recovery, Pre-emption, User Space Statistics, Multi-Protocol, Scalability
HAL – HFI (GSM), IB
AIX/OFED IB Verbs
DD HYP
IP
IF_LS
HMC
HFI CNM
Hardware Platforms: Power, P7IH
Network Adapter(s): Galaxy2, IB HCAs, HFI (GSM)
Network(s): IB, HFI-ISR
Operating Systems: AIX/Linux

Figure 1-50 POWER HPC software stack

Table 1-6 describes the IBM Power HPC software stack.

Table 1-6 HPC software stack for POWER

Application Development Environment	Available tools for IBM POWER	Resources
HPC Workbench Integrated Development Environment that is based on Eclipse PTP (open source)	C and Fortran Development Tools	http://www.eclipse.org/photran/
	PTP (Parallel tools platform) Programming models support: MPI, LAPI, OpenShmem, UPC	http://www.eclipse.org/ptp/
High Scalable Communications Protocol	IBM Parallel Environment (MP, LAPI/PAMI, Debug Tools, OpenShmem) Note: User space support IB, HFI	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pe.doc/pebooks.html
Performance Tuning Tools	IBM HPC Toolkit (part of PE)	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pe.doc/pebooks.html

Application Development Environment	**Available tools for IBM POWER**	**Resources**
PGAS language support	Unified Parallel C (UPC)	http://upc.lbl.gov/
	X10	http://x10-lang.org/
	OpenMP	http://openmp.org/
Compilers	XL C/C++	http://www.ibm.com/software/awdtools/xlcpp/
	XL Fortran	http://www.ibm.com/software/awdtools/fortran/
Performance Counter	PAPI	http://icl.cs.utk.edu/papi/
Debugger	Allinea Parallel Debugger	http://www.allinea.com/products/ddt
	Totalview	http://www.roguewave.com/products/totalview-family.aspx
	Eclipse debugger	http://eclipse.org/ptp/
	PERCS debugger	
GPU support	OpenCL	http://www.khronos.org/opencl/
Scientific Math libraries	ESSL	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.essl.doc/esslbooks.html
	Parallel ESSL	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pessl.doc/pesslbooks.html
	HPCS Toolkit	http://domino.research.ibm.com/comm/research_projects.nsf/pages/hpcst.index.html
Advanced Systems Management		
Development, maintenance	xCAT	http://xcat.sourceforge.net/
Remote hardware controls		
System monitoring	RSCT (RMC)	http://www.redbooks.ibm.com/abstracts/sg246615.html http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.related_libraries.doc/related.htm?path=3_6#rsct_link
Event handling	Toolkit for Event Analysis and Logging (TEAL)	http://pyteal.sourceforge.net
Workload and Resource Management		
Scheduler	IBM LoadLeveler	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.loadl.doc/llbooks.html
Integrated resource manager		
Cluster File System		

Application Development Environment	**Available tools for IBM POWER**	**Resources**
Advanced scalable file system	GPFS	http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html
Network	NFS	
Operating System Support		
Base support	AIX 7.1B	http://www.ibm.com/systems/power/software/aix/index.html
	RHEL 6	http://www.ibm.com/systems/power/software/linux/
Key OS enhancements for HPC	AIX	http://www.ibm.com/systems/power/software/aix/index.html
	Linux	http://www.ibm.com/systems/power/software/linux/
Network Management		
InfiniBand	Vendor supported tools	http://www.infinibandta.org/
HFI	Switch management (CNM)	
	Route management	
	Failover/recovery	
	Performance counter collection and analysis	
Firmware level	GFW	
	LNMC	
Cluster Database		
Database	DB2®	http://www.ibm.com/software/data/db2/
Other Key Features		
Scalability	16K OS images (special bid)	
OS Jitter	Best practices guide	
	Jitter migration bases on synchronized global clock	
	Kernel patches	
RAS	Failover	
	Striping with multiple links	
Multilink/bonding support	Supported	

1.9.1 Integrated Switch Network Manager

The ISNM subsystem package is installed on the executive management server of a high-performance computing cluster that consists of IBM Power 775 Supercomputers and contains the network management commands. The local network management controller runs on the server service processor as part of the system of the drawers and is shipped with the Power 775.

Network management services

As shown in Figure 1-51 on page 66, the ISNM provides the following services:

- ISR network configuration and installation:
  - Topology validation
  - Miswire detection
  - Works with cluster configuration as defined in the cluster database
  - Hardware global counter configuration
  - Phased installation and Optical Link Connectivity Test (OLCT)
- ISR network hardware status:
  - Monitors for ISR, HFI, link, and optical module events
  - Command line queries to display network hardware status
  - Performance counter collection
  - Some RMC monitor points (for example, HFI Down)
- Network maintenance:
  - Set up ISR route tables during drawer power-on
  - Thresholds on certain link events, might disable a link
  - Dynamically update route tables to reroute around problems or add to routes when CECs power on
  - Maintain data to support software route mode choices, makes the data available to the OS through PHYP
  - Monitor global counter health
- Report hardware failures:
  - Analyzes the EMS
  - Most events that are forwarded to TEAL Event DB and Alert DB
  - Link events due to CEC power off/power on are consolidated within CNM to reduce unnecessary strain on analysis
  - Events reported via TEAL to Service Focal Point™ on the HMC

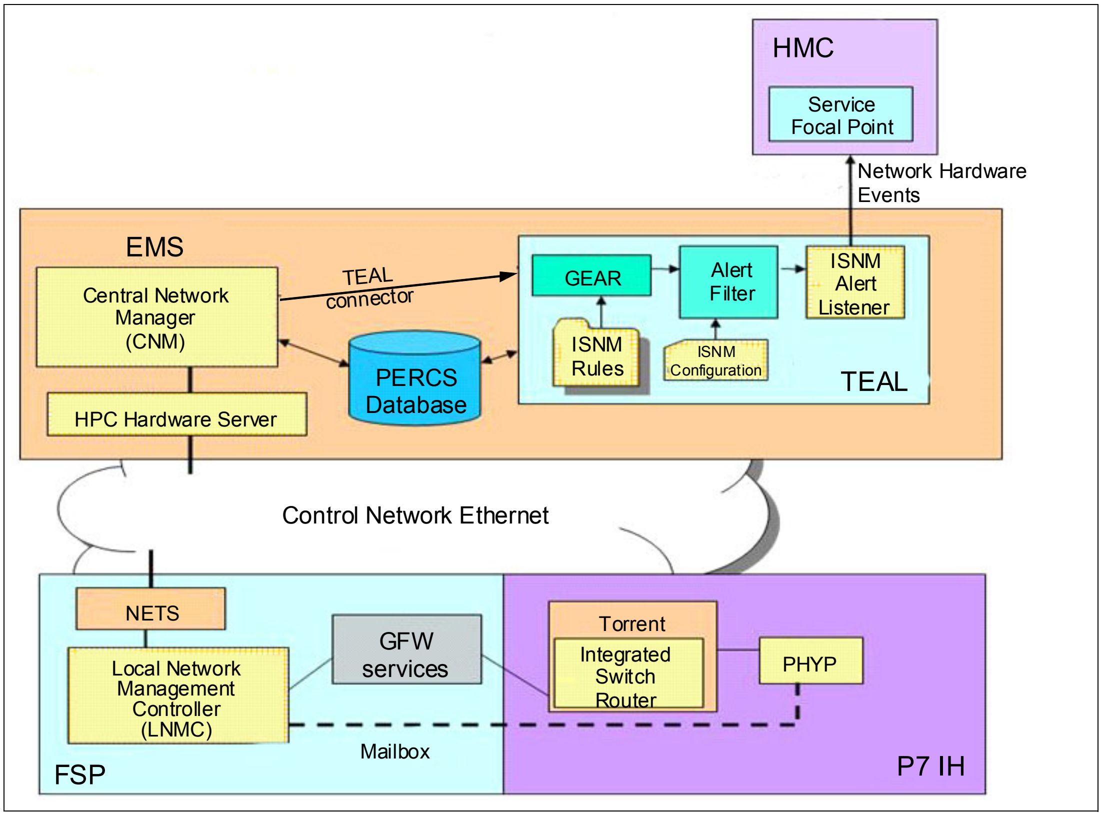


Figure 1-51 ISNM operating environment

MCRSA for the ISNM: IBM offers Machine Control Program Remote Support Agreement (MCRSA) for the ISNM. This agreement includes remote call-in support for the central network manager and the hardware server components of the ISNM, and for the local network management controller machine code.

MCRSA enables a single-site or worldwide enterprise customer to maintain machine code entitlement to remote call-in support for ISNM throughout the life of the MCRSA.

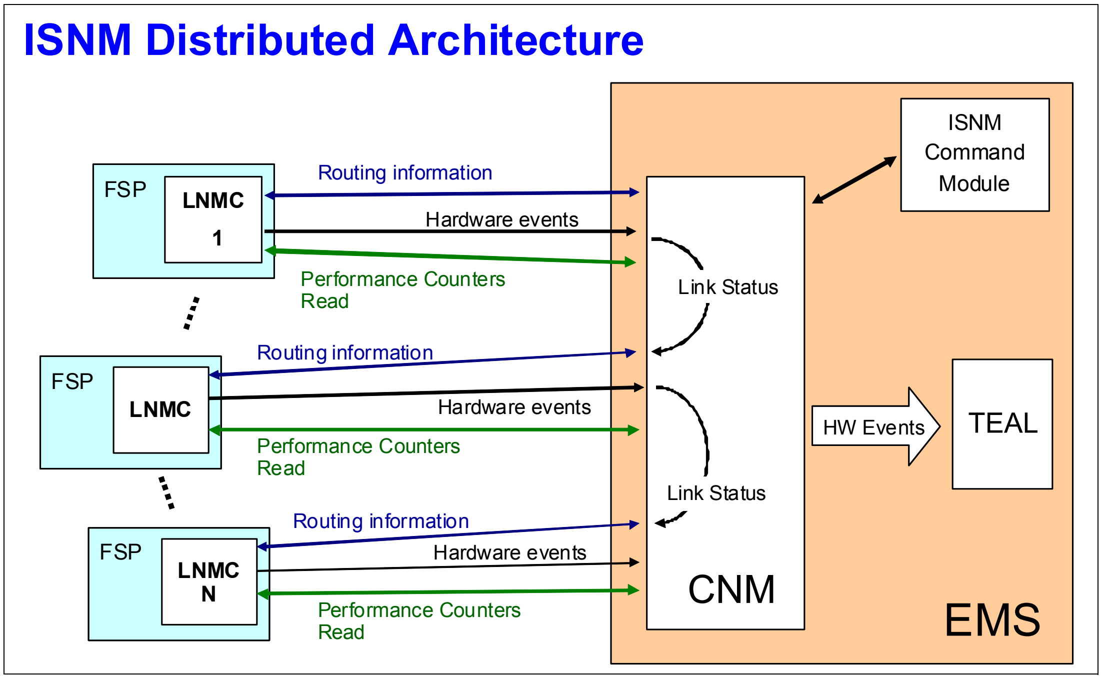


Figure 1-52 ISNM distributed architecture

A high-level representation of ISNMs distributed architecture is shown in Figure 1-52. An instance of Local Network Manager (LNMC) software runs on each FSP. Each LNMC generates routes for the eight hubs in the local drawer specific to the supernode, drawer, and hub.

A Central Network Manager (CNM) runs on the EMS and communicates with the LNMCs. Link status and reachability information flows between the LNMC instances and CNM. Network events flow from LNMC to CNM, and then to Toolkit for Event Analysis and Logging (TEAL).

Local Network Management Controller

The LNMC present on each node features the following primary functions:

- Event management:
  - Aggregates local hardware events, local routing events, and remote routing events.
- Route management:
  - Generates routes that are based on configuration data and the current state of links in the network.
- Hardware access:
  - Downloads routes.
  - Allows the hardware to be examined and manipulated.

Figure 1-53 on page 68 shows a logical representation of these functions.

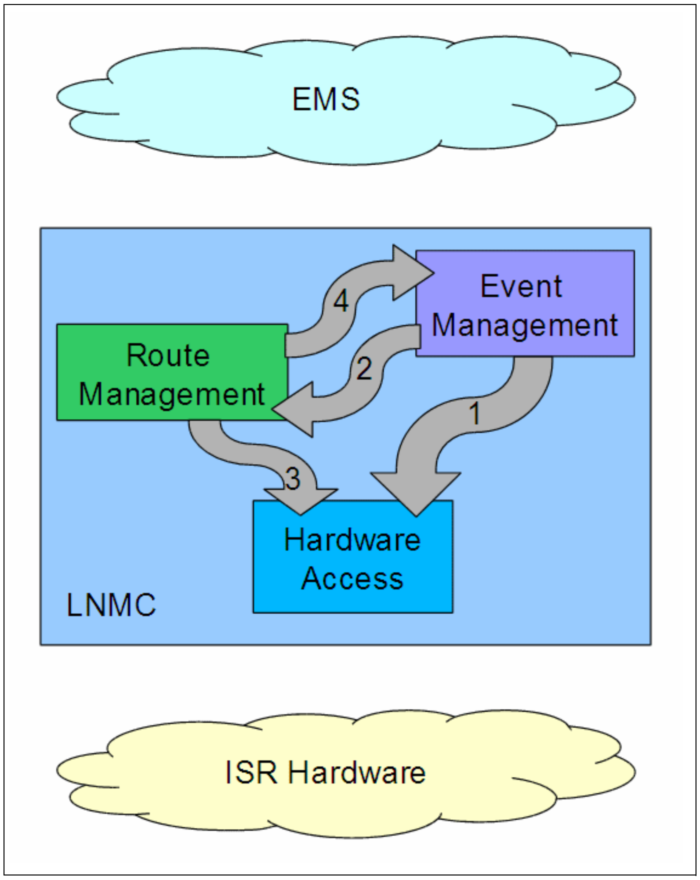


Figure 1-53 LNMC functional blocks

The LNMC also interacts with the EMS and with the ISR hardware to support the execution of vital management functions. Figure 1-54 on page 69 provides a high-level visualization of the interaction between the LNMC components and other external entities.

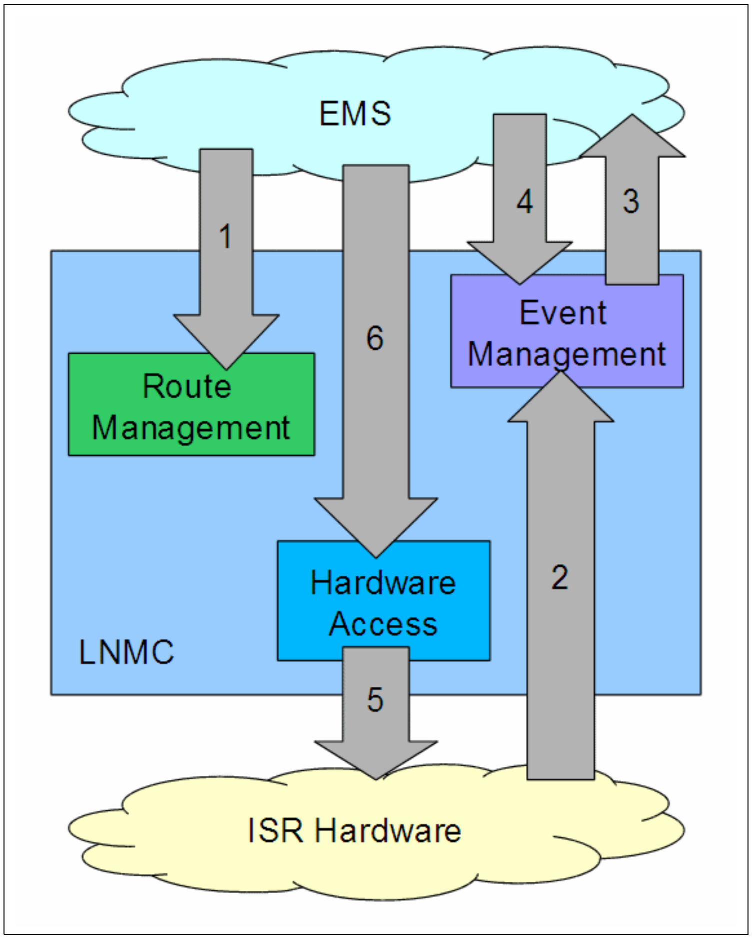


Figure 1-54 LNMC external interactions

As shown in Figure 1-54, the following external interactions are featured in the LNMC:

1. Network configuration commands

   The primary function of this procedure is to uniquely identify the Power 775 server within the network. This includes the following information:

   - Network topology
   - Supernode identification
   - Drawer identification within the Supernode
   - Frame identification (from BPA via FSP)
   - Cage identification (from BPA via FSP)
   - Expected neighbors table for mis-wire detection

2. Local network hardware events

   All network hardware events flow from the ISR into the LNMCs event management, where they are examined and acted upon. The following list of potential actions that are taken by event management:

   - Threshold checking
   - Actions upon hardware
   - Event aggregation
   - Network status update. Involves route management and CNM reporting.
   - Reporting to EMS

3. Network event reporting

   Event management examines each local network hardware event and, if appropriate, forwards the event to the EMS for analysis and reports to the service focal point. Event management also sends the following local routing events that indicate changes in the link status or route tables within the local drawer that other LNMCs need to react to:

   – Link usability masks (LUM): One per hub in the drawer, indicates whether each link on that hub is available for routing

   – PRT1 and PRT2 validity vectors: One each per hub in the drawer, more data is used in making routing decisions

   General changes in LNMC or network status are also reported via this interface.

4. Remote network events

   After a local routing event (LUM, PRT1, PRT2) is received by CNM, CNM determines which other LNMCs need the information to make route table updates, and sends the updates to the LNMCs.

   The events are aggregated together by event management and then passed to route management. Route management generates a set of appropriate route table updates and potentially some PRT1 and PRT2 events of its own.

   Changed routes are downloaded via hardware access. Event management sends out new PRT1 and PRT2 events, if applicable.

5. Local hardware management

   Hardware access provides the following facilities to both LNMC and CNM for managing the network hardware:

   – Reads and writes route tables
   – Reads and writes hardware registers
   – Disables and enables ports
   – Controls optical link connectivity test
   – Allows management of multicast
   – Allows management of global counter
   – Reads and writes performance counters

6. Centralized hardware management

   The following functions are managed centrally by CNM with support from LNMC:

   – Global counter
   – Multicast
   – Port Enable/Disable

Central Network Manage

The CNM daemon waits for events and handles each one as separate transactions. There are software threads within CNM that handle different aspects of the network management tasks.

The service network traffic flows through another daemon called *High Performance Computing Hardware Server*.

Figure 1-55 on page 71 shows the relationships between the CNM software components. The components are described in the following section.

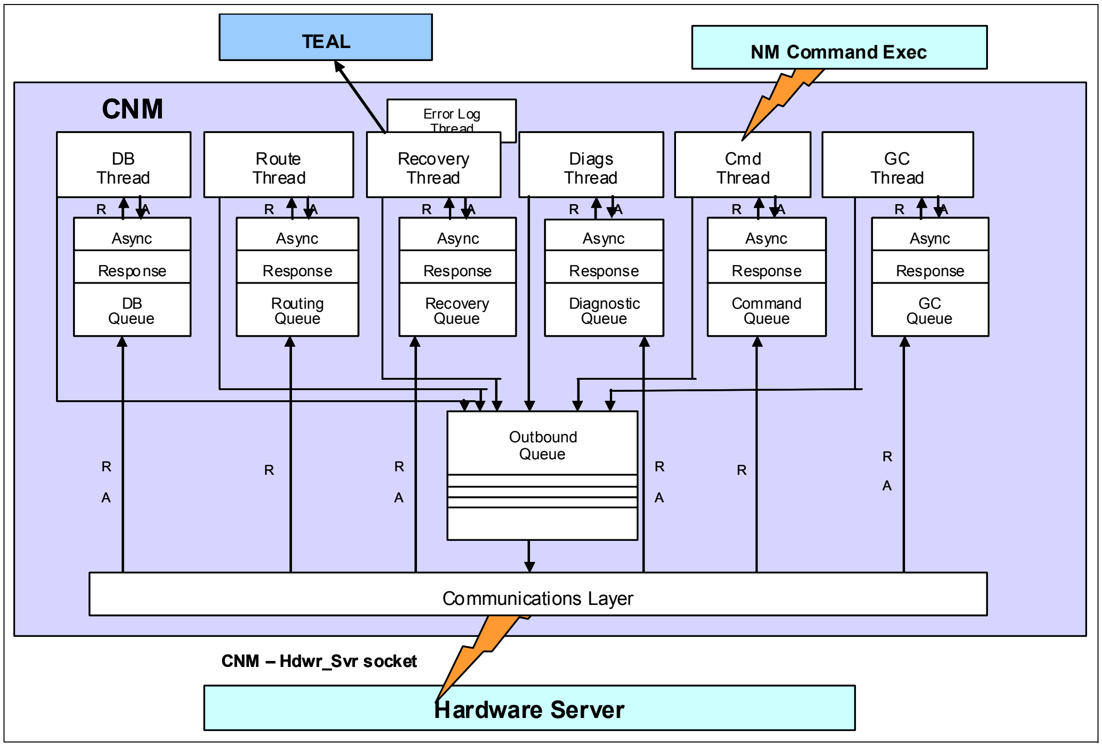


Figure 1-55 CNM software structure

Communication layer

This layer provides a packet library with methods for communicating with LNMC. The layer manages incoming and outgoing messages between FSPs and CNM component message queues.

The layer also manages event aggregation and the virtual connections to the Hardware Server.

Database component

This component maintains the CNM internal network hardware database and updates the status fields in this in-memory database to support reporting the status to the administrator. The component also maintains required reachability information for routing.

Routing component

This component builds and maintains the hardware multicast tree. The component also writes multicast table contents to the ISR and handles the exchange of routing information between the LNMCs to support route generation and maintenance.

Global counter component

This component sets up and monitors the hardware global counter. The component also maintains information about the location of the ISR master counter and configured backups.

Recovery component

The recovery component gathers network hardware events and frame-level events. This component also logs each event in the CNM_ERRLOG and sends most events to TEAL.

The recovery component also performs some event consolidation to avoid flooding the TEAL with too many messages in the event of a CEC power up or power down.

Performance counter data management

This data management periodically collects ISR and HFI aggregate performance counters from the hardware and stores the counters in the cluster database. The collection interval and amount of data to keep are configurable.

Command handler

This handler is a socket listener to the command module. This handler manages ISNM commands, such as requests for hardware status, configuration for LNMC, and link diagnostics.

IBM High Performance Computing Hardware Server

In addition to the CNM software components, the HPC Hardware Server (HWS) handles the connections to the service network. Its primary function is to manage connections to service processors and provide an API for clients to communicate with the service processors. HWS assigns every service processor connection a unique handle that is called a *virtual port number* (vport). This handle is used by clients to send synchronous commands to the hardware.

In a Power 775 cluster, HPC HWS runs on the EMS, and on each xCAT service node.

1.9.2 DB2

IBM DB2 Workgroup Server Edition 9.7 for High Performance Computing (HPC) V1.1 is a scalable, relational database that is designed for use in a local area network (LAN) environment and provides support for both local and remote DB2 clients. DB2 Workgroup Server Edition is a multi-user version of DB2 packed with features that are designed to reduce the overall costs of owning a database. DB2 includes data warehouse capabilities, high availability function, and is administered remotely from a satellite control database.

The IBM Power 775 Supercomputer cluster solution requires a database to store all of the configuration and monitoring data. DB2 Workgroup Server Edition 9.7 for HPC V1.1 is licensed for use only on the executive management server (EMS) of the Power 775 high-performance computing cluster.

The EMS serves as a single point of control for cluster management of the Power 775 cluster. The Power 775 cluster also includes a backup EMS, service nodes, compute nodes, I/O nodes, and login nodes. DB2 Workgroup Server Edition 9.7 for HPC V1.1 must be installed on the EMS and backup EMS.

1.9.3 Extreme Cluster Administration Toolkit

Extreme Cloud Administration Toolkit (xCAT) is an open source, scalable distributed computing management, and provisioning tool that provides a unified interface for hardware control, discovery, and operating system stateful and stateless deployment. This robust toolkit is used for the deployment and administration of AIX or Linux clusters, as shown in Figure 1-56 on page 73.

xCAT makes simple clusters easy and complex clusters possible through the following features:

- Remotely controlling hardware functions, such as power, vitals, inventory, events logs, and alert processing. xCAT indicates which light path LEDs are lit up remotely.
- Managing server consoles remotely via serial console, SOL.
- Installing an AIX or Linux cluster with utilities for installing many machines in parallel.
- Managing an AIX or Linux cluster with tools for management and parallel operation.
- Setting up a high-performance computing software stack, including software for batch job submission, parallel libraries, and other software that is useful on a cluster.
- Creating and managing stateless and diskless clusters.

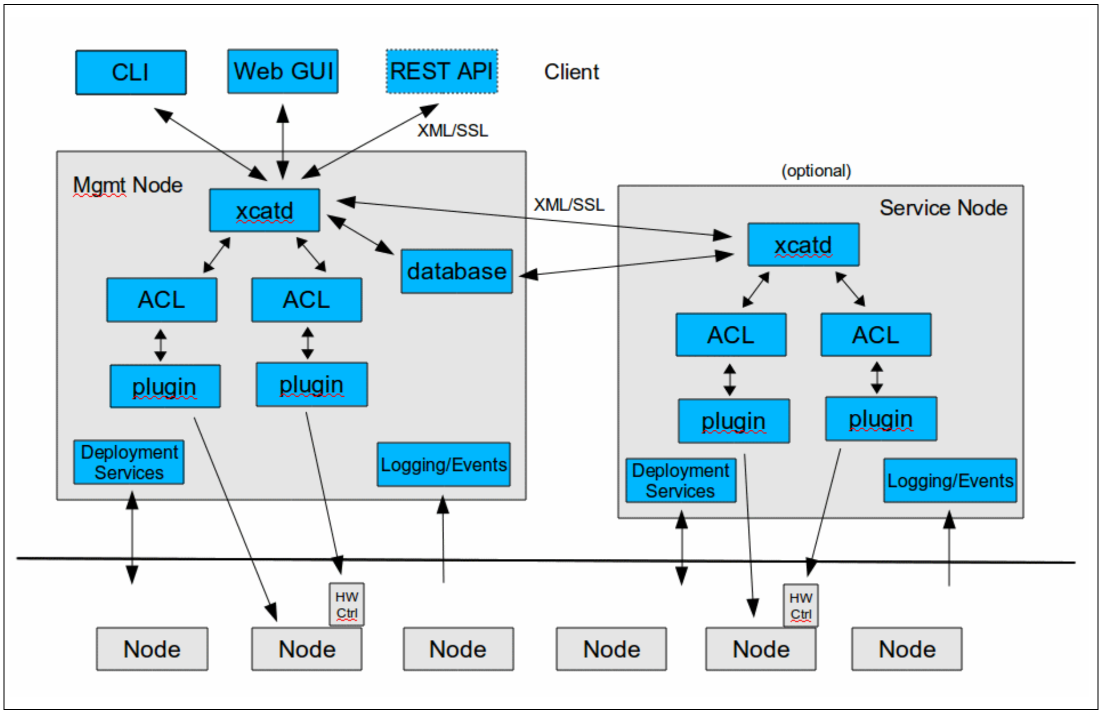


Figure 1-56 xCAT architecture

xCAT supports both Intel and POWER based architectures, which provide operating system support for AIX, Linux (RedHat, SuSE and CentOS), and Windows installations. the following provisioning methods are available:

- Local disk
- Stateless (via Linux ramdisk support)
- iSCSI (Windows and Linux)

xCAT manages a Power 775 cluster by using a hierarchical distribution that is based on management and service nodes. A single xCAT management node with multiple service nodes provides boot services to increase scaling (to thousands and up to tens of thousands of nodes).

The number of nodes and network infrastructure determine the number of Dynamic Host Configuration Protocol/Trivial File Transfer Protocol/Hypertext Transfer Protocol (DHCP/TFTP/HTTP) servers that are required for a parallel reboot without DHCP/TFTP/HTTP timeouts.

The number of DHCP servers does not need to equal the number of TFTP or HTTP servers. TFTP servers NFS mount read-only the /tftpboot and image directories from the management node to provide a consistent set of kernel, initrd, and file system images.

xCAT version 2 provides the following enhancements that address the requirements of a Power 775 cluster:

- Improved ACLs and non-root operator support:
  - Certificate-authenticated client/server XML protocol for all xCAT commands
- Choice of databases:
  - Use a database (DB) like SQLite, or an enterprise DB like DB2 or Oracle
  - Stores all of the cluster config data, status information, and events
  - Information is stored in DB by other applications and customer scripts
  - Data change notification is used to drive automatic administrative operations
- Improved monitoring:
  - Hardware event and simple Network Management Protocol (SNMP) alert monitoring
  - More HPC stack (GPFS, LL, Torque, and so on) setup and monitoring
- Improved RMC conditions:
  - Condition triggers when it is true for a specified duration
  - Batch multiple events into a single invocation of the response
  - Micro-sensors: ability to extend RMC monitoring efficiently
  - Performance monitoring and aggregation that is based on TEAL and RMC
- Automating the deployment process:
  - Automate creation of LPARs in every CEC
  - Automate set up of infrastructure nodes (service nodes and I/O nodes)
  - Automate configuration of network adaptors, assign node names/IDs, IP addresses, and so on
  - Automate choosing and pushing the corresponding operating system and other HPC software images to nodes
  - Automate configuration of the operating system and HPC software so that the system is ready to use
  - Automate verification of the nodes to ensure their availability
- Boot nodes with a single shared image among all nodes of a similar configuration (diskless support)
- Allow for deploying the cluster in phases (for example, a set of new nodes at-a-time by using the existing cluster)
- Scan the connected networks to discover the various hardware components and firmware information of interest:
  - Uses the standard SLP protocol
  - Finds: FSPs, BPAs, hardware control points
- Automatically defines the discovered components to the administration software, assigning IP addresses, and hostnames
- Hardware control (for example, powering components on and off) is automatically configured
- ISR and HFI components are initialized and configured
- All components are scanned to ensure that firmware levels are consistent and at the wanted version

- Firmware is updated on all down-level components when necessary
- Provide software inventory:
  - Utilities to query the software levels that are installed in the cluster
  - Utilities to choose updates to be applied to the cluster
- With diskless nodes, software updates are applied to the OS image on the server (nodes apply the updates on the next reboot)
- HPC software (LoadLeveler, GPFS, PE, ESSL, Parallel ESSL, compiler libraries, and so on) is installed throughout the cluster by the system management software
- HPC software relies on system management to provide configuration information. System Management stores the configuration information in the management database
- Uses RMC monitoring infrastructure for monitoring and diagnosing the components of interest
- Continuous operation (rolling update):
  - Apply upgrades and maintenance to the cluster with minimal impact on running jobs
  - Rolling updates are coordinated with CNM and LL to schedule updates (reboots) to a limited set of nodes at a time, allowing the other nodes to still be running jobs

1.9.4 Toolkit for Event Analysis and Logging

The Toolkit for Event Analysis and Logging (TEAL) is a robust framework for low-level system event analysis and reporting that supports both real-time and historic analysis of events. TEAL provides a central repository for low-level event logging and analysis that addresses the new Power 775 requirements.

The analysis of system events is delivered through alerts. A rules-based engine is used to determine which alert must be delivered. The TEAL configuration controls the manner in which problem notifications are delivered.

Real-time analysis provides a pro-active approach to system management, and the historical analysis allows for deeper on-site and off-site debugging.

The primary users of TEAL are the system administrator and operator. The output of TEAL is delivered to an alert database that is monitored by the administrator and operators through a series of monitoring methods.

TEAL runs on the EMS and commands are issued via the EMS command line. TEAL supports the monitoring of the following functions:

- ISNM/CNM
- LoadLeveler
- HMCs/Service Focal Points
- PNSD
- GPFS

For more information about TEAL, see Table 1-6 on page 62.

1.9.5 Reliable Scalable Cluster Technology

Reliable Scalable Cluster Technology (RSCT) is a set of software components that provide a comprehensive clustering environment for AIX, Linux, Solaris, and Windows. RSCT is the infrastructure that is used by various of IBM products to provide clusters with improved system availability, scalability, and ease of use.

RSCT includes the following components:

- Resource monitoring and control (RMC) subsystem

  This subsystem is the scalable, reliable backbone of RSCT. RMC 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 use RMC for single system monitoring or for monitoring nodes in a cluster. However, 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

  This RSCT component provides the security infrastructure that enables RSCT components to authenticate the identity of other parties.

- Topology services subsystem

  This RSCT component provides node and network failure detection on some cluster configurations.

- Group services subsystem

  This RSCT component provides cross-node/process coordination on some cluster configurations.

For more information, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.related_libraries.doc/related.htm?path=3_6#rsct_link

1.9.6 GPFS

The IBM General Parallel File System (GPFS) is distributed, high-performance, massively scalable enterprise file system solution that addresses the most challenging demands in high-performance computing.

GPFS provides online storage management, scalable access, and integrated information lifecycle management tools capable of managing petabytes of data and billions of files. Virtualizing your file storage space and allowing multiple systems and applications to share common pools of storage provides you the flexibility to transparently administer the infrastructure without disrupting applications. This configuration improves cost and energy efficiency and reduces management overhead.

Massive namespace support, seamless capacity and performance scaling, and proven reliability features and flexible architecture of GPFS helps your company foster innovation by simplifying your environment and streamlining data work flows for increased efficiency.

GPFS plays a key role in the shared storage configuration for Power 775 clusters. Virtually all large-scale systems are connected to disk over HFI via GPFS Network Shared Disk (NSD) servers, which are referred GPFS I/O nodes or Storage nodes in Power 775 terminology. The system interconnect features higher performance and is far more scalable than traditional storage fabrics, and is RDMA capable.

GPFS includes a Native RAID function that is used to manage the disks in the disk enclosures. In particular, the disk hospital function is queried regularly to ascertain the health of the disk subsystem. This function is not always necessary because disk problems that require service are reported to the HMC serviceable events and to TEAL.

For more information about GPFS, see Table 1-6 on page 62.

GPFS Native RAID

GPFS Native RAID is a software implementation of storage RAID technologies within GPFS. By using conventional dual-ported disks in a Just-a-Bunch-Of-Disks (JBOD) configuration, GPFS Native RAID implements sophisticated data placement and error correction algorithms to deliver high levels of storage reliability, availability, and performance. Standard GPFS file systems are created from the NSDs defined through GPFS Native RAID.

This section describes the basic concepts, advantages, and motivations behind GPFS Native RAID: redundancy codes, end-to-end checksums, data declustering, and administrator configuration, including recovery groups, declustered arrays, virtual disks, and virtual disk NSDs.

Overview

GPFS Native RAID integrates the functionality of an advanced storage controller into the GPFS NSD server. Unlike an external storage controller, in which configuration, LUN definition, and maintenance are beyond the control of GPFS, GPFS Native RAID takes ownership of a JBOD array to directly match LUN definition, caching, and disk behavior to GPFS file system requirements.

Sophisticated data placement and error correction algorithms deliver high levels of storage reliability, availability, serviceability, and performance. GPFS Native RAID provides a variation of the GPFS NSD called a *virtual disk*, or VDisk. Standard NSD clients transparently access the VDisk NSDs of a file system by using the conventional NSD protocol.

The GPFS Native RAID includes the following features:

- Software RAID: GPFS Native RAID runs on standard AIX disks in a dual-ported JBOD array, which does not require external RAID storage controllers or other custom hardware RAID acceleration.
- Declustering: GPFS Native RAID distributes client data, redundancy information, and spare space uniformly across all disks of a JBOD. This distribution reduces the rebuild (disk failure recovery process) overhead that is compared to conventional RAID.
- Checksum: An end-to-end data integrity check (by using checksums and version numbers) is maintained between the disk surface and NSD clients. The checksum algorithm uses version numbers to detect silent data corruption and lost disk writes.
- Data redundancy: GPFS Native RAID supports highly reliable two-fault tolerant and three-fault-tolerant Reed-Solomon-based parity codes and three-way and four-way replication.
- Large cache: A large cache improves read and write performance, particularly for small I/O operations.

- Arbitrarily sized disk arrays: The number of disks is not restricted to a multiple of the RAID redundancy code width, which allows flexibility in the number of disks in the RAID array.
- Multiple redundancy schemes: One disk array supports VDisks with different redundancy schemes; for example, Reed-Solomon and replication codes.
- Disk hospital: A disk hospital asynchronously diagnoses faulty disks and paths, and requests replacement of disks by using past health records.
- Automatic recovery: Seamlessly and automatically recovers from primary server failure.
- Disk scrubbing: A disk scrubber automatically detects and repairs latent sector errors in the background.
- Familiar interface: Standard GPFS command syntax is used for all configuration commands, including, maintaining, and replacing failed disks.
- Flexible hardware configuration: Support of JBOD enclosures with multiple disks physically mounted together on removable carriers.
- Configuration and data logging: Internal configuration and small-write data are automatically logged to solid-state disks for improved performance.

GPFS Native RAID features

This section describes three key features of GPFS Native RAID and how the functions work: data redundancy that use RAID codes, end-to-end checksums, and declustering.

RAID codes

GPFS Native RAID automatically corrects for disk failures and other storage faults by reconstructing the unreadable data by using the available data redundancy of either a Reed-Solomon code or N-way replication. GPFS Native RAID uses the reconstructed data to fulfill client operations, and in the case of disk failure, to rebuild the data onto spare space. GPFS Native RAID supports two- and three-fault tolerant Reed-Solomon codes and three-way and four-way replication, which detect and correct up to two or three concurrent faults1. The redundancy code layouts that are supported by GPFS Native RAID, called *tracks*, are shown in Figure 1-57.

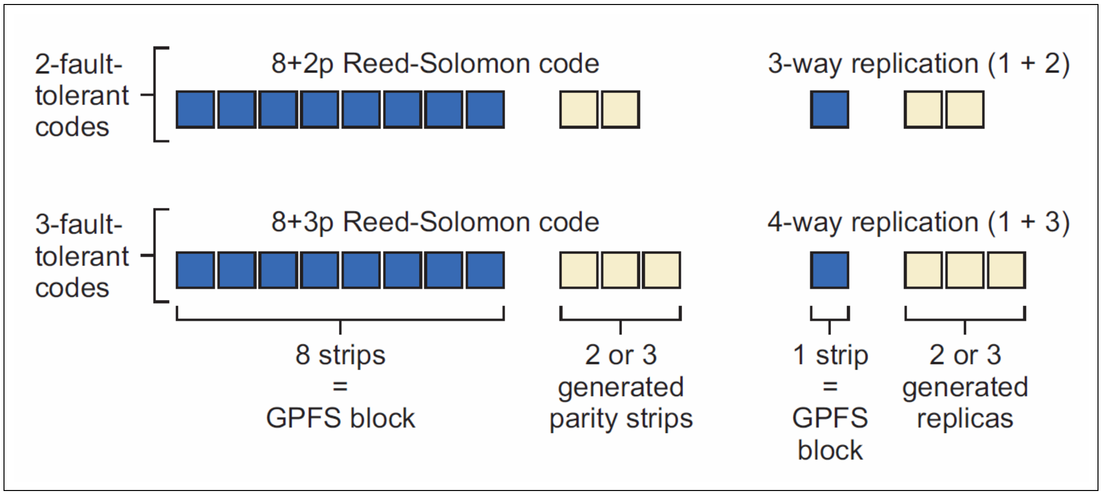


Figure 1-57 Redundancy codes that are supported by GPFS Native RAID

GPFS Native RAID supports two- and three-fault tolerant Reed-Solomon codes, which partition a GPFS block into eight data strips and two or three parity strips. The N-way replication codes duplicate the GPFS block on N - 1 replica strips.

GPFS Native RAID automatically creates redundancy information, depending on the configured RAID code. By using a Reed-Solomon code, GPFS Native RAID equally divides a GPFS block of user data into eight data strips and generates two or three redundant parity strips. This configuration results in a stripe or track width of 10 or 11 strips and storage efficiency of 80% or 73% (excluding user configurable spare space for rebuild).

By using N-way replication, a GPFS data block is replicated N - 1 times, implementing 1 + 2 and 1 + 3 redundancy codes, with the strip size equal to the GPFS block size. Thus, for every block or strip written to the disks, N replicas of that block or strip are also written. This configuration results in track width of three or four strips and storage efficiency of 33% or 25%.

End-to-end checksum

Most implementations of RAID codes implicitly assume that disks reliably detect and report faults, hard-read errors, and other integrity problems. However, studies show that disks do not report some read faults and occasionally fail to write data, although it was reported that the data was written.

These errors are often referred to as *silent errors*, *phantom-writes*, *dropped-writes*, or *off-track writes*. To compensate for these shortcomings, GPFS Native RAID implements an end-to-end checksum that detects silent data corruption that is caused by disks or other system components that transport or manipulate the data.

When an NSD client is writing data, a checksum of 8 bytes is calculated and appended to the data before it is transported over the network to the GPFS Native RAID server. On reception, GPFS Native RAID calculates and verifies the checksum. GPFS Native RAID stores the data, a checksum, and version number to disk and logs the version number in its metadata for future verification during read.

When GPFS Native RAID reads disks to satisfy a client read operation, it compares the disk checksum against the disk data and the disk checksum version number against what is stored in its metadata. If the checksums and version numbers match, GPFS Native RAID sends the data along with a checksum to the NSD client. If the checksum or version numbers are invalid, GPFS Native RAID reconstructs the data by using parity or replication and returns the reconstructed data and a newly generated checksum to the client. Thus, both silent disk read errors and lost or missing disk writes are detected and corrected.

Declustered RAID

Compared to conventional RAID, GPFS Native RAID implements a sophisticated data and spare space disk layout scheme that allows for arbitrarily sized disk arrays and reduces the overhead to clients that are recovering from disk failures. To accomplish this configuration, GPFS Native RAID uniformly spreads or declusters user data, redundancy information, and spare space across all the disks of a declustered array. A conventional RAID layout is compared to an equivalent declustered array in Figure 1-58 on page 80.

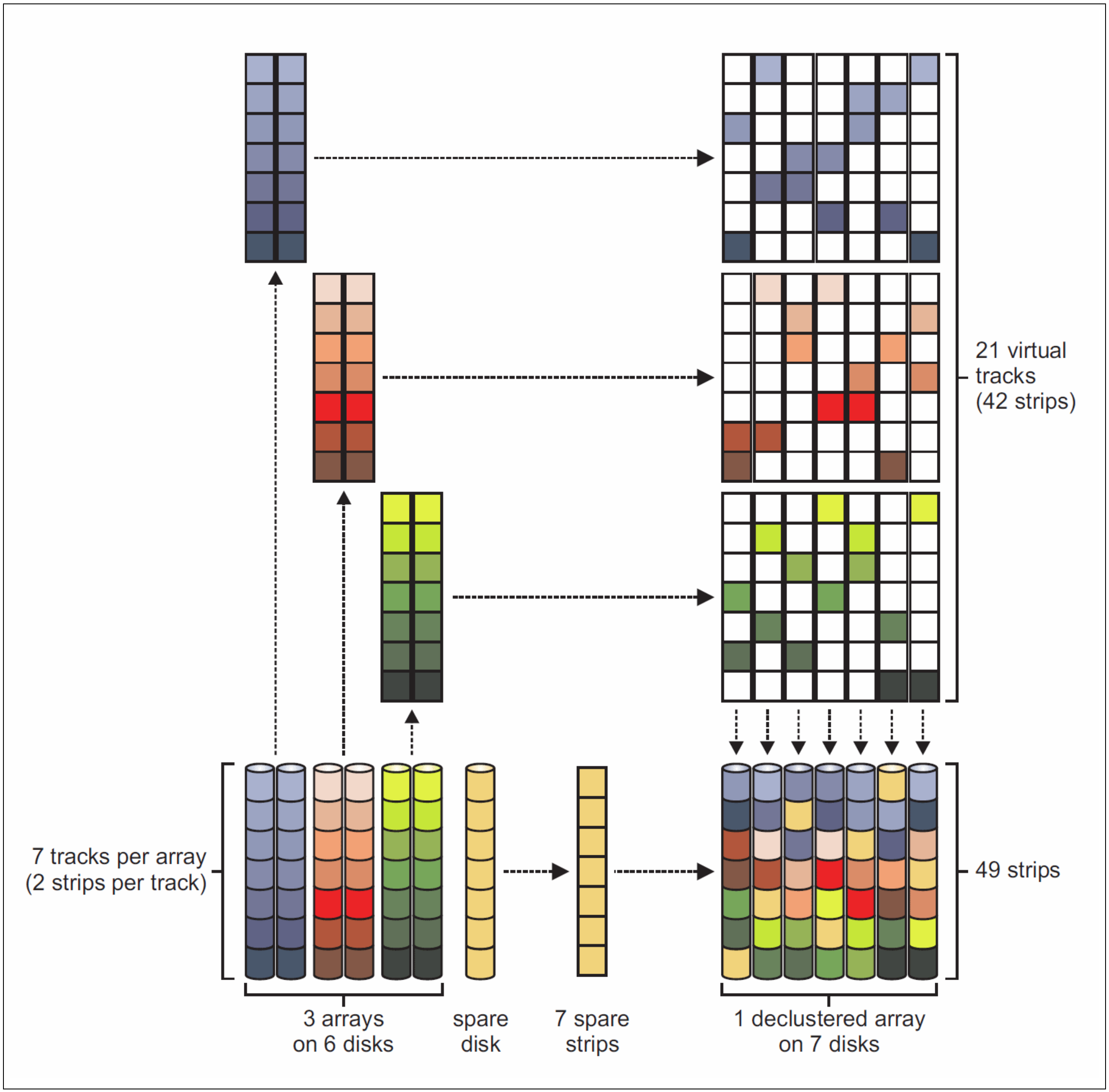

21 virtual tracks (42 strips)
7 tracks per array (2 strips per track)
49 strips
3 arrays on 6 disks
spare disk
7 spare strips
1 declustered array on 7 disks

Figure 1-58 Conventional RAID versus declustered RAID layouts

Figure 1-58 shows an example of how GPFS Native RAID improves client performance during rebuild operations by using the throughput of all disks in the declustered array. This is illustrated by comparing a conventional RAID of three arrays versus a declustered array, both using seven disks. A conventional 1-fault-tolerant 1 + 1 replicated RAID array is shown with three arrays of two disks each (data and replica strips) and a spare disk for rebuilding. To decluster this array, the disks are divided into seven tracks, two strips per array. The strips from each group are then spread across all seven disk positions, for a total of 21 virtual tracks. The strips of each disk position for every track are then arbitrarily allocated onto the disks of the declustered array (in this case, by vertically sliding down and compacting the strips from above). The spare strips are uniformly inserted, one per disk.

As illustrated in Figure 1-59 on page 81, a declustered array significantly shortens the time that is required to recover from a disk failure, which lowers the rebuild overhead for client applications. When a disk fails, erased data is rebuilt by using all of the operational disks in the declustered array, the bandwidth of which is greater than the fewer disks of a conventional RAID group. If another disk fault occurs during a rebuild, the number of impacted tracks that require repair is markedly less than the previous failure and less than the constant rebuild overhead of a conventional array.

The decrease in declustered rebuild impact and client overhead might be a factor of three to four times less than a conventional RAID. Because GPFS stripes client data across all the storage nodes of a cluster, file system performance becomes less dependent upon the speed of any single rebuilding storage array.

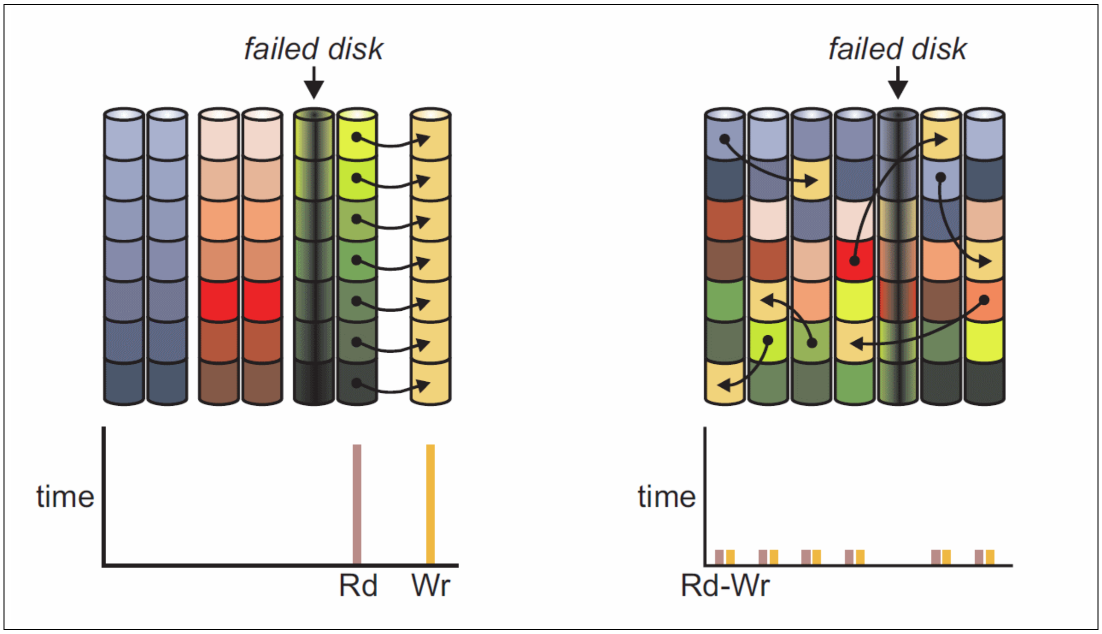


Figure 1-59 Lower rebuild overhead in conventional RAID versus declustered RAID

When a single disk fails in the 1-fault-tolerant 1 + 1 conventional array on the left, the redundant disk is read and copied onto the spare disk, which requires a throughput of seven strip I/O operations. When a disk fails in the declustered array, all replica strips of the six impacted tracks are read from the surviving six disks and then written to six spare strips, for a throughput of two strip I/O operations. As shown in Figure 1-59, disk read and write I/O throughput during the rebuild operations.

Disk configurations

This section describes recovery group and declustered array configurations.

Recovery groups

GPFS Native RAID divides disks into recovery groups in which each disk is physically connected to two servers: primary and backup. All accesses to any of the disks of a recovery group are made through the active primary or backup server of the recovery group.

Building on the inherent NSD failover capabilities of GPFS, when a GPFS Native RAID server stops operating because of a hardware fault, software fault, or normal shutdown, the backup GPFS Native RAID server seamlessly assumes control of the associated disks of its recovery groups.

Typically, a JBOD array is divided into two recovery groups that are controlled by different primary GPFS Native RAID servers. If the primary server of a recovery group fails, control automatically switches over to its backup server. Within a typical JBOD, the primary server for a recovery group is the backup server for the other recovery group.

Figure 1-60 illustrates the ring configuration where GPFS Native RAID servers and storage JBODs alternate around a loop. A particular GPFS Native RAID server is connected to two adjacent storage JBODs and vice versa. The ratio of GPFS Native RAID server to storage JBODs is thus one-to-one. Load on servers increases by 50% when a server fails.

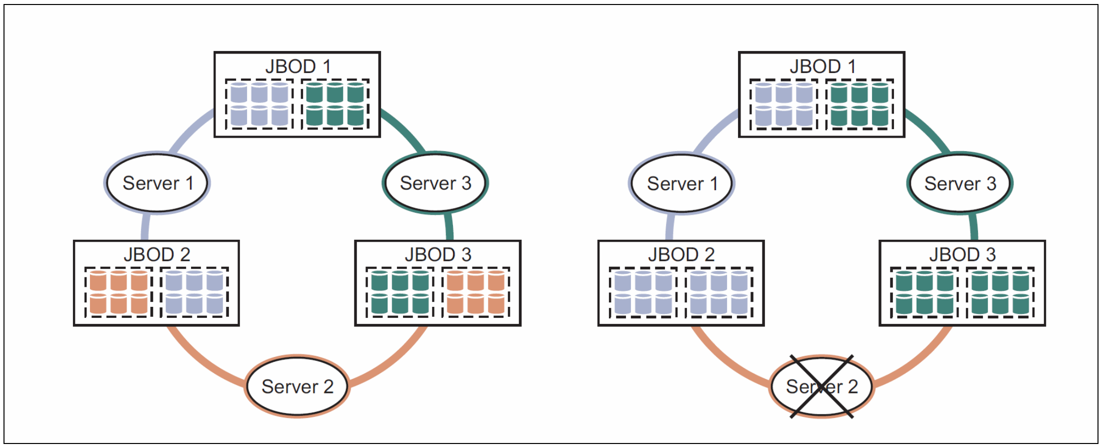


Figure 1-60 GPFS Native RAID server and recovery groups in a ring configuration

Declustered arrays

A declustered array is a subset of the physical disks (pdisks) in a recovery group across which data, redundancy information, and spare space are declustered. The number of disks in a declustered array is determined by the RAID code-width of the VDisks that are housed in the declustered array. One or more declustered arrays can exist per recovery group. Figure 1-61 on page 83 illustrates a storage JBOD with two recovery groups, each with four declustered arrays.

A declustered array can hold one or more VDisks. After redundancy codes are associated with VDisks, a declustered array simultaneously contains Reed-Solomon and replicated VDisks.

If the storage JBOD supports multiple disks that are physically mounted together on removable carriers, removal of a carrier temporarily disables access to all of the disks in the carrier. Thus, pdisks on the same carrier must not be in the same declustered array, as VDisk redundancy protection is weakened upon carrier removal.

Declustered arrays are normally created at recovery group creation time but new arrays are created or existing arrays are grown by adding pdisks later.

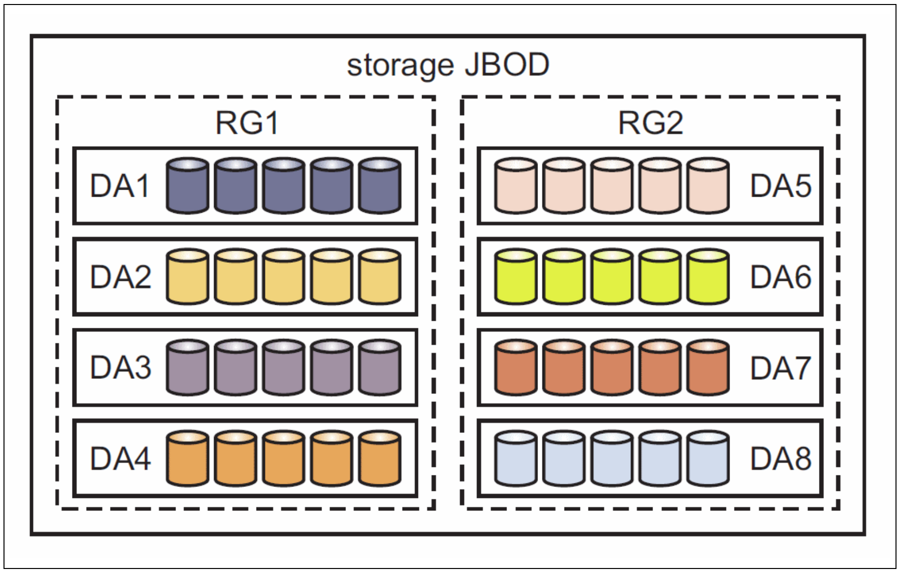


Figure 1-61 Example of declustered arrays and recovery groups in storage JBOD

Virtual and physical disks

A VDisk is a type of NSD that is implemented by GPFS Native RAID across all the pdisks of a declustered array. Multiple VDisks are defined within a declustered array, typically Reed-Solomon VDisks for GPFS user data and replicated VDisks for GPFS metadata.

Virtual disks

Whether a VDisk of a particular capacity is created in a declustered array depends on its redundancy code, the number of pdisks and equivalent spare capacity in the array, and other small GPFS Native RAID overhead factors. The **mmcrvdisk** command automatically configures a VDisk of the largest possible size a redundancy code and configured spare space of the declustered array.

In general, the number of pdisks in a declustered array cannot be less than the widest redundancy code of a VDisk plus the equivalent spare disk capacity of a declustered array. For example, a VDisk that uses the 11-strip-wide 8 + 3p Reed-Solomon code requires at least 13 pdisks in a declustered array with the equivalent spare space capacity of two disks. A VDisk that uses the three-way replication code requires at least five pdisks in a declustered array with the equivalent spare capacity of two disks.

VDisks are partitioned into virtual tracks, which are the functional equivalent of a GPFS block. All VDisk attributes are fixed at creation and cannot be altered.

Physical disks

A pdisk is used by GPFS Native RAID to store user data and GPFS Native RAID internal configuration data.

A pdisk is either a conventional rotating magnetic-media disk (HDD) or a solid-state disk (SSD). All pdisks in a declustered array must have the same capacity.

Pdisks are also assumed to be dual-ported with one or more paths that are connected to the primary GPFS Native RAID server and one or more paths that are connected to the backup server. Often there are two redundant paths between a GPFS Native RAID server and connected JBOD pdisks.

Solid-state disks

GPFS Native RAID assumes several solid-state disks (SSDs) in each recovery group in order to redundantly log changes to its internal configuration and fast-write data in non-volatile memory, which is accessible from either the primary or backup GPFS Native RAID servers after server failure. A typical GPFS Native RAID log VDisk might be configured as three-way replication over a dedicated declustered array of four SSDs per recovery group.

Disk hospital

The disk hospital is a key feature of GPFS Native RAID that asynchronously diagnoses errors and faults in the storage subsystem. GPFS Native RAID times out an individual pdisk I/O operation after approximately 10 seconds, limiting the effect of a faulty pdisk on a client I/O operation. When a pdisk I/O operation results in a timeout, an I/O error, or a checksum mismatch, the suspect pdisk is immediately admitted into the disk hospital. When a pdisk is first admitted, the hospital determines whether the error was caused by the pdisk or by the paths to it. Although the hospital diagnoses the error, GPFS Native RAID, if possible, uses VDisk redundancy codes to reconstruct lost or erased strips for I/O operations that otherwise are used the suspect pdisk.

Health metrics

The disk hospital maintains internal health assessment metrics for each pdisk: time badness, which characterizes response times; and data badness, which characterizes media errors (hard errors) and checksum errors. When a pdisk health metric exceeds the threshold, it is marked for replacement according to the disk maintenance replacement policy for the declustered array.

The disk hospital logs selected Self-Monitoring, Analysis, and Reporting Technology (SMART) data, including the number of internal sector remapping events for each pdisk.

Pdisk discovery

GPFS Native RAID discovers all connected pdisks when it starts, and then regularly schedules a process that rediscovers a pdisk that newly becomes accessible to the GPFS Native RAID server. This configuration allows pdisks to be physically connected or connection problems to be repaired without restarting the GPFS Native RAID server.

Disk replacement

The disk hospital tracks disks that require replacement according to the disk replacement policy of the declustered array. The disk hospital is configured to report the need for replacement in various ways. The hospital records and reports the FRU number and physical hardware location of failed disks to help guide service personnel to the correct location with replacement disks.

When multiple disks are mounted on a removable carrier, each of which is a member of a different declustered array, disk replacement requires the hospital to temporarily suspend other disks in the same carrier. To guard against human error, carriers are also not removable until GPFS Native RAID actuates a solenoid controlled latch. In response to administrative commands, the hospital quiesces the appropriate disks, releases the carrier latch, and turns on identify lights on the carrier that is next to the disks that require replacement.

After one or more disks are replaced and the carrier is re-inserted, in response to administrative commands, the hospital verifies that the repair took place. The hospital also automatically adds any new disks to the declustered array, which causes GPFS Native RAID to rebalance the tracks and spare space across all the disks of the declustered array. If service personnel fail to reinsert the carrier within a reasonable period, the hospital declares the disks on the carrier as missing and starts rebuilding the affected data.

Two Declustered Arrays/Two Recovery Group

Figure 1-62 shows a “Two Declustered Array/Two Recovery Group” configuration of a Disk Enclosure. This configuration is referred to as 1/4 populated. The configuration features four SDDs (shown in dark blue in Figure 1-62) in the first recovery group and the four SSDs (dark yellow in Figure 1-62) in the second recovery group.

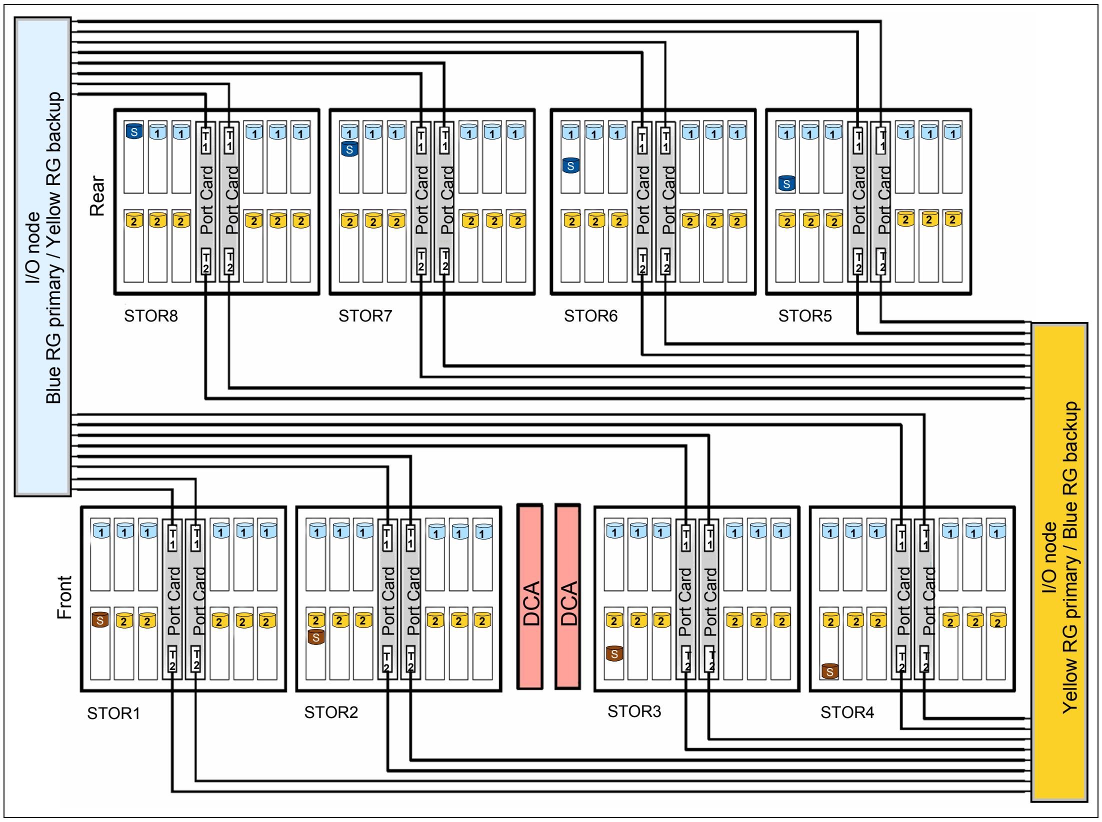


Figure 1-62 Two Declustered Array/Two Recovery Group DE configuration

Four Declustered Arrays/Two Recovery Group

Figure 1-63 shows a Four Declustered Array/Two Recovery Group configuration of a disk enclosure. This configuration is referred to as 1/2 populated. The configuration features four SDDs (shown in dark blue in Figure 1-63) in the first recovery group and the four SSDs (dark yellow in Figure 1-63) in the second recovery group.

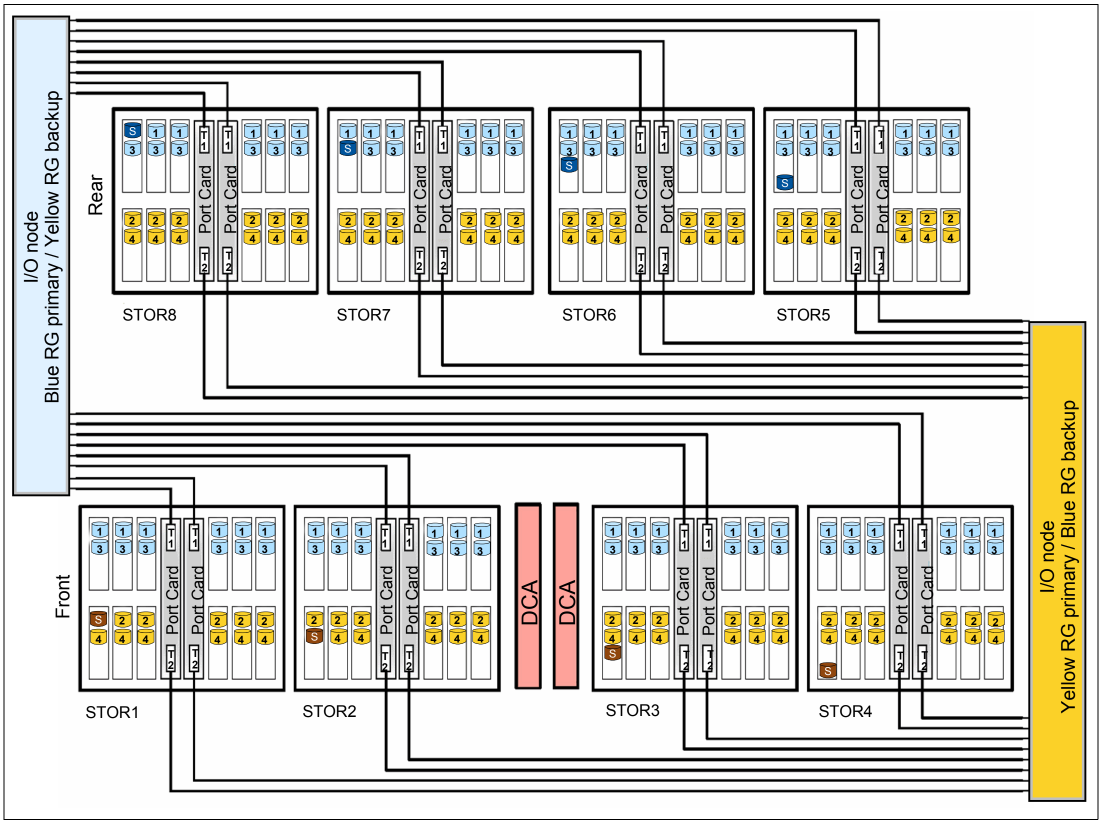


Figure 1-63 Four Declustered Array/Two Recovery Group DE configuration

Six Declustered Arrays/Two Recovery Group

Figure 1-64 shows a Six Declustered Array/Two Recovery Group configuration of a Disk Enclosure. This configuration is referred to as 3/4 populated. This configuration features four SDDs (shown in dark blue in Figure 1-64) in the first recovery group and the four SSDs (dark yellow in Figure 1-64) in the second recovery group.

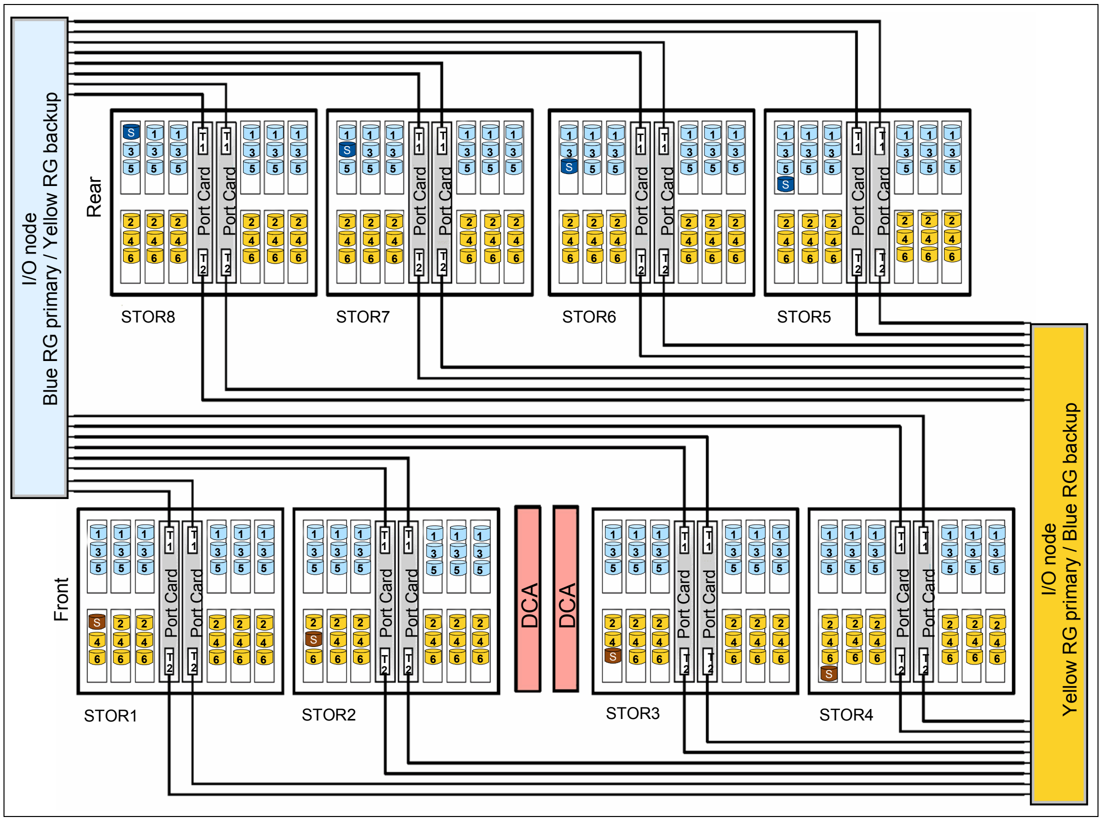


Figure 1-64 Six Declustered Array/Two Recovery Group DE configuration

Eight Declustered Arrays/Two Recovery Group

Figure 1-65 describes a Eight Declustered Array / Two Recovery Group configuration of a Disk Enclosure. This configuration is referred to as fully populated. The configuration features four SDDs (shown in dark blue in Figure 1-65) in the first recovery group and the four SSDs (dark yellow in Figure 1-65) in the second recovery group.

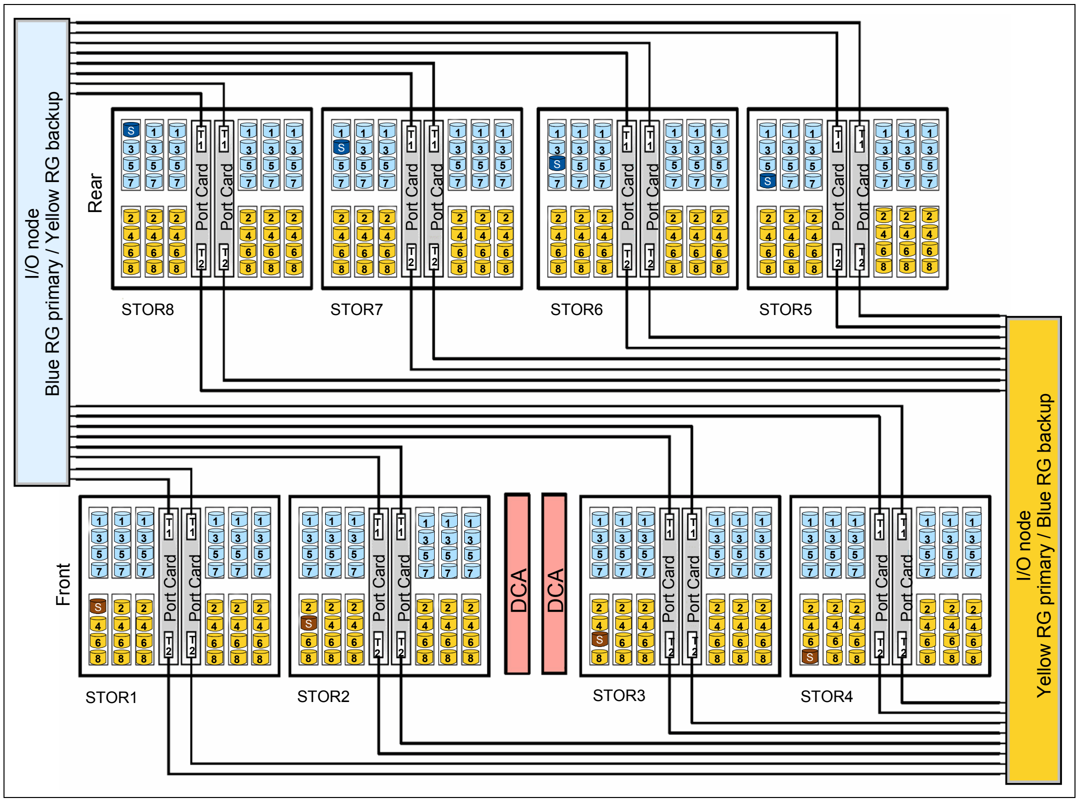


Figure 1-65 Eight Declustered Array/Two Recovery Group DE configuration

1.9.7 IBM Parallel Environment

Parallel Environment is a high-function development and execution environment for parallel applications (including distributed-memory and message-passing applications that run across multiple nodes). This environment is designed to help organizations develop, test, debug, tune, and run high-performance parallel applications that are written in C, C++, and Fortran on Power Systems clusters. Parallel Environment also runs on AIX or Linux.

Parallel Environment includes the following components, as shown in Figure 1-66 on page 89:

- The Parallel Operating Environment (POE) for submitting and managing jobs.
- The IBM MPI and LAPI libraries for communication between parallel tasks.
- A parallel debugger (pdb) for debugging parallel programs.
- IBM High Performance Computing Toolkit for analyzing performance of parallel and serial applications.

For more information about cluster products, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pe.doc/pebooks.html

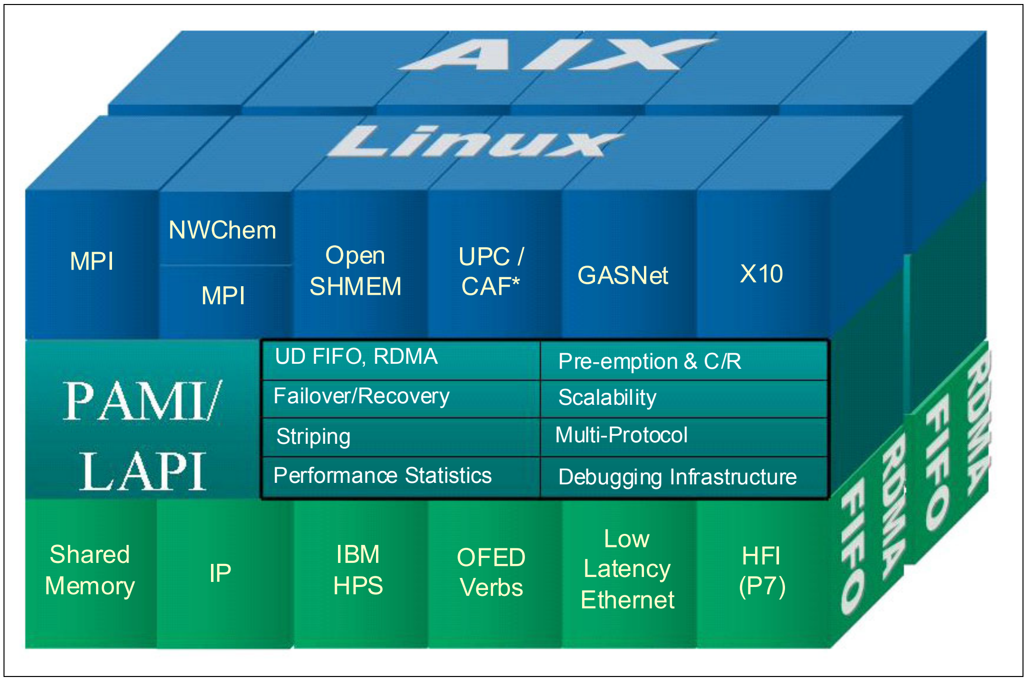


Figure 1-66 Parallel Environment architecture

POE Runtime environment

POE enables users to develop and execute parallel applications across multiple operating system images (nodes). POE includes parallel application compile scripts for programs that are written in C, C++, and Fortran, and a command line interface to submit commands and applications in parallel. POE also provides an extensive set of options and more functions to fine-tune the application environment to suit the execution of the application and system environment.

POE works with IBM LoadLeveler to assist in resource management, job submission, node allocation and includes the following features:

- Provides scalable support to more than one million tasks:
  - Hierarchical tree support
- Corefile support:
  - Separate corefile per task
  - Supports lightweight corefile that can dump to output
- Supports third-party schedulers:
  - Supports JCF submission through POE command option
- Provides job termination and resource cleanup:
  - Handles shared memory cleanup used
- Supports per task output labeling
- Supports multiprotocol applications:
  - Mixed MPI, LAPI, UPC, CAF, OpenShmem, and so on
- POE co-scheduler support to minimize OS jitter impacts:
  - CPU time allocation through priority adjustment daemon configurable by the system administrator on job class basis. User further adjusts settings through environment variable

- Stand-alone job launch without any resource manager (interactive) – US and IP:
  - System administrator controlled configuration file per operating system
  - Loads network resource table for User Space jobs
- Support for SSH and RSH authentication
- Support for Affinity control of CPU and memory and adapter:
  - Support Resources Set (RSET) control on AIX
  - Support OpenMP interoperability
- Support for third-party resource manager APIs
- Support application Checkpoint/Restart by using application containers:
  - AIX 6.1
  - Linux – target is 2011 but dependent on community acceptance of Checkpoint/Restart
- Support application Checkpoint/Restart with migration of subset of tasks:
  - AIX 7.1 – 2011
  - Linux – 2011 – dependent on community acceptance of C/R
- Support for multiple levels of PE runtime libraries
  - Rolling migration support – 2011

Unified messaging layer

The unified messaging layer architecture is shown in Figure 1-67.

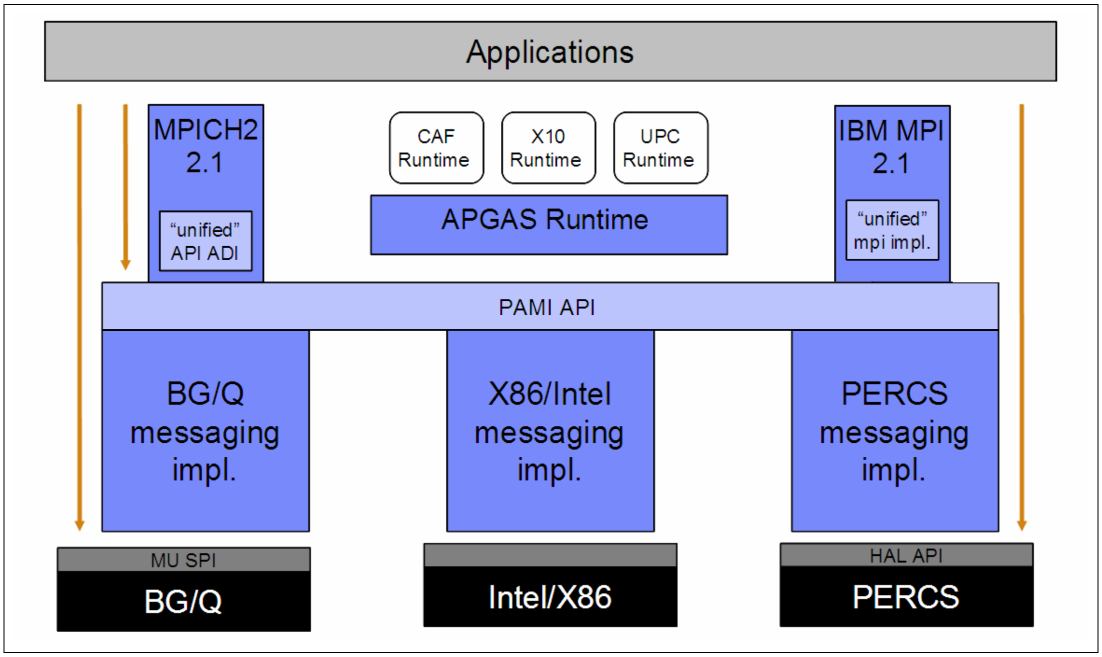


Figure 1-67 Unified messaging layer architecture

Active Messaging Interface: Beginning with Parallel Environment Runtime Edition version 1.1 for AIX, the low-level application programming interface (LAPI) was replaced by a new messaging API called the *parallel active messaging interface* (PAMI). LAPI supported only point-to-point communications and PAMI supports both point-to-point and collective communications.

This change means that PE MPI now runs on top of PAMI. Existing calls to LAPI point-to-point functions are replaced by PAMI point-to-point functions, and some collectives are replaced by PAMI collectives.

LAPI is still supported by PE. However, because LAPI no longer receives any functional enhancements, and PE removes support for LAPI, users must migrate from LAPI to PAMI.

For more information about installing LAPI or PAMI, see *Parallel Environment Runtime Edition for AIX V1.1: Installation*, SC23-6780. For more information about migrating from LAPI to PAMI, and about PAMI in general, see the *IBM Parallel Environment Runtime Edition: PAMI Programming Guide*, SA23-2273.

MPI and LAPI provide communication between parallel tasks, enabling application programs to be *parallelized*.

MPI provides message passing capabilities that enable parallel tasks to communicate data and coordinate execution. The message passing routines call communication subsystem library routines to handle communication among the processor nodes.

LAPI differs from MPI in that LAPI is based on an *active message style* mechanism that provides a one-sided communications model in which one process initiates an operation. The completion of that operation does not require any other process to take a complementary action. LAPI also is the common transport layer for MPI and is packaged as part of the AIX RSCT component.

Parallel Environment debug environment

The parallel debugger (pdb) streamlines debugging of parallel applications, presenting the user with a single command line interface that supports most dbx/gdb execution control commands and the ability to examine running tasks. To simplify the management of a large numbers of tasks, dbx/gdb grpups tasks together so that the user might examine any subset of the debugged tasks.

PDB allows users to invoke a POE job or attach to a running POE job and place it under debug control. pdb starts a remote dbx/gdb session for each task of the POE job that is placed under debugger control.

PDB provides advanced features, including: dynamic tasking support, multiple console display, and output filtering.

Performance Tools environment

This section describes the performance tools.

HPC toolkit

The IBM HPC Toolkit is a collection of tools that you use to analyze the performance of parallel and serial applications that are written in C or FORTRAN and running the AIX or Linux operating systems on IBM Power Systems Servers. The Xprof GUI also supports C++ applications. These tools perform the following functions:

- Provide access to hardware performance counters for performing low-level analysis of an application, including analyzing cache usage and floating-point performance.
- Profile and trace an MPI application for analyzing MPI communication patterns and performance problems.
- Profile an OpenMP application for analyzing OpenMP performance problems and to help you determine whether an OpenMP application properly structures its processing for best performance.
- Profile the application I/O for analyzing the I/O patterns of an application and determine whether you can improve the I/O performance of the application.
- Profile the execution of an application for identifying hot spots in the application, and for locating relationships between functions in your application to help you better understand the performance of the application.

HPCS Toolkit

Figure 1-68 on page 93 shows the following toolkit high-level design flow:

- Based on existing IBM HPC Toolkit for application tuning
- HPCS Toolkit is a set of productivity enhancement technologies:
  - Performance Data Collection (extensible):
    - Scalable, dynamic, programmable
    - Binary: no source code modification to instrument application
    - Retains ability to correlate all performance data with source code
  - Bottleneck Discovery (extensible):
    - Make sense of the performance data
    - Mines the performance data to extract bottlenecks
  - Solution Determination (extensible):
    - Make sense of the bottlenecks
    - Mines bottlenecks and suggests system solutions (hardware or software)
    - Assist compiler optimization (including custom code transformations)
  - Performance Visualization (extensible):
    - Performance Data/Bottleneck/Solution Information feedback to User
    - Logging (textual information)
    - Compiler feedback
    - Output to other tools (for example, Kojak analysis, Paraver visualization, and Tau)
- HPCS Toolkit provides Automated Framework for Performance Analysis:
  - Intelligent automation of performance evaluation and decision system
  - Interactive capability with graphical/visual interface always available, but always optional

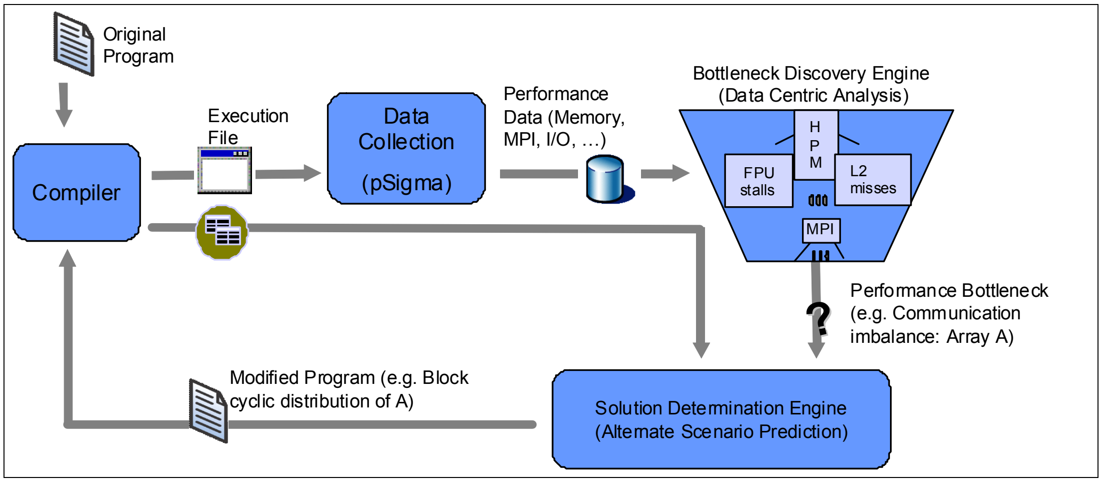

Original Program
Compiler
Execution File
Data Collection (pSigma)
Performance Data (Memory, MPI, I/O, …)
Bottleneck Discovery Engine (Data Centric Analysis)
H P M
FPU stalls
L2 misses
MPI
Performance Bottleneck (e.g. Communication imbalance: Array A)
?
Solution Determination Engine (Alternate Scenario Prediction)
Modified Program (e.g. Block cyclic distribution of A)

Figure 1-68 HPCS Toolkit high-level design flow

1.9.8 LoadLeveler

LoadLeveler is a parallel job scheduling system that allows users to run more jobs in less time by matching the processing needs and priority of each job with the available resources, which maximizes resource utilization. LoadLeveler also provides a single point of control for effective workload management and supports high-availability configurations. LoadLeveler also offers detailed accounting of system utilization for tracking or charge back.

When jobs are submitted to LoadLeveler, the jobs are not executed in the order of submission. Instead, LoadLeveler dispatches jobs that are based on their priority, resource requirements, and special instructions. For example, administrators specify that long-running jobs run only on off-hours that short-running jobs are scheduled around long-running jobs or that jobs that belong to certain users or groups are prioritized. In addition, resources are tightly controlled. The use of individual machines is limited to specific times, or users, job classes, or LoadLeveler use machines only when the keyboard and mouse are inactive.

LoadLeveler tracks the total resources that are used by each serial or parallel job and offers several reporting options to track jobs and utilization by user, group, account, or type over a specified time. To support charge back for resource use, LoadLeveler incorporates machine speed to adjust charge back rates and is configured to require an account for each job.

LoadLeveler supports high-availability configurations to ensure reliable operation and automatically monitors the available compute resources to ensure that no jobs are scheduled to failed machines. For more information about LoadLeveler, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.loadl.doc/llbooks.html

As shown in Figure 1-69 on page 94, LoadLeveler includes the following characteristics:

- Split into Resource Manager and Job Scheduler
- Better third-party scheduler support through LL RM APIs
- Performance and Scaling improvements for large core systems
- Multi-Cluster support
- Faster and scalable job launch as shown in Figure 1-69 on page 94
- Workflow support with enhanced reservation function
- Database option with xCAT integration

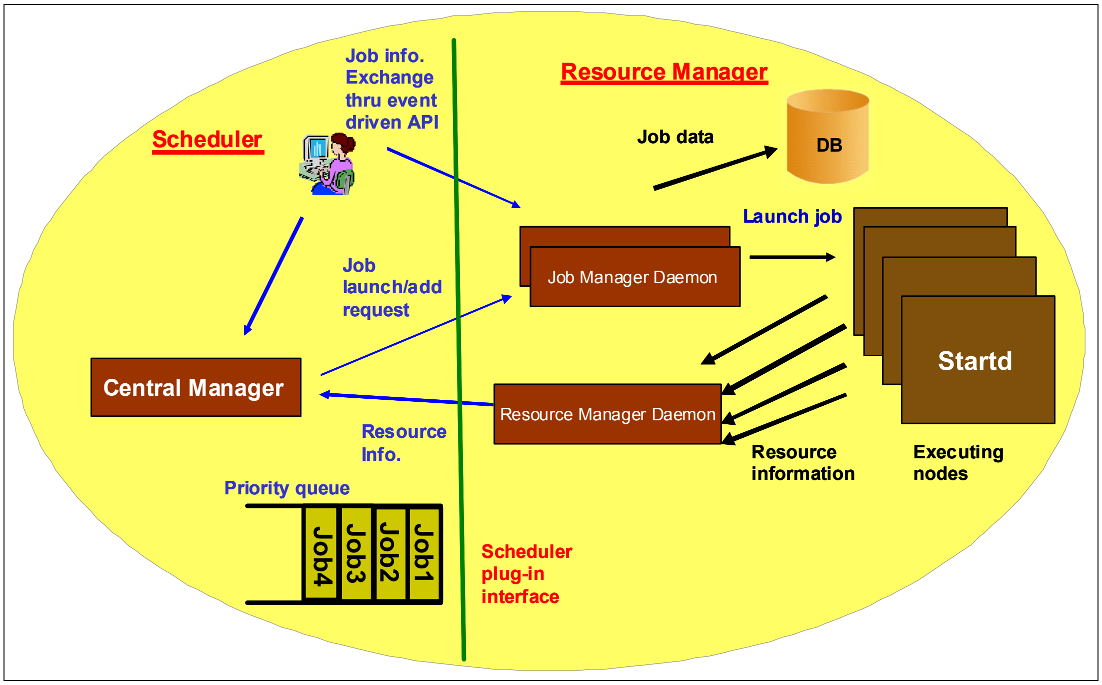


Figure 1-69 Scheduling and resource management flows in LoadLeveler

1.9.9 Engineering and Scientific Subroutine Library

The Engineering and Scientific Subroutine Library (ESSL) is a collection of high-performance mathematical subroutines providing a wide range of functions for many common scientific and engineering applications. The mathematical subroutines are divided into the following computational areas:

- Linear Algebra Subprograms
- Matrix Operations
- Linear Algebraic Equations
- Eigensystem Analysis
- Fourier Transforms, Convolutions, Correlations, and Related Computations
- Sorting and Searching
- Interpolation
- Numerical Quadrature
- Random Number Generation

All of the libraries are designed to provide high levels of performance for numerically intensive computing jobs and provide mathematically equivalent results. The ESSL subroutines are called from application programs that are written in Fortran, C, and C++ that run on the AIX and Linux operating systems. For more information about ESSL, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.essl.doc/esslbooks.html

1.9.10 Parallel ESSL

Parallel ESSL is a scalable mathematical subroutine library for stand-alone clusters or clusters of servers that are connected via a switch and running AIX and Linux. Parallel ESSL supports the Single Program Multiple Data (SPMD) programming model by using the Message Passing Interface (MPI) library.

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

For communication, Parallel ESSL includes the Basic Linear Algebra Communications Subprograms (BLACS), which use MPI. For computations, Parallel ESSL uses the ESSL subroutines (ESSL is a pre-requisite).

The Parallel ESSL subroutines are called from 32-bit and 64-bit application programs that are written in Fortran, C, and C++ that run the AIX and Linux operating systems.

The Parallel ESSL SMP Libraries are provided for use with the IBM Parallel Environment MPI library. You run single or multi-threaded US or IP applications on all types of nodes. However, you cannot simultaneously call Parallel ESSL from multiple threads. For more information about Parallel ESSL, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pessl.doc/pesslbooks.html?resultof=%22%70%65%73%73%6c%22%20

1.9.11 Compilers

This section describes the characteristics and capabilities of the next generation of compilers, as shown in Figure 1-70 on page 96.

- Advanced Memory Hierarchy Optimizations:
  - Data reorganization with loop optimization
  - Static and dynamic delinquent load identification
  - Data pre-fetching
  - Advanced optimizations
- Loop transformation by using the polyhedral framework:
  - Parallelizes and uses data locality in imperfectly nested loops
- Assist Threads:
  - Automatic data pre-fetching
- Compiler Transformation Reports:
  - Provide compile-time information to user through compiler generated reports (XML)
- UPC Compiler:
  - Support for UPC in the XL compiler
  - UPC-specific compiler optimizations
  - Runtime Optimizations

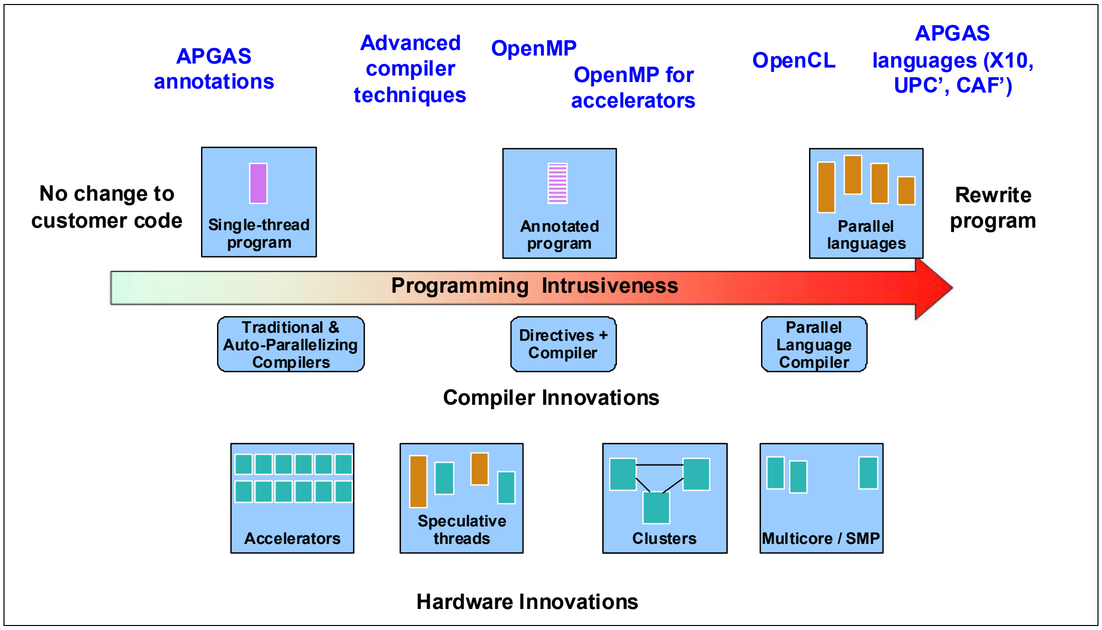


Figure 1-70 Next generation programming languages

With the move to multicore, many-core, and GPU accelerators and the massive size of clusters used today, it is clear that existing programming models and languages cannot make full use of the hardware or deliver high sustained performance in all cases.

This shortcoming creates languages such as CUDA and OpenCL and programming models such as Partitioned Global Address Space (PGAS). Languages such as UPC, CAF, and X10 are better-suited to use the hardware and deliver better performance, as shown in Figure 1-71 on page 97.

New applications written today easily use these new languages or programming models to extract good performance, but it is not realistic to expect that mature applications are rewritten entirely to these new languages. We need to provide a mechanism for mature applications to to scale up to the size of clusters used today and use new hardware capabilities.

A range of choices are available to programmers who are writing new applications that modify mature code. These choices range from adding PGAS annotations to existing code, rewriting critical modules in existing applications in new languages to improve performance and scaling, and calling special library subroutines from mature applications to use new hardware.

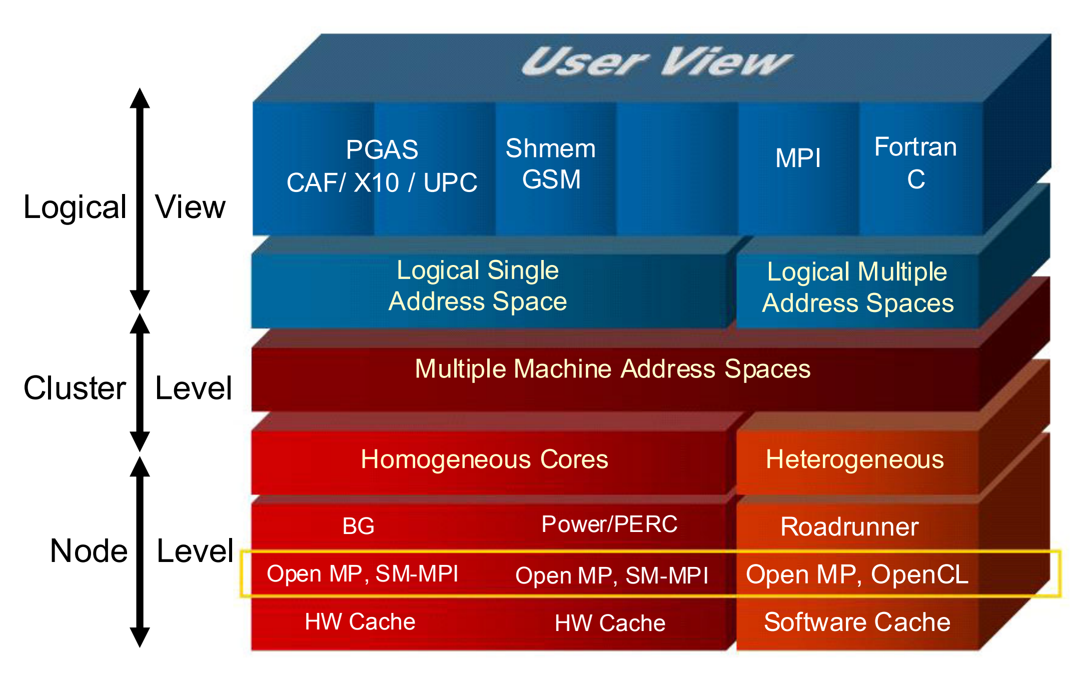
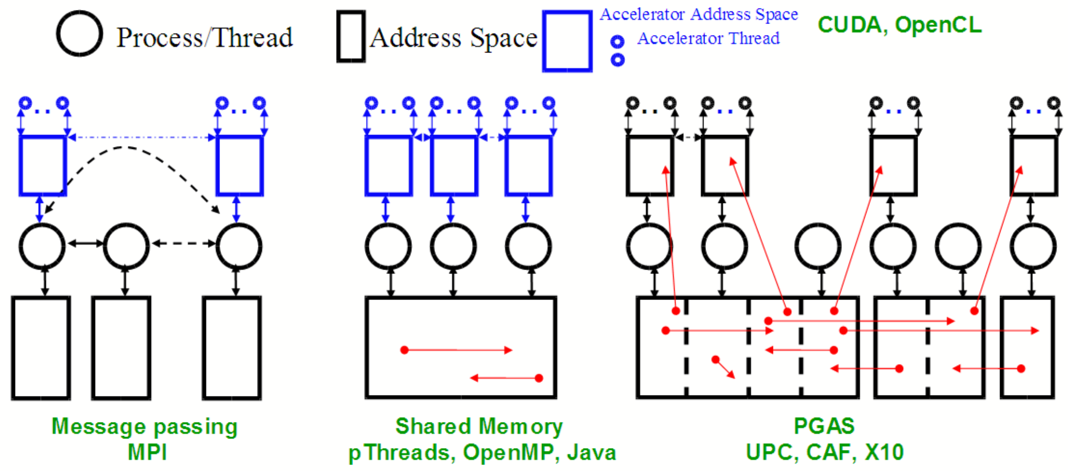


Figure 1-71 Programming models

This is all made possible through substantial investment in compilers, common runtimes and collective frameworks and tools that allow the mixing and matching of all these various techniques, as shown in Figure 1-72.



- Computation is performed in multiple places.
- A place contains data that is operated on remotely.
- Data lives in the place it was created, for its lifetime.
- A datum in one place references a datum in another place.
- Data-structures (arrays) might be distributed across many places.
- Places might have different computational properties

Figure 1-72 Programming models comparison

For more information about programming tools and models, see these websites:

- http://www-01.ibm.com/software/awdtools/xlcpp/
- http://www-01.ibm.com/software/awdtools/fortran/

1.9.12 Parallel Tools Platform

As shown in Figure 1-73, the Parallel Tools Platform (PTP) is an open source project that provides a highly integrated environment for parallel application development. The environment is based on a standard, portable parallel IDE that supports a wide range of parallel architectures and runtime systems and provides a powerful scalable parallel debugger. For more information, see this website:

http://www.eclipse.org/ptp/

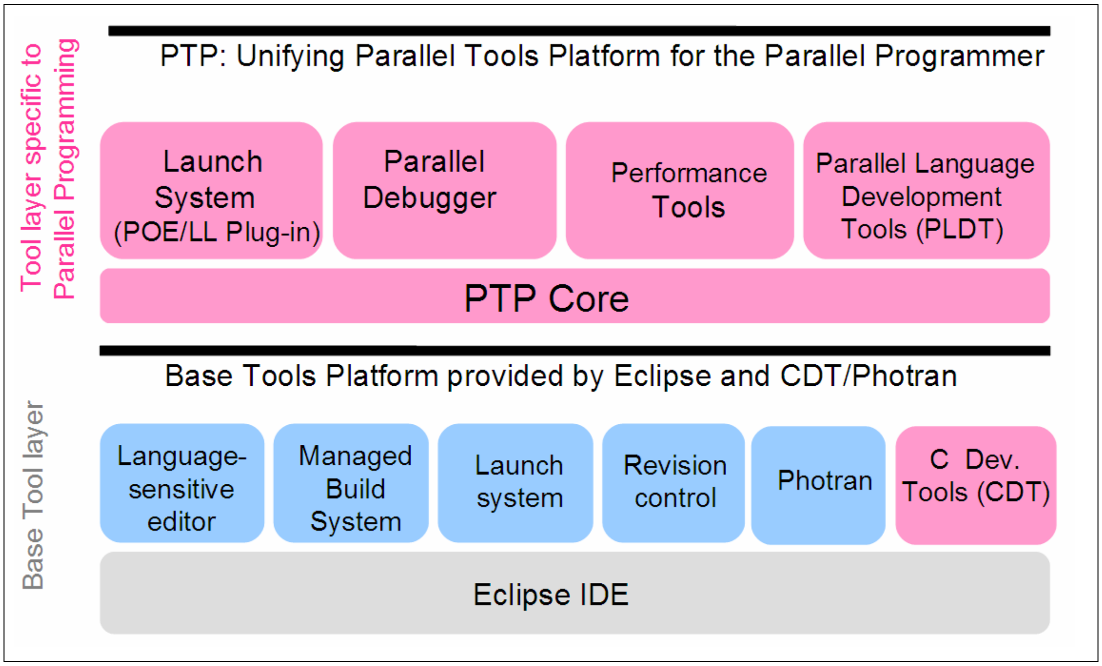

Tool layer specific to Parallel Programming
PTP: Unifying Parallel Tools Platform for the Parallel Programmer
Launch System (POE/LL Plug-in)
Parallel Debugger
Performance Tools
Parallel Language Development Tools (PLDT)
PTP Core
Base Tool layer
Base Tools Platform provided by Eclipse and CDT/Photran
Language-sensitive editor
Managed Build System
Launch system
Revision control
Photran
C Dev. Tools (CDT)
Eclipse IDE

Figure 1-73 Eclipse Parallel Tools Platform (PTP)

2

Application integration

This chapter provides information and best practices on how to integrate the IBM Power Systems 775 cluster and the IBM High Performance Computing (HPC) software stack into practical workload scenarios. This chapter describes the application level characteristics of a Power 775 clustered environment and provides guidance to better take advantage of the new features that are introduced with this system.

This chapter includes the following topics:

- Power 775 diskless considerations
- System capabilities
- Application development
- Parallel Environment optimizations for Power 775
- IBM Parallel Environment Developer Edition for AIX
- Running workloads by using IBM LoadLeveler

2.1 Power 775 diskless considerations

The service node is the only LPAR in the Power 775 frames that is diskfull. The utility, compute, and storage nodes are diskless. *Diskless* is a generic term that describes a system that is not booted off disk. The following node states are available:

- Diskfull node

  For AIX systems, this node has local disk storage that is used for the operating system (a stand-alone node). Diskfull AIX nodes often are installed by using the *NIM rte* or *mksysb install* methods.

- Diskless node

  The operating system is not stored on local disk. For AIX systems, the file systems are mounted from a NIM server. An AIX diskless image is essentially a SPOT, which provides a `/usr` file system for diskless nodes and a root directory. The contents of the file system are used for the initial diskless nodes root directory. This node also provides network boot support.

  You choose to have your diskless nodes as *stateful* or *stateless*. If you want a stateful node, you must use a NIM root resource. If you want a stateless node, you must use a NIM `shared_root` resource.

- Stateful node

  A stateful node maintains its state after it is shut down and rebooted. The node state is any node-specific information that is configured on the node. For AIX diskless nodes, each node has its own NIM root resource that is used to store node-specific information. Each node mounts its own root directory and preserves its state in individual mounted root file systems. When the node is shut down and rebooted, any information that is written to a root file system is available.

- Stateless node

  A stateless node that does not maintain its state after it is shut down and rebooted. For AIX diskless nodes, all of the nodes use the same NIM shared_root resource. Each node mounts the same root directory. Anything that is written to the local root directory is redirected to memory and is lost when the node is shut down. Node-specific information must be re-established when the node is booted.

  The advantage of stateless nodes is that there is less network traffic and fewer resources are used, which is important in a large cluster environment.

2.1.1 Stateless system versus Statelite system

A stateless system is one type of diskless system that does not save data during a reboot. Any data that is written to writable directories is lost. Differences also exist in how AIX and Linux implement stateless. The common recommendation for both platforms on the Power 775 system is to implement statelite. A statelite stateless diskless system offers the option to make some data read/write or persistent.

An AIX stateless system mounts the operating system over NFS. This is a common image that is used by all nodes so that the `/usr` file system cannot be modified. A Linux stateless system loads the entire image into memory (ramdisk) so the user writes to any location.

This information is important because the user manages the overall system, and the user runs applications that are needed to make configuration decisions together. The application user is no longer able to perform simple tasks that they are able to do with diskfull. The use of statelite, image updates, and postscripts handles the customization. This relationship continues for the life of the cluster.

Statelite implementation

xCAT is used to implement statelite. The following tables control the data that is read/write or persistent:

- Litefile: Lists the files or directories that are read/write or persistent
- Statelite: Lists the NFS server in which the persistent data is stored and mounted from during a reboot
- Litetree: This table is usually left blank.

For more information about how to use these tables and the diskless xCAT procedure, see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_AIX_Diskless_Nodes

The user must decide which data to include in the litefile table. Any new entries to the litefile table must be carefully reviewed and tested. This table must be populated before **mkdsklsnode** (for AIX) or **genimage** (for Linux) is run. Example 2-1 shows a litefile table for a Power 775 system.

Example 2-1 litefile table example

```
# tabdump litefile
#image,file,options,comments,disable
"ALL","/etc/basecust","persistent",,
"ALL","/etc/microcode/","rw",,
"ALL","/gpfslog/","persistent",,
"ALL","/var/adm/ras/errlog","persistent",,
"ALL","/var/adm/ras/gpfslog/","persistent",,
"ALL","/var/mmfs/","persistent",,
"ALL","/var/spool/cron/","persistent",,
```

Example 2-1includes the following entries:

- `/etc/basecust`

  The ODM objectrepos files and directories must not be included in the statelite table. Data in the ODM must be accessed during boot and before statelite is applied. However, the user safely mounts the `/etc/basecust` which stores information about changes that are made to the ODM. We are using `/etc/basecust` because the storage nodes boot with multipath enabled, which is not desirable. Currently, we run a script that removes the devices, disables multipath, and rediscovers the disk. If `/etc/basecust` is not added to the statelite table, we must run `/etc/basecust` after every reboot, which takes up to one hour to complete. After `/etc/basecust` is added to statelite, the second discovery is added to the ODM and saved on the NFS mounted space.

- `/etc/microcode`

  This entry is used so that microcode is placed on the storage nodes to update the disk enclosure.

- /gpfslog/

  This entry is an example of a user-defined directory in which we chose to store GPFS tracing logs. The directory needs to be persistent in the event the node becomes inoperable and persistent tracing data must be used to debug after the node is returned to a usable state. GPFS tracing logs become large so they are recommended only for testing and debugging purposes.

- /var/adm/ras/errlog

  The errlog is explicitly made persistent. Originally, we made the entire /var/adm/ras directory persistent, but we found that the console cannot be NFS mounted. You continue to make /var/adm/ras directory persistent and use swcons after boot to move the conslog or only statelite mount /var/adm/ras/errlog. (A postscript is required to create the initial errlog.)

- /var/adm/ras/gpfslog/

  By default, GPFS logs data from the mmfs log files to /var/log/ras and rotates copies by appending the date to the files. Therefore, it is impossible to explicitly list these files in the litefile table. As a result, a logDir attribute is added to GPFS. This entry is used during configuration to route logs to /var/adm/ras/gpfslog instead of /var/log/adm.

- /var/mmfs/

  This directory contains the GPFS configuration for each node. After reboot, the node uses this information to join the GPFS cluster.

- /var/spool/cron/

  This entry allows for the crontab to be persistent.

If you need to modify the statelite configuration, you do so without updating your image. You must reset the node configuration file so that after the statelite config is updated, you rerun **mkdsklsnode** for AIX and **genimage** for Linux. You verify the run by reviewing the /install/nim/shared_root/<image_name>_shared_root/litefile.table file on the service node.

Any changes must be carefully reviewed and tested before implemented in production because unexpected circumstances might occur. In one test, the /tmp/serverlog for PSND logging was added to the statelite configuration and MPI response time was adversely affected. Therefore, the entry was removed from the statelite configuration. An alternative was to use the /etc/PNSD configuration file to define a location, such as GPFS.

The following considerations must be reviewed when statelite is used:

- Avoid making configuration files statelite. Subsequent code updates might add parameters that are overwritten by the old configuration file that is mounted over NFS during boot.
- In Linux, the kernel must be mounted via statelite. Ensure that when a kernel change is made, the data on the NFS-mounted statelite directory is removed before boot.
- Think about space when directories and files are added to statelite. By default, the data is stored on the Service Node and has limited disk space. Ensure that the file system that contains the statelite data has adequate space because all of the nodes are writing to the file system. Filling up the file system leads to adverse affects when applications are run. An external NFS server to which extra disk is added is used and defined in the statelite table instead of in the service node.

An example of a statelite system is shown in Example 2-2 on page 103.

Example 2-2 Statelite table example

```
# df  | awk '{print($1,$7)}'
Filesystem Mounted
<service node>:/install/nim/shared_root/GOLD_71Bdskls_1142A_HPC_shared_root /
<service node>:/install/nim/spot/GOLD_71Bdskls_1142A_HPC/usr /usr
<service node/external nfs ip>:/nodedata /.statelite/persistent
/.statelite/persistent/<node hostname>/etc/basecust /etc/basecust
/.default/etc/microcode /etc/microcode
/.statelite/persistent/<node hostname>/gpfslog/ /gpfslog
/.statelite/persistent/<node hostname>/var/adm/ras/errlog /var/adm/ras/errlog
/.statelite/persistent/<node hostname>/var/adm/ras/gpfslog/ /var/adm/ras/gpfslog
/.statelite/persistent/<node hostname>/var/mmfs/ /var/mmfs
/.statelite/persistent/<node hostname>/var/spool/cron/ /var/spool/cron
/proc /proc
<service node>:/sn_local /sn_local
<nfs server>:/gpfs/nfs01/u /u
/dev/gpfs2 /gpfs2
/dev/gpfs1 /gpfs1
```

In Example 2-2, the data is represented on the various nodes. On this particular node, the data is shown in Example 2-3.

Example 2-3 Command to gather the data that is represented on the nodes

```
# df | awk '{print($1,$7)}' | grep errlog
/.statelite/persistent/<node hostname>/var/adm/ras/errlog /var/adm/ras/errlog
# ls -l /var/adm/ras/errlog
-rw-r--r--    1 root     system       101103 Oct 23 20:31 /var/adm/ras/errlog
```

The service node data is shown in Example 2-4.

Example 2-4 Data that is shown in the service node

```
# df | awk '{print($1,$7)}' | grep nodedata
/dev/nodedatalv /nodedata
# ls -l /nodedata/<node hostname>/var/adm/ras/errlog
-rw-r--r--    1 root     system       101103 Oct 23 20:31 /nodedata/<node
hostname>/var/adm/ras/errlog
```

If you need to remove a statelite file and the node is not booted, you remove the file directly on the service node.

More modifications to the system might be needed and might not fall under the rules of statelite (specifically, modifications that are tuned, such as `vmo`, password files, and smt settings). Some of these modifications must be made to the image directly; a PostScript is sufficient for others. These modifications are examples that are made to a Power 775 system, as shown in Example 2-5 on page 104.

Example 2-5 Example updates that are made to the image

```
1) xcatchroot -i <spot>"bosdebug -D"
nim -o check <spot>
xcatchroot -i <spot> "vmo -r -o lgpg_size=16777216 -o lgpg_regions=256"
nim -o check <spot>
xcatchroot -i <spot>"smtctl -t 2 -w boot"
nim -o check <spot>
2) emgr fixes. See /usr/lpp/bos.sysmgt/nim/README. Note: When these commands are
run the emgr fix will stay at *Q* but it is applied.
```

Example 2-6 shows examples of updates that are made via postscripts.

Example 2-6 Updates that are made via postscripts

```
no -o rfc1323=1
no -o sb_max=16777216
no -o tcp_sendspace=9437184
no -o tcp_recvspace=9437184
no -o udp_sendspace=131072
no -o udp_recvspace=1048576

cp -p /xcatpost/admin_files/passwd /etc/passwd
cp -p /xcatpost/admin_files/security.user /etc/security/user
cp -p /xcatpost/admin_files/security.passwd /etc/security/passwd
cp -p /xcatpost/admin_files/security.limits /etc/security/limits
cp -p /xcatpost/admin_files/hosts /etc/hosts
/usr/sbin/mkpasswd -c

cp -p /xcatpost/admin_files/.profile /.profile

if [[ ! -f /sn_local ]]
then
mkdir /sn_local
mount -o soft c250f19c04ap01-hf0:/sn_local /sn_local
fi

if [[ ! -f /u/00U00 ]]
then
rm /u
mkdir /u
mount -o soft c250nfs01-pvt:/gpfs/nfs01/u /u
fi

echo "MP_POE_LAUNCH=all" > /etc/poe.limits
echo "COMPAT" > /etc/poe.security

vmo -p -o v_pinshm=1
chdev -l sys0 -a fullcore=true
chdev -l sys0 -a maxuproc=8192

mkdir /gpfs
mkdir /gpfsuser
mkdir /gpfs1
mkdir /gpfs2
```

```
stopsrc -s pnsd
sed 's/log_file_size = 10485760/log_file_size = 104857600/' /etc/PNSD.cfg >
/etc/PNSD.cfg.new
cp -p /etc/PNSD.cfg.new /etc/PNSD.cfg
startsrc -s pnsd
```

2.1.2 System access

This section describes the system access component.

Login/Gateway node

The utility CEC provides users with an entry point to the Power 775 cluster through a utility node, which is configured as a Login/Gateway node. For most configurations, the LPAR utility node is in octant 0, with 75% of processor and memory resources. This resource allocation represents three POWER7 chips and respective memory resources. The remaining 25% (one POWER7 processor and its resources) is designated to the service node (SN). For more information about resource requirements for service and utility nodes, see 1.7.3, “Service node” on page 54, and 1.7.7, “Utility nodes” on page 58.

The Login/Gateway node connects to the public VLAN (GbE or 10GbE) and allows users to access the system resources, which include development and runtime environments. More network adapter cards are attached to available PCI slots and assigned to the Login node LPAR to provide this connectivity, as shown in Figure 2-1.

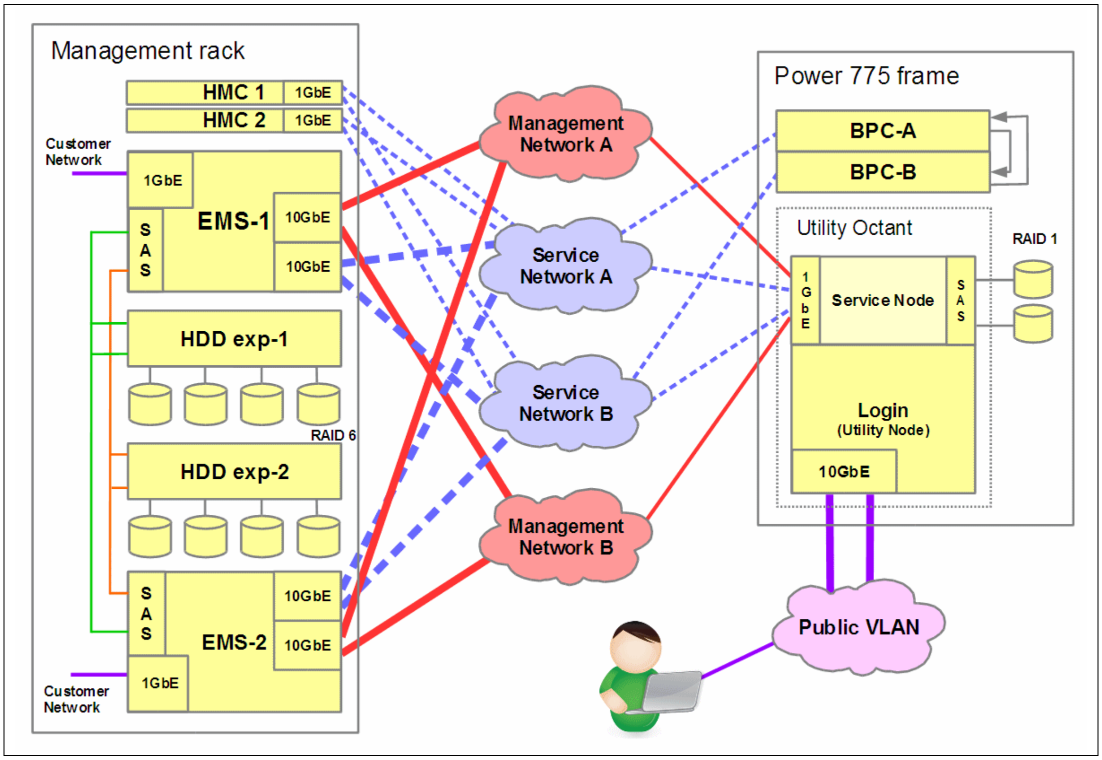


Figure 2-1 Network layout example

User (home) directories are in GPFS. The directory `/sn_local` or the shared storage between the SN and the compute nodes is accomplished if GPFS is added to the SN.

User shared directories often are placed on the DE and mounted over GPFS on the login and compute nodes.

2.2 System capabilities

This section describes the key features of the Power 775 system. For more information, see 1.1, “Overview of the IBM Power System 775 Supercomputer” on page 2.

SN_single and SN_all

On InfinBand systems, sn_single is used to target one adapter and assign the wanted number of windows (MP_instance value) per task. When subsequent jobs are launched, the windows are selected from the next adapter. When sn_all is selected, it assigns the wanted number of windows per task on all available adapters. With HFI, there is only one physical adapter. As a result, sn_single and sn_all are the same.

The number of windows available per adapter differs. For InfiniBand, each adapter has 128 windows (0-127). The HFI interfaces have 220 (36-256). When jobs run on HFI, the adapters windows are selected sequentially and go to the next adapter after 256 windows are reached on the first adapter.

2.3 Application development

There are several tool chain enhancements to aid parallel program development on Power 775 systems. The enhancements provide support for the newest POWER7 processor features, and allow efficient use of parallel programming models for performance leaps on Power 775 clusters.

2.3.1 Xpertise Library compilers support for POWER7 processors

IBM XL C/C++ for AIX, V11.1 and IBM Xpertise Library (XL) Fortran for AIX, V13.1 support POWER7 processors. New features are introduced in support of POWER7 processors.

The new features and enhancements for POWER7 processors fall into the following categories:

- Vector scalar extension data types and intrinsic functions
- MASS libraries for POWER7 processors
- Built-in functions for POWER7 processors
- Compiler options for POWER7 processors

Vector scalar extension data types and intrinsic functions

This release of the compiler supports the Vector Scalar eXtension (VSX) instruction set in the POWER7 processors. New data types and intrinsic functions are introduced to support the VSX instructions. With the VSX intrinsic functions and the original Vector Multimedia eXtension (VMX) intrinsic functions, you manipulate vector operations in your application.

Mathematical Acceleration Subsystem libraries for POWER7

This section provides details about the Mathematical Acceleration Subsystem (MASS) libraries.

Vector libraries

The vector MASS library `libmassvp7.a` contains vector functions that are tuned for the POWER7 architecture. The functions are used in 32-bit mode or 64-bit mode.

Functions supporting previous POWER processors (single-precision or double-precision) are included for POWER7 processors.

The following functions are added in single-precision and double-precision function groups:

- exp2
- exp2m1
- log21p
- log2

SIMD libraries

The MASS SIMD library `libmass_simdp7.a` contains an accelerated set of frequently used math intrinsic functions that provide improved performance over the corresponding standard system library functions.

POWER7 hardware intrinsic

New hardware intrinsic are added to support the following POWER7 processor features:

- New POWER7 prefetch extensions and cache control
- New POWER7 hardware instructions

New compiler option for POWER7 processors

The -qarch compiler option specifies the processor architecture for which code is generated. The -qtune compiler option tunes instruction selection, scheduling, and other architecture-dependent performance enhancements to run best on a specific hardware architecture.

-qarch=pwr7 produces object code containing instructions that run on the POWER7 hardware platforms. With -qtune=pwr7, optimizations are tuned for the POWER7 hardware platforms.

> **For more information:** For more information, see the *XL C/C++ Optimization and Programming Guide*, SC23-5890, and the *XL Fortran Optimization and Programming Guide*, SC23-5836.

2.3.2 Advantage for PGAS programming model

The partitioned global address space (PGAS) programming model is an explicitly parallel programming model that divides the global shared address space into a number of logical partitions. As is the case in the shared memory programming model, each thread addresses the entire shared memory space. In addition, a thread has a logical association with the portion of the global shared address space that is physically on the computational node where the thread is running.

The PGAS programming model is designed to combine the advantages of the shared memory programming model and the message passing programming model. In the message passing programming model, each task has direct access to only its local memory. To access shared data, tasks communicate with one another by sending and receiving messages. UPC introduces the concept of *affinity,* which refers to the physical association between shared memory and a particular thread. The PGAS programming model facilitates data locality exploitation. In addition, the PGAS programming model uses one-sided communication to reduce the cost of inter-thread communication, as shown in Figure 2-2.

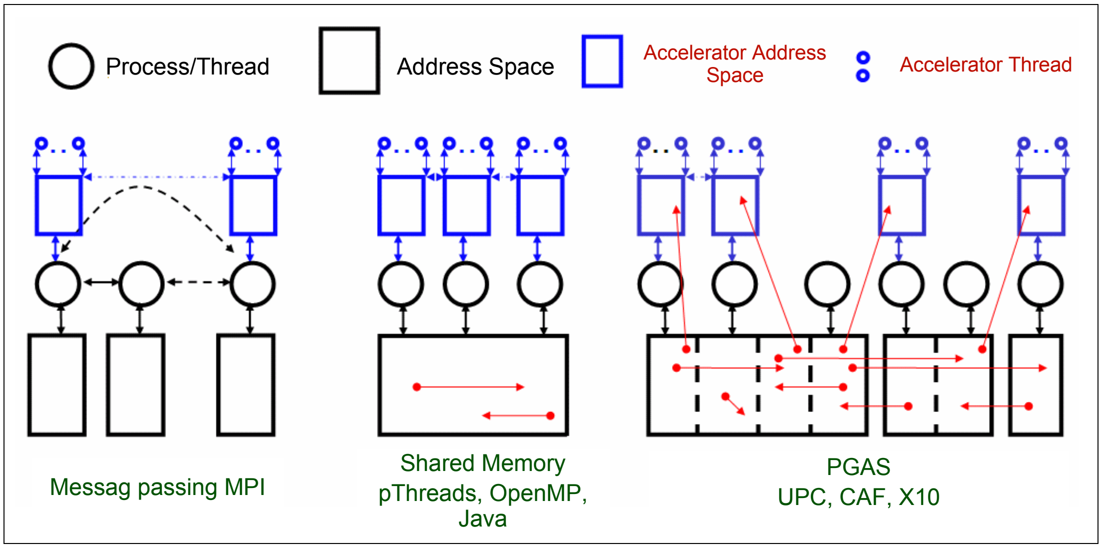


Figure 2-2 Partitioned Global Address Space (PGAS) model

The UPC and other PGAS programming models are the primary target for the global shared memory facility of the hib chip. The UPC compiler and run time generate calls to the parallel active messaging interface (PAMI) communication library. The compiler and run time also optimize the functions that efficiently map to the global shared memory, atomics, and Collective Acceleration Units (CAUs) in the hub chip.

2.3.3 Unified Parallel C

Unified Parallel C (UPC) is an explicitly parallel extension of the C programming language that is based on the PGAS programming model. UPC preserves the efficiency of the C language and supports effective programming on numerous architectures. Scientific applications that are written in UPC efficiently minimize the time that is required to transfer shared data between threads.

UPC includes the following features:

- Explicitly parallel execution model

  The execution model that is used by UPC is called Single Program Multiple Data (SPMD). All threads in a UPC program execute the same program concurrently. Synchronization between threads is explicitly controlled by the user.

- Separate shared and private address spaces

  UPC threads access their private memory space and the entire global shared space. The global shared memory space is partitioned and each thread has a logical association with its local portion of shared memory.

- Synchronization primitives

  UPC makes no implicit assumptions about the interaction between threads. Therefore, it is the responsibility of the user to control thread interaction explicitly with the following synchronization primitives: barriers, locks, and fences.

- Memory consistency

  UPC supports two memory consistency models: strict and relaxed. Strict shared data accesses are implicitly synchronized, relaxed shared memory accesses are not implicitly synchronized. By default, every shared data access follows the relaxed memory consistency model.

The IBM XL UPC compiler is a conforming implementation of the latest UPC language specification (version 1.2), supporting IBM Power Systems that run the Linux operating system.

In addition to extensive syntactic and semantic checks, the XL UPC compiler incorporates the following advanced optimizations that are developed and tailored to reduce the communication cost of UPC programs:

- Shared-object access privatization
- Shared-object access coalescing
- All upc_forall loop optimizations
- Remote shared-object updates
- UPC parallel loop optimizations

Important: IBM XL UPC is a prototype and targets POWER7 and Power7 775 hardware that run RHEL6.

Concepts of data affinity

Data affinity refers to the logical association between a portion of shared data and a thread. Each partition of shared memory space has affinity to a particular thread. A well-written UPC program attempts to minimize communication between threads. An effective strategy for reducing unnecessary communication is to choose a data layout that maximizes accesses to shared data by the thread with affinity to the data.

Concepts of shared and private data

UPC has two types of data: shared data and private data. Shared data is declared with the shared type qualifier, and is allocated in the shared memory space. Private data is declared without the shared type qualifier. One instance of the private data is allocated in the private memory space of each thread. Private data is accessed only by the thread to which it has affinity.

Example 2-7 shows the data layout for different declarations. The following code declares three identifiers of different types.

Example 2-7 Data layout

```
int a; // private scalar variable
shared int b; // shared scalar variable
shared int c[10]; // shared array
```

Assuming four threads, these data objects are allocated, as shown in Figure 2-3 on page 110.

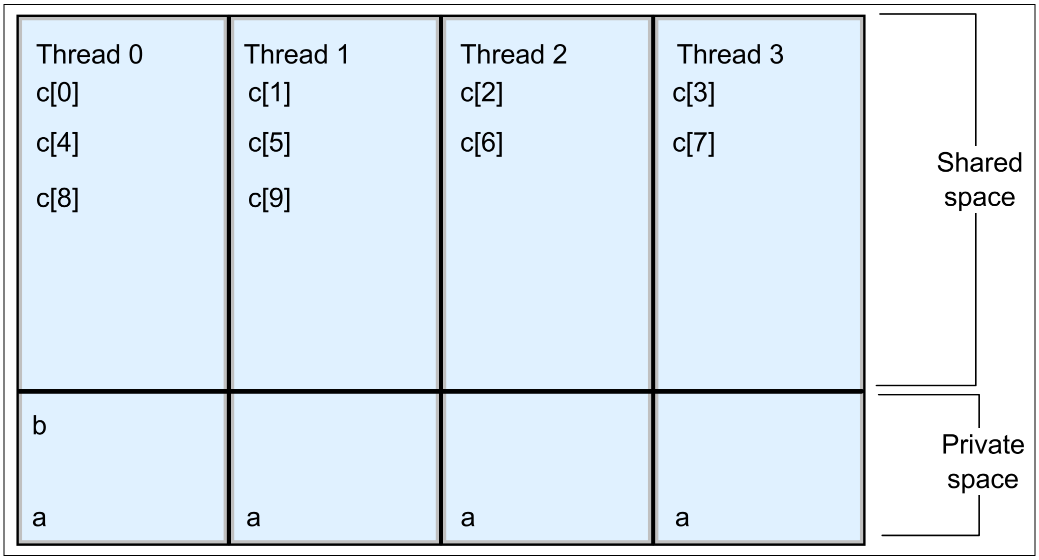


Figure 2-3 Memory allocation and data affinity

Figure 2-3 shows that a thread-local instance of the scalar variable *a* is allocated in the private memory space of each thread. The shared scalar variable *b* is allocated in the shared memory space of thread 0. The array elements of the shared array *c* are allocated in the shared memory space in a round-robin manner among all threads.

Compiler commands

This section provides information about invoking the XL UPC compiler and setting the target execution environment for the program.

Invocation command

To compile a UPC program, use the **`xlupc`** command to invoke the compiler. By default, a `.c` file is compiled as a UPC program unless the option -qsourcetype=c is specified.

> **Important:** If you want to mix `.c` files with `.upc` files in your application, `.c` files must be compiled and linked with **xlupc**.

Execution environments

The execution environment of a UPC program is static or dynamic. In the static execution environment, the number of threads for the target program is known at compile time. In the dynamic execution environment, the number of threads and the number of nodes are not known at compile time.

To set the number of threads for a program in the static environment, you use the -qupc=threads option. For example, to compile the `test.upc` program that runs with four threads, enter the following command:

```
xlupc -o test1 -qupc=threads=4 test.upc
```

To set the number of threads for a program in the dynamic environment, you use the following command:

```
export UPC_NTHREADS=N
```

In this command, N is the number of threads with which the program runs. The environment variable UPC_NTHREADS is used to specify the number of threads that a UPC program runs within the static and dynamic environments.

> **Important:** If the number of threads for a program is specified at compile time, the program is not allowed to attempt to run the compiled program with a different number of threads.

To set the number of nodes for a program in the static environment, you use the compiler option -qupc=dnodes=M, where M is the number of nodes that the program runs on. To compile the `test.upc` program to run with N threads on M nodes, enter the following command:

```
xlupc -qupc=threads=N -qupc=dnodes=M test.upc
```

In this command, N >= M and N must be a multiple of M (that is, N % M = 0).

The executable program must run on the same number of nodes as specified by the -qupc=dnodes option when the program is compiled. To run the executable program, you must use the IBM Parallel Operating Environment (POE). For example, to run the executable program `a.out`, enter the following command:

```
a.out -procs 3 -msg_api 'pgas' -hostfile hosts
```

In this command, the following executables are run:

- -proc: Specifies the number of processes
- -msg_api: Indicates to POE which message passing API is used by the parallel job
- -hostfile: Specifies a host file that contains the names or the IP address of the hosts used

For example, to specify three nodes, you create the host file that lists the IP addresses of the three nodes in the following cluster:

```
1.2.3.4
1.2.3.5
1.2.3.6
```

The following example demonstrates how to specify the number of threads in the dynamic environment:

```
xlupc test.upc
export UPC_NTHREADS=8
a.out -procs 4 -msg_api 'pgas' -hostfile hosts
```

Runtime tuning

In this topic, we describe the environment variable that you use to tune the partitioned global address space (PGAS) run time that is used by UPC.

XLPGASOPTS

The PGAS runtime options affect debugging, performance, and program requirements. You specify these runtime options with the XLPGASOPTS environment variable. You also specify the runtime options on the command line with the prefix **`–xlpgas`**. This environment variable must be set before you run an application. The syntax of the environment variable is shown in Figure 2-4.

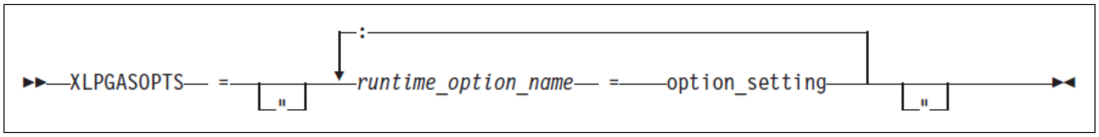


Figure 2-4 Environment variable syntax for XLPGASOPTS

The XLPGASOPTS environment variable accepts multiple colon-separated options.

An individual option also accepts an argument. The runtime_option_name is an option and option_setting is an argument that is accepted by that option. The runtime_option_name is one of the following options:

- stackcheck=*num* | nostackcheck
- stacksize=*num*
- bind=auto | none | file=*filename*

Parameters

If the application is compiled by using -qupc=stackcheck, stackcheck=*num* | nostackcheck controls the stack threshold checking at run time. In this parameter, *num* represents the stack check threshold in terms of a percentage of the total stack size.

If a thread exceeds its stack check threshold for the first time, the executable file issues a warning message on that thread. The stack check warning message is displayed only once for one thread, even if the stack check threshold of the thread is exceeded more than once. The default value for *num* is 80%. If any thread exceeds its local stack size, the executable file issues a message and immediately stops the execution of the application when possible.

If the application is compiled by using **`-qupc=stackcheck`**, nostackcheck prevents the issuing of the runtime warning messages when the stack check threshold is exceeded. The option nostackcheck only turns off the warnings, it does not stop the stack overflow message or program termination.

> **Important:** The option **`stackcheck=num`** takes effect only when you specify **`-qupc=stackcheck`** at compile time.

To remove the instrumentation of **`-qupc=stackcheck`**, you must recompile the program without specifying **`-qupc=stackcheck`** or recompile the program with **`-qupc=nostackcheck`**. Therefore, stacksize=*num* specifies the minimal amount of space in bytes (*num*) that is allocated to the stack that is used by a thread. If the requested stack size is too small, the operating system overrides the requested stack size with the minimal stack size that it allows. If you do not specify the option, the default stack size for each thread is at least 16 MB.

To get the actual stack size for a thread, use the function size_t xlupc_stacksize(). This function is declared in the header file `upc.h` unless XLUPC_NO_EXT is defined before including `upc.h`. XLUPC_NO_EXT, which is a macro that removes any IBM UPC extensions. When this function is called, it returns the actual stack size that is allocated for the current thread. Therefore, bind=auto | none | file=*filename* specifies whether and how threads bind to physical processors. If a thread is bound to a processor, it is executed on the same logical processor during the entire execution.

The *auto* parameter specifies that threads automatically bind to processors. You specify this suboption to enable automatic thread binding on different architectures. When XLPGASOPTS=bind=auto is specified, the run time allocates program threads to hardware threads in a way that minimizes the number of threads assigned to any one processor.

The *none* parameter specifies that no threads bind to processors. When XLPGASOPTS=bind=none is in effect, a thread is suspended and wakes up on another hardware thread (none is the default suboption).

The *file*=`filename` specifies that threads bind to processors in a way that follows the instructions in the `filename` file. You can use XLPGASOPTS=bind=file=filename to manually control how threads bind to processors.

In the *file*=`filename` parameter, `filename` is the name of a binding file, which must be a plain text file. If the binding file is not in current directory, you must specify the path to that file. The path is an absolute path or relative path. The binding file contains one CPU identifier for one line. The CPU identifier in the first line is mapped to thread 0; the CPU identifier in the second line is mapped to thread 1; the CPU identifier in the third line is mapped to thread 2, and so on.

A runtime error message is issued if any of the following situations is true:

- The binding file does not exist.
- There are insufficient entries in the binding file for the requested number of threads.
- The binding file is malformed.

In the following example, you manually control how threads bind to processors with the file `map.txt`:

UPC_NTHREADS=8

./a.out -procs 4 -xlpgasbind=file=./home/user/map.txt

where the content of map.txt is as follows:

```
0
4
8
12
1
5
9
13
```

The binding file `map.txt` in this example maps the following threads to CPUs:

- thread 0 to CPU 0
- thread 1 to CPU 4
- thread 2 to CPU 8
- thread 3 to CPU 12
- thread 4 to CPU 1
- thread 5 to CPU 5
- thread 6 to CPU 9
- thread 7 to CPU 13

On the Linux operating system, information about CPU identifiers is found in /proc/cpuinfo or /sys/devices/system/cpu/ online.

A CPU identifier is an integer value between 0 and (x - 1), in which x is the return value of the function sysconf(_SC_NPROCESSORS_ONLN). This function returns the number of processors available, and it is declared in the header file unistd.h.

To get the CPU identifier that is bound to the current thread, use xlupc_thread_affinity(). When this function is called, it returns the CPU identifier for the current thread. If the current thread is not bound to any processor, this function returns -1. The function xlupc_thread_affinity() is declared in upc.h unless XLUPC_NO_EXT is defined before upc.h is included.

The *runtime_option_name* is not case-sensitive. For example, XLPGASOPTS=nostackcheck is equivalent to XLPGASOPTS=NoStacKcheck.

Options of the XLPGASOPTS environment variable also are specified on the command line with the prefix –xlpgas. For example, XLPGASOPTS=stacksize=30mb is equivalent to ./a.out –xlpgasstacksize=30mb.

For an option specified in the environment variable, you add white spaces before or after the colons (:) or equal signs (=) to improve readability. However, you must enclose the entire option string in quotation marks (“ ”) if *XLPGASOPTS* contains any embedded white spaces. For example, specifying options in any of the following ways has the same effect:

- XLPGASOPTS=stacksize=1 mb:nostackcheck ./a.out
- XLPGASOPTS=” stacksize = 1 mb: nostackcheck “./a.out
- ./a.out -xlpgasstacksize=1 mb -xlpgasnostackcheck

You separate numbers by the underscore sign to improve readability. For example, XLPGASOPTS=stacksize=10000000 is equivalent to LPGASOPTS=stacksize=10_000_000.

Options are processed in order. Options that are specified in the XLPGASOPTS environment variable are processed before the variables specified on the command line. If you specify an option multiple times or specify conflicting options, the last option takes precedence. For example:

```
xlupc -qupc=threads=4 -qupc=stackcheck helloworld.upc

XLPGASOPTS=nostackcheck

./a.out -procs 4 -msg_api ’pgas’ -hostfile hosts –xlpgasstackcheck=80%
```

In this example, XLPGASOPTS=nostackcheck instructs the compiler not to check for stack overflow, but –xlpgasstackcheck=80% specified on the command line takes precedence. You must specify -qupc=stackcheck to instrument the executable file. Otherwise, –xlpgasstackcheck=80% has no effect.

If you specify an unsupported option in the environment variable or on the command line with the –xlpgas prefix, the program issues a warning message and ignores the option.

If you specify an option in an incorrect format, the runtime issues a warning message and uses the default value for the option.

You use the following suffixes to indicate the size in bytes. If you do not use the suffixes, the size is specified by byte. For example, XLPGASOPTS=stacksize=200_000 specifies that the space allocated to the stack used by a thread is at least 200,000 bytes:

- kb (1 kb represents 1024 bytes)
- mb (1 mb represents 1024 kb)
- gb (1 gb represents 1024 mb)
- tb (1 tb represents 1024 gb)

The suffixes are not case-sensitive. The letter b is optional and is omitted in the suffixes. For example, XLPGASOPTS=STACKSIZE=10Mb is equivalent to XLPGASOPTS=stacksize=10m.

You also use the suffixes percent and % to indicate a percentage. For example, the following commands have the same effect:

- **`XLPGASOPTS=stackcheck=20`**
- **`XLPGASOPTS=stackcheck=20%`**
- **`XLPGASOPTS=stackcheck=20percent`**

You specify stack size in hexadecimal format with the prefix 0x. For example, XLPGASOPTS=stacksize=0x800kb is equivalent to XLPGASOPTS=stacksize=2mb.

Compiling and running an example program

This section provides a simple UPC program, the commands to compile and execute the program, and the program output.

In Example 2-8 (`hello.upc`), each thread prints a message to standard output.

Example 2-8 hello.upc

```
# include <upc.h>
# include <stdio.h>
int main()
{
printf("Hello world! (THREAD %d of %d THREADS)\n", MYTHREAD, THREADS);
return 0;
}
```

Use the following command to compile the program in the static environment targeting four threads:

```
xlupc -o hello -qupc=threads=4 hello.upc
```

The compiler compiles the code and generates an executable file, `hello`. To run the executable program, you use the following command:

```
poe ./hello -hostfile hosts -procs 1 -msg_api 'pgas'
```

In this command, **`-procs`** specifies the number of processes to use. The program prints to standard output a message on each thread, as shown in Example 2-9.

Example 2-9 Output of hello.upc

```
Hello world! (THREAD 3 of 4 THREADS)
Hello world! (THREAD 1 of 4 THREADS)
Hello world! (THREAD 0 of 4 THREADS)
Hello world! (THREAD 2 of 4 THREADS)
```

2.3.4 ESSL/PESSL optimized for Power 775 clusters

This section describes ESSL and Parallel ESSL optimized for IBM Power System 775 clusters.

ESSL

The ESSL 5.1 Serial Library and the ESSL SMP Library contain the following subroutines:

- A VSX (SIMD) version of selected subroutines for use on POWER7 processor-based servers
- An AltiVec (SIMD) version of selected subroutines for use on POWER6 processor-based servers

This release of ESSL provides the following changes:

- Operating systems:

  Support is added for the following operating system version:

  - AIX 7.1

  Support is no longer provided for the following operating systems:

  - SUSE Linux Enterprise Server 10 for POWER (SLES10)
  - Red Hat Enterprise Linux 5 (RHEL5)
- Servers and processors:
  - Support is added for the IBM POWER7 processor.
  - Support is no longer provided for the following servers and processors:
    - IBM BladeCenter JS2
    - IBM POWERPC 450
    - IBM POWERPC 450D
    - IBM POWER5
    - IBM POWER5+
    - IBM POWERPC970 processors
    - IBM Blue Gene/P
- Subroutines

  ESSL 5.1 is the last release to support non-LAPACK-conforming subroutines; that is, those ESSL subroutines whose name is the same as an existing LAPACK subroutine, but whose calling-sequence arguments and functionality are different from that LAPACK subroutine.

  The new LAPACK subroutine includes the following value:

  - DSYGVX (Selected Eigenvalues and, optionally, the Eigenvectors of a Positive Definite Real Symmetric Generalized Eigenproblem)
- The following new Fourier Transform subroutines are now included:
  - SRCFTD and DRCFTD (Multidimensional Real-to-Complex Fourier Transform)
  - SCRFTD and DCRFTD (Multidimensional Complex-to-Real Fourier Transform)
  - Fastest Fourier Transform in the West (FFTW) Wrappers

  Support is added to the ESSL FFTW Wrapper Libraries corresponding to the new ESSL Fourier Transform subroutines.

For more information about FFTW Version 3.1.2, see this website:

http://www.fftw.org

Parallel ESSL

Parallel ESSL for AIX, V4.1 supports IBM Power 775 clusters that use the Host Fabric Interface (HFI) and selected stand-alone POWER7 clusters or POWER7clusters that are connected with a LAN supporting IP running AIX 7.1.

This release of Parallel ESSL includes the following changes:

- AIX 7.1 support is added.
- The following new subroutines or subroutine performance enhancements are added:
  - New Dense Linear Algebraic Equation Subroutines:
    - PDTRTRI and PZTRTRI - Triangular Matrix Inverse
  - New Fourier Transform Subroutines:
    - PSCFTD and PDCFTD (Multidimensional Complex Fourier Transforms)
    - PSRCFTD and PDRCFTD (Multidimensional Real-to-Complex Fourier Transforms)
    - PSCRFTD and PDCRFTD (Multidimensional Complex-to-Real Fourier Transforms)
- Support is not provided with Parallel ESSL V4 for the following components:
  - AIX 5.3 or AIX 6.1 operating systems
  - The following servers:
    - BladeCenter JS20 and JS21
    - POWER4
    - POWER 4+
    - POWER5
    - POWER 5+
    - POWER6
    - POWER6+™
    - Power 755
  - Qlogic 9000 Series DDR InfiniBand switches
  - High Performance Switch
  - Myrinet-2000 switch with Myrinet/PCI-X adapters
  - Parallel ESSL GM libraries

Required software products on AIX

Table 2-1 lists the required software products for Parallel ESSL for AIX.

Table 2-1 Required Software Products for Parallel ESSL for AIX

	Required software products	**On AIX Version 7.1**
For compiling	IBM XL Fortran for AIX	13.1.0.1 or later
	IBM XL C/C++ for AIX	11.1.0.1 or later
For linking, loading, or running (For more information, see Note 1)	IBM XL Fortran Run-Time Environment for AIX (For more information, see Note 2)	13.1.0.1 or later
	IBM ESSL for AIX (For more information, see Note 3)	5.1.0.2 or later
	IBM XL C libraries	For more information, see Note 4
	Parallel Environment Runtime Edition for AIX (PE)	1.1 or later

Table notes:

1. Optional file sets are required for building applications. For more information, see the AIX and compiler documentation.
2. The correct version of IBM XL Fortran Run-Time Environment for AIX is automatically shipped with the compiler and is available for downloading from the following website: `http://www.ibm.com/support/docview.wss?rs=43&uid=swg21156900`
3. ESSL for AIX must be ordered separately.
4. The AIX product includes the C and math libraries in the Application Development Toolkit.

2.4 Parallel Environment optimizations for Power 775

The parallel environment optimization for the Power 775 supports the HFI interconnect of the IBM POWER7 processor-based server in User Space. IBM Parallel Environment Runtime Edition 1.1 contains the following functional enhancements:

- The IBM PE Runtime Edition is a new product, with new installation paths and support structures. For more information, see *Parallel Environment Runtime Edition for AIX V1.1: Installation*, SC23-6780.
- Support for launching and managing multiple parallel dynamic subjobs by using a single scheduler or resource management allocation of cluster resources.
- A generic tool that is called Parallel Environment shell (PESH), which resembles a distributed shell with advanced grouping and filtering. The tool is used for executing commands on distributed nodes, and sending commands to, and collecting distributed statistics from, running jobs. PESH also allows PNSD commands to run concurrently on multiple nodes.
- Enhanced scalability such that PE is now designed to run one million tasks per POE job.
- Support for the HFI interconnect of the IBM POWER7 server in User Space.

- Support for the HFI global counter of the IBM POWER7 server, which replaced the global counter of the high performance switch as a time source.
- A new messaging API called PAMI, which replaces the LAPI interface that is used in earlier versions of Parallel Environment. In addition to providing point-to-point messaging support, PAMI provides support for collective communications. This collective support is used by PE MPI when it is appropriate. LAPI is still supported by PE, but its use is deprecated, and users must migrate to PAMI.
- A new run queue-based co-scheduler, in addition to the POE priority adjustment co-scheduler. The run queue-based co-scheduler uses features of AIX 7.1 and the POWER7 architecture to minimize the impact of jitter on user applications.
- Compliance with all requirements of the Message Passing Interface 2.2 standard, including the revisions that are listed in the Annex B Change-Log. The Partition Manager daemon (PMD) is now a setuid (set-userid-on-exec) program, which is owned by only the root user.

This section focus on the enhancement for Power 775 cluster systems that have HFI, CAU, and some run jobs with large numbers of tasks.

2.4.1 Considerations for using HFI

The HFI provides the non-coherent interface between a POWER7 chip quad (QCM: 32-way SMP consisting of four POWER7 chips) and the clustered network. Figure 2-5 on page 120 shows two instances of HFI in a hub chip. The HFIs also attach to the CAU.

Each HFI includes one PowerBus command and four PowerBus data interfaces. The PowerBus directly connects to the processors and memory controllers of four POWER7 chips via the WXYZ links. PowerBus also indirectly coherently connects to the other POWER7 chips within a 256-way drawer via the local links (LL). Although fully supported by the HFI hardware, this path provides reduced performance. Each HFI has four ports to the Integrated Switch Router (ISR). The ISR connects to other Hub chips through the D, LL, and local remote (LR) links. The collection of ISRs and D, LL, and LR links that interconnect then form the cluster nework.

The building block of cluster systems is a set of four POWER7 chips, its associated memory, and a hub chip. The cluster systems consist of multiple building blocks, and are connected to one another via the cluster network.

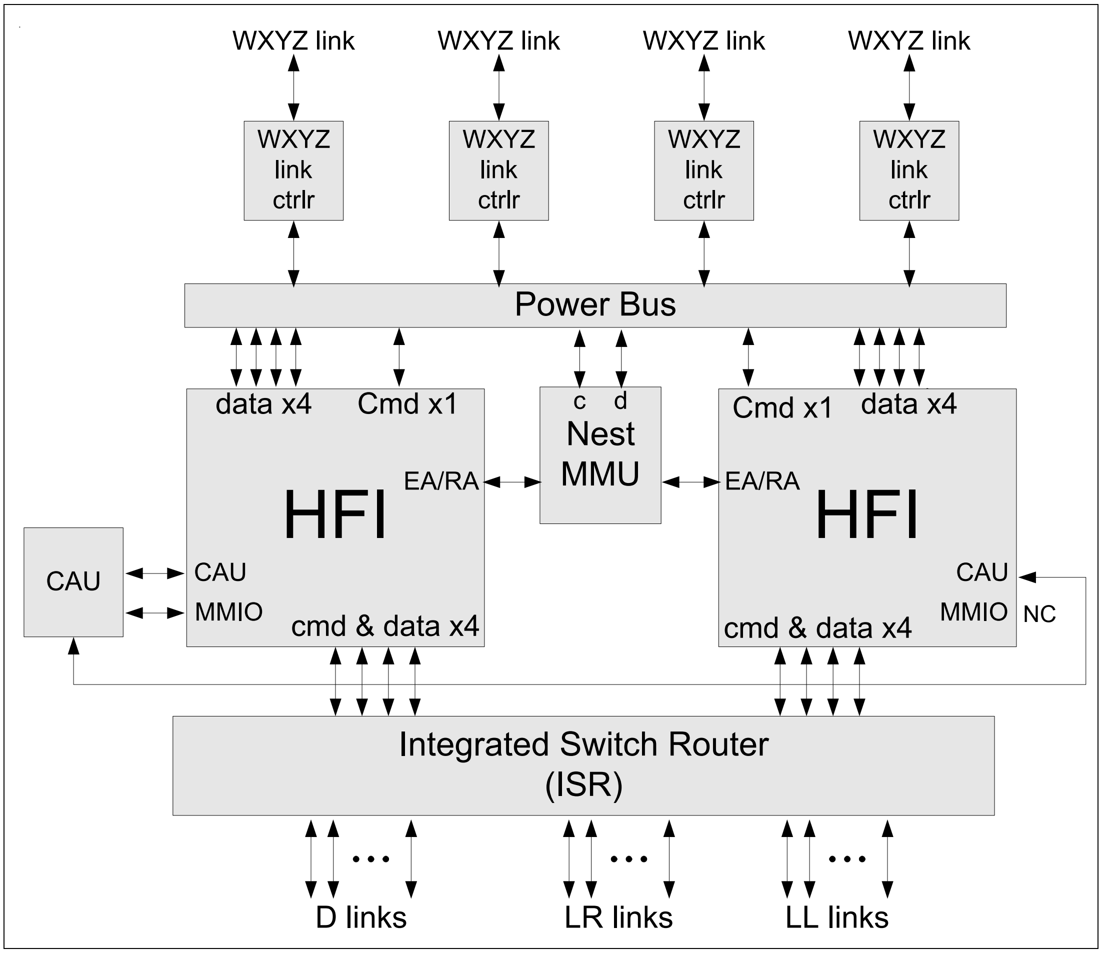


Figure 2-5 HFI attachment to neighboring devices

Packet processing

Serving as the interface between POWER7 chip quads and the cluster network, the HFI moves data between PowerBus and the Integrated Switch Router (ISR). The data is in packets of varying formats, but all packets are processed in the following manner:

- Sending:
  - Pulls or receives data from PowerBus-attached devices in a POWER7 chip
  - Translates data into network packets
  - Injects network packets into the cluster network via the ISR
- Receiving:
  - Receives network packets from the cluster network via the ISR
  - Translates the packets into transactions
  - Pushes the transactions to PowerBus-attached devices in a POWER7 chip
- Packet ordering

  The HFIs and cluster network provide no ordering guarantees among packets. Packets that are sent from the same source window and node to the same destination window and node might reach that destination in a different order.

Figure 2-6 on page 121 shows two HFIs cooperating to move data from devices attached to one PowerBus to devices that are attached to another PowerBus through the cluster network.

Important: The path between any two HFIs might be indirect, requiring multiple hops through intermediate ISRs.

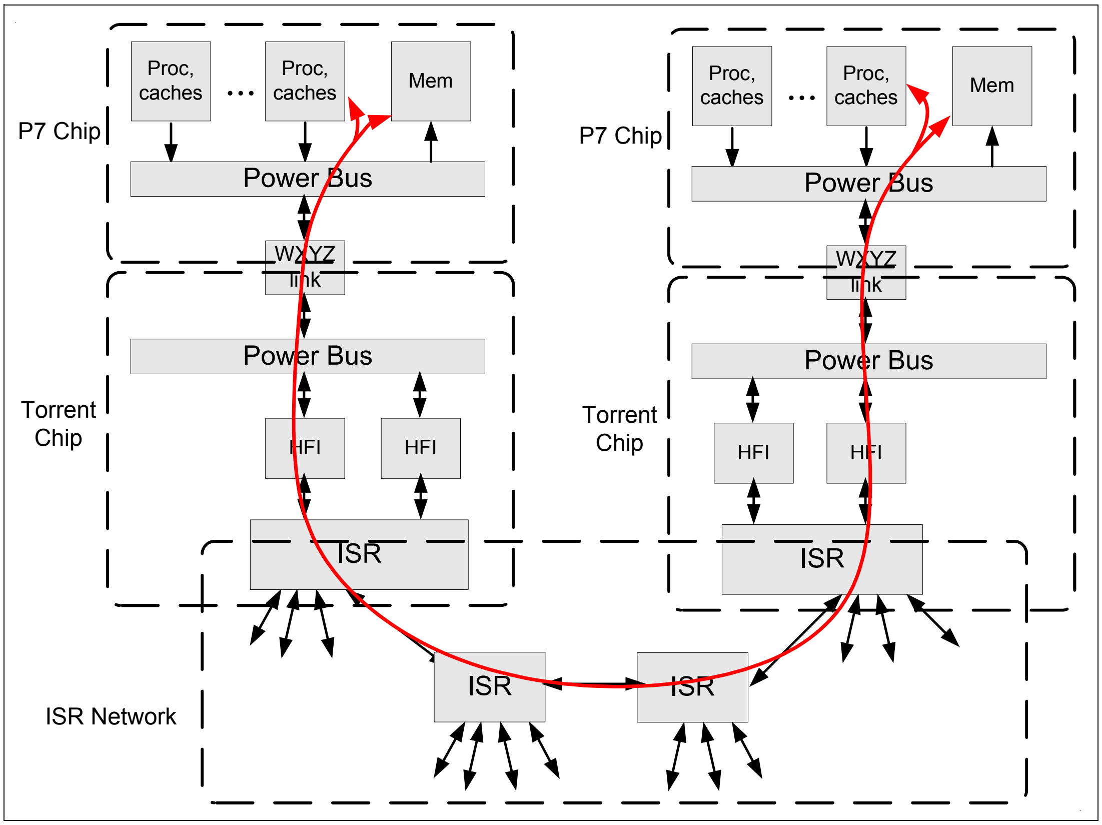


Figure 2-6 HFI moving data from one quad to another

Selecting the HFI route mode

With the HFI, the route of each packet is chosen by the hardware by default, and is usually a direct route to the destination. However, you specify a different route mode that is appropriate for your system by using the MP_FIFO_ROUTE_MODE or MP_RDMA_ROUTE_MODE environment variables (you also can use the -fifo-route_mode or -rdma_route_mode command line flags).

Choosing the appropriate route mode helps you increase outgoing bandwidth and avoid traffic bottlenecks in the network.

The MP_FIFO_ROUTE_MODE and MP_RDMA_ROUTE_MODE route modes include the following attributes:

- Hardware direct (hw_direct)
- Software indirect (sw_indirect)
- Hardware direct striped (hw_direct_striped)
- Hardware indirect (hw_indirect)

You choose to let PAMI/LAPI select the preferred route for the application that is most appropriate, based on the configuration of the system. In particular, the software indirect value allows PAMI/LAPI to select an indirect route to the destination when the origin and destination tasks are on different supernodes.

So that the performance improvements are seen, the application must use at least four striping instances (MP_EUIDEVICE must be set to sn_all and the value that is specified for MP_INSTANCES must greater than four).

Controlling the number of immediate send buffers

By using immediate send rather than FIFO send, you significantly reduce the latency of a send/receive message. You request the number of immediate send buffers to use for your application by using the MP_IMM_SEND_BUFFERS environment variable or the -imm_send_buffers command line flag. You specify any number that is greater than or equal to zero, and less than or equal to the maximum number of available send buffers on the HFI (the default value is 1).

There are 128 immediate send buffers per HFI, and these buffers are shared between windows. If there is only a single LPAR on the octant, there are 128 available hardware buffers. An octant with four LPARs has 32 available hardware buffers. The sum of the buffers that are used by each window might not exceed the total number of buffers that are available on the HFI.

Using RDMA

As shown in Figure 2-7, the Remote Direct Memory Access (RDMA) is a mechanism that allows large contiguous messages to be transferred while the message transfer overhead is reduced. PE support for RDMA differs, depending on the operating system you are using.

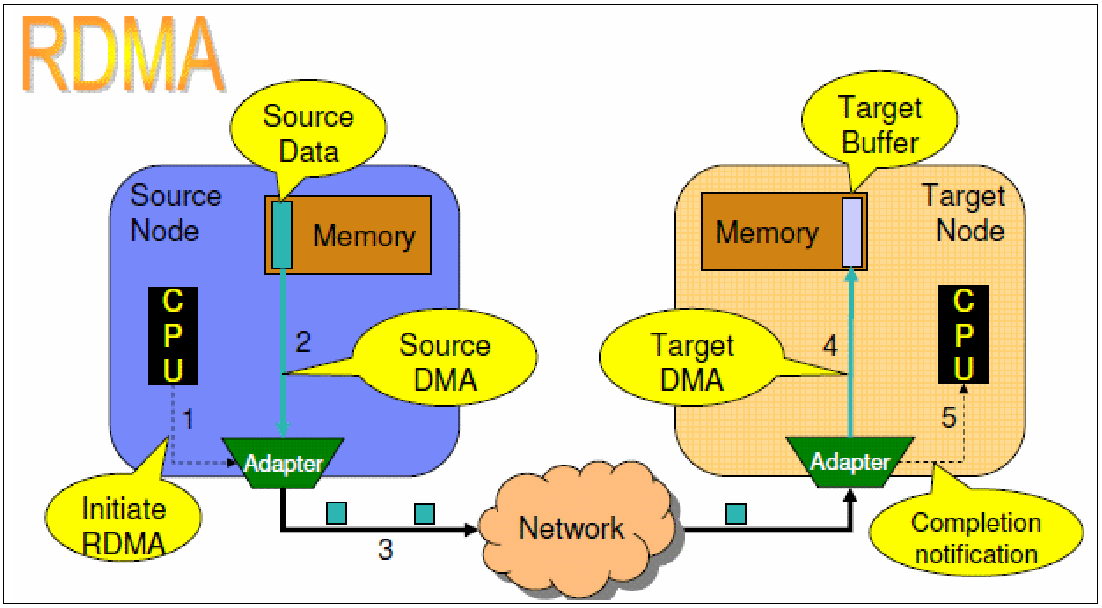


Figure 2-7 Remote Direct Memory Access model

RDMA is supported by the HFI (PE for AIX only).

Using RDMA with the Host Fabric Interface

To use RDMA with the Host Fabric Interface (HFI), you must perform the following tasks:

- Verify that MP_DEVTYPE is set to hfi.
- Request the use of bulk transfer by completing one of the following tasks:
  - Set the MP_USE_BULK_XFER environment variable to yes:

    MP_USE_BULK_XFER=yes

The default setting for MP_USE_BULK_XFER is no.

For batch users, when LoadLeveler is used as the resource manager, setting @bulkxfer results in the setup of the MP_USE_BULK_XFER POE environment variable. Existing users might want to consider removing the @bulkxfer setting from JCF command files and set the MP_USE_BULK_XFER environment variable instead.

- Set the minimum message length for bulk transfer with the MP_BULK_MIN_MSG_SIZE environment variable. Contiguous messages with data lengths greater than the value you specify for this environment variable use the bulk transfer path. Messages that are non-contiguous or have data lengths that are smaller than or equal to the value you specify for this environment variable use the Unreliable Datagram (UD) packet mode method of transfer.
- Set the MP_FORCED_INTERRUPTS_ENABLED environment variable to yes. MP_FORCED_INTERRUPTS_ENABLED allows rendezvous messages to force an interrupt at the target, allowing better overlap of computation and communication for non-blocking requests. By default, MP_FORCED_INTERRUPTS_ENABLED is set to no.

2.4.2 Considerations for data striping with PE

PE MPI depends on PAMI (or LAPI) as a lower-level protocol and the support for striping is entirely within the PAMI/LAPI layer. The layering of PE MPI on PAMI/LAPI is transparent to the MPI user. *Striping* is the distribution of message data across multiple communication adapters to increase bandwidth. By using striping with the bulk transfer transport mechanism, applications experience gains in communication bandwidth performance. Applications that do not use the bulk transfer communication mode often cannot benefit from striping over multiple adapters.

In this case, although the striping implementation is within PAMI/LAPI, it has implications that affect PE MPI users. These instructions are PAMI and LAPI-oriented, but are included here to provide information that you might find valuable. Figure 2-8 on page 124 shows the striping model. For more information about striping, see *Parallel Environment Runtime Edition fo AIX V1.1: LAPI Programming Guide*, SA23-2272 or *Parallel Environment Runtime Edition for AIX V1.1: PAMI Programming Guide*, SA23-2273.

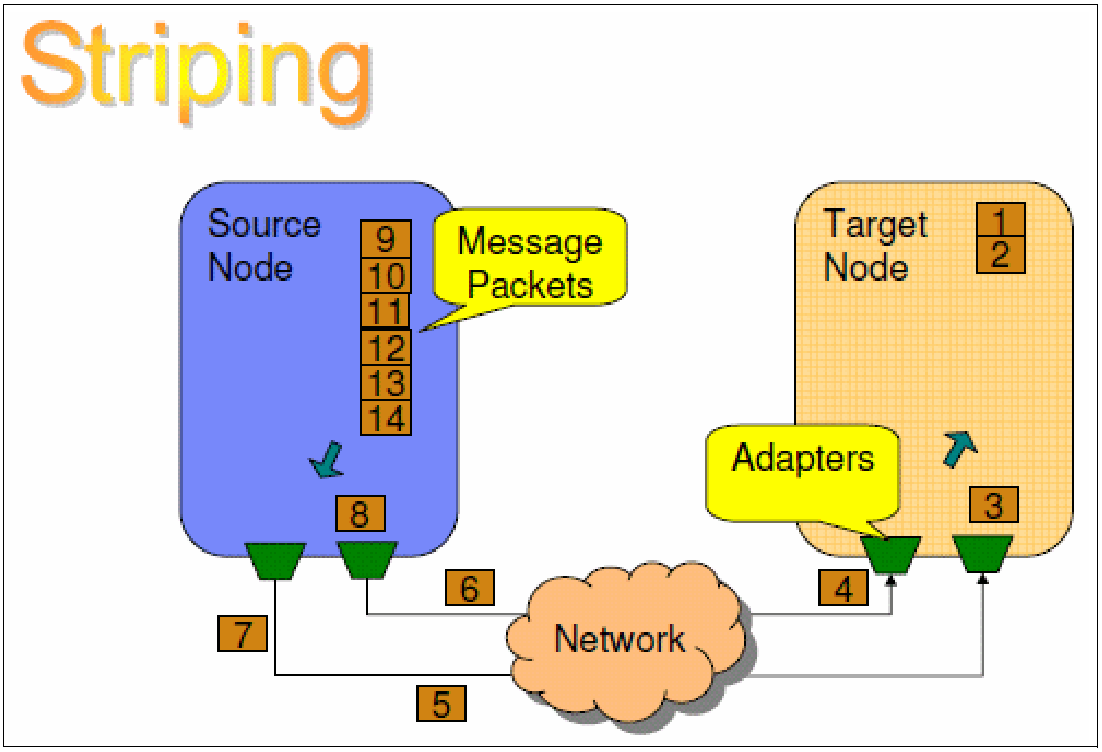


Figure 2-8 Striping

Data striping

When parallel jobs are run, it is possible to stripe data through multiple adapter windows. This striping is supported for both IP and User Space protocols.

If the system has more than one switch network, the resource manager allocates adapter windows from multiple adapters. A switch network is the circuit of adapters that connect to the same interconnect. One window is assigned to an adapter, with one adapter each selected from a different switch network.

If the system has only one switch network, the adapter windows are most likely allocated from different adapters, if there are sufficient windows available on each adapter. If there are not enough windows available on one of the adapters, all of the adapter windows might be allocated from a single adapter.

PAMI (or LAPI) manages communication among multiple adapter windows. By using resources that are allocated by a resource manager, PAMI/LAPI opens multiple user space windows for communication. Every task of the job opens the same number of user space windows, and a particular window on a task communicates only with the corresponding window on other tasks. These windows form a set of virtual networks in which each virtual network consists of a window from each task that communicates with the corresponding windows from the other tasks. The distribution of data among the various windows on a task is referred to as striping, which improves communication bandwidth performance for PAMI/LAPI clients.

To enable striping in user space mode, use environment variable settings that result in the allocation of multiple instances. For a multi-network system, striping is done by setting MP_EUIDEVICE to sn_all. On a single-network system with multiple adapters per operating system image, striping is done by setting MP_EUIDEVICE to sn_single and setting MP_INSTANCES to a value that is greater than 1.

For example, on a node with two adapter links in a configuration in which each link is part of a separate network, the result is a window on each of the two networks that are independent paths from one node to others. For IP communication and for messages that use the user space FIFO mechanism (in which PAMI/LAPI creates packets and copies them to the user space FIFOs for transmission), striping provides no performance improvement. Therefore, PAMI/LAPI does not perform striping for short messages, non-contiguous messages, and all communication in which bulk transfer is disabled through environment variable settings.

For large contiguous messages that use bulk transfer, striping provides a vast improvement in communication performance. Bandwidth scaling is nearly linear with the number of adapters (up to a limit of eight) for sufficiently large messages. This improvement in communication bandwidth stems from the following factors:

- The low overhead that is needed to initiate the remote direct memory access (RDMA) operations that are used to facilitate the bulk transfer.
- The major proportion of RDMA work that is done by the adapters.
- High levels of concurrency in the RDMA operations for various parts of the contiguous messages that are transferred by RDMA by each of the adapters.

To activate striping or failover for an interactive parallel job, you must use the following settings for the MP_EUIDEVICE and MP_INSTANCES environment variables:

- For instances from multiple networks:

  MP_EUIDEVICE=sn_all, which guarantees that the assigned adapters are from different networks.

- For instances from a single network:

  MP_EUIDEVICE=sn_single and MP_INSTANCES=*n* (in which *n* is greater than 1 and less than *max_protocol_instances*), which features improved striping performance by using RDMA that is seen only if windows are allocated from multiple adapters on the single network. Such an allocation might not be possible if there is only one adapter on the network or if there are multiple adapters, but there are resources available on only one of the adapters.

To activate striping for a parallel job that is submitted to the resource manager batch system, the network statement of the resource manager command file must be coded by using the following network statements:

- For a PAMI/LAPI User Space job: #@ network.pami = sn_all,shared,us
- For an MPI and PAMI/LAPI User Space job on multiple networks and shares adapter windows: #@ network.mpi_pami = sn_all,shared,us

The value of MP_INSTANCES ranges from 1 to the maximum value specified by *max_protocol_instances*, as defined for the class in the LoadLeveler `LoadL_admin` file or database configuration. The default value of *max_protocol_instances* is 1. For more information, see *Tivoli® Workload Scheduler LoadLeveler: Using and Administering,* SA22-7881-06.

2.4.3 Confirmation of HFI status

For confirmation of HFI status, the **hfi_read_regs** command is useful. As shown in Example 2-10, the **hfi_read_regs** command shows the status of many types of registers, such as window, non-window, cau, nmmu, and performance.

Example 2-10 hfi_read_regs command usage

```
usage: hfi_read_regs -l hfiX [-Z | {-w win_num | -m | -c | -s cau_slot | -n | -p |
-i}]
       -Z          - Print all registers
       -w win_num  - Print window registers
       -m          - Print non-window registers
       -c          - Print cau registers
       -s cau_slot - Print cau registers related to a specific slot
       -n          - Print nmmu registers
       -p          - Print performance registers
       -i          - Print PHyp internals
```

Example 2-11 shows the hfi_read_regs sample output.

Example 2-11 `hfi_read_regs sample output`

```
# hfi_read_regs -p -l hfi0

Nonwindow Registers
!!!================

   Nonwindow General
      number_of_windows(0:8) . . . . . 0x100 . . . . . .  [256]
      isr_id                           0x0000000000000000 [0]
      rcxt_cache_window_flush_req(0:9) 0x0 . . . . . . .  [0]
         RCWFR.win flush busy(0:0)     0x0                Inactive
         RCWFR.reserved(1:7) . . . . . 0x0 . . . . . . .  [0]
         RCWFR.window id(8:15)         0x0                [0]

   Nonwindow Page Migration
      page_migration_regs[0] . . . . . 0x0000000000000000 [0]
         PMR0.valid(0:0)               0x0                Invalid
         PMR0.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR0.new real addr(18:51)     0x0                [0]
         PMR0.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR0.read target(57:57)       0x0                Old
         PMR0.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[1]           0x0000000000000000 [0]
         PMR1.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR1.reserved1(1:17)          0x0                [0]
         PMR1.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR1.reserved2(52:56)         0x0                [0]
         PMR1.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR1.page size(58:63)         0x0                Reserved
      page_migration_regs[2] . . . . . 0x0000000000000000 [0]
         PMR2.valid(0:0)               0x0                Invalid
         PMR2.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR2.new real addr(18:51)     0x0                [0]
         PMR2.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR2.read target(57:57)         0x0                Old
         PMR2.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[3]           0x0000000000000000 [0]
         PMR3.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR3.reserved1(1:17)          0x0                [0]
         PMR3.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR3.reserved2(52:56)         0x0                [0]
         PMR3.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR3.page size(58:63)         0x0                Reserved
      page_migration_regs[4] . . . . . 0x0000000000000000 [0]
         PMR4.valid(0:0)               0x0                Invalid
         PMR4.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR4.new real addr(18:51)     0x0                [0]
         PMR4.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR4.read target(57:57)       0x0                Old
         PMR4.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[5]           0x0000000000000000 [0]
         PMR5.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR5.reserved1(1:17)          0x0                [0]
         PMR5.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR5.reserved2(52:56)         0x0                [0]
         PMR5.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR5.page size(58:63)         0x0                Reserved
      page_migration_regs[6] . . . . . 0x0000000000000000 [0]
         PMR6.valid(0:0)               0x0                Invalid
         PMR6.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR6.new real addr(18:51)     0x0                [0]
         PMR6.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR6.read target(57:57)       0x0                Old
         PMR6.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[7]           0x0000000000000000 [0]
         PMR7.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR7.reserved1(1:17)          0x0                [0]
         PMR7.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR7.reserved2(52:56)         0x0                [0]
         PMR7.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR7.page size(58:63)         0x0                Reserved

   Nonwindow Page Migration Reservation
      page_migration_reservation[0]. . 0x0000000000000000 [0]
         PMRs0.offset(48:56)           0x0                [0]
         PMRs0.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[1]    0x0000000000000000 [0]
         PMRs1.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs1.reservatn(63:63)        0x0                False
      page_migration_reservation[2]. . 0x0000000000000000 [0]
         PMRs2.offset(48:56)           0x0                [0]
         PMRs2.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[3]    0x0000000000000000 [0]
         PMRs3.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs3.reservatn(63:63)        0x0                False
      page_migration_reservation[4]. . 0x0000000000000000 [0]
         PMRs4.offset(48:56)           0x0                [0]
         PMRs4.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[5]    0x0000000000000000 [0]
```

```
         PMRs5.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs5.reservatn(63:63)        0x0                False
      page_migration_reservation[6]. . 0x0000000000000000 [0]
         PMRs6.offset(48:56)           0x0                [0]
         PMRs6.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[7]    0x0000000000000000 [0]
         PMRs7.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs7.reservatn(63:63)        0x0                False

Performance Counters
:::=================

   Performance Counters: ISR
      cycles blocked sending . . . . . 0x0000000000000000 [0]
      flits sent                       0x0000000000000000 [0]
      flits dropped. . . . . . . . . . 0x0000000000000000 [0]
      link retries                     0x0000000000000000 [0]

   Performance Counters: HFI
      agg pkts sent. . . . . . . . . . 0x000000001bb3f9a8 [464779688]
      agg pkts dropped sendng          0x0000000000000000 [0]
      agg pkts received. . . . . . . . 0x00000000076a4f4f [124407631]
      agg pkts dropped rcving          0x00000000000000af [175]
      agg imm send pkt count . . . . . 0x00000000001ba3f6 [1811446]
      agg send recv pkt count          0x0000000001c83b05 [29899525]
      agg fullRDMA sent count. . . . . 0x00000000188e9613 [411997715]
      agg halfRDMA sent count          0x0000000000000036 [54]
      agg smallRDMA sent count . . . . 0x0000000000000000 [0]
      agg ip pkt sent count            0x0000000000dad28f [14340751]
      agg cau pkt sent count . . . . . 0x0000000000000000 [0]
      agg gups pkt sent count          0x0000000000000000 [0]
      addr xlat wait count . . . . . . 0x000000031589d5d5 [13246256597]
```

As shown in Table 2-2 on page 129, the last part of this output shows “Performance Counters: HFI”. By getting the difference in the numbers in this part between before and after your parallel job is run, you check the amount of traffic that was on the specific HFI.

Often the traffic of IP protocol communication on each HFI is checked by an AIX command, such as **topas** or **nmon**. However, US protocol communication cannot be checked by such AIX commands. In this case, **hfi_read_regs** command is useful. For example, you confirm whether both HFI0 and HFI1 are used as intended in parallel jobs that use US protocol communication. Table 2-2 on page 129 shows some output of Performance Counters for HFIs from **hfi_read_regs** command in running simple pingpong benchmarks that change the number of MP_INSTANCES environment variable. You see that only HFI0 is used in MP_INSTANCES=1 whereas HFI0 and HFI1 are used when MP_INSTANCES is set to more than two. This behavior is reasonable.

Table 2-2 Performance counter numbers for HFIs

		MP_INSTANCES				
		1	2	4	8	16
hfi0	agg pkts sent	1627201	865133	868662	877346	877322
	agg pkts dropped sending	0	0	0	0	0
	agg pkts received	1627181	865097	868630	877309	877294
	agg pkts dropped receiving	0	0	0	0	0
	agg imm send pkt count	24607	12304	12304	12304	12304
	agg send recv pkt count	98409	98398	98394	98395	98391
	agg fullRDMA sent count	1491820	747036	748159	751852	751847
	agg halfRDMA sent count	0	0	0	0	0
	agg smallRDMA sent count	0	0	0	0	0
	agg ip pkt sent count	202	191	197	218	202
	agg cau pkt sent count	0	0	0	0	0
	agg gups pkt sent count	0	0	0	0	0
hfi1	agg pkts sent	31	766560	770090	778757	778746
	agg pkts dropped sending	0	0	0	0	0
	agg pkts received	34	766578	770110	778776	778765
	agg pkts dropped receiving	0	0	0	0	0
	agg imm send pkt count	0	12303	12303	12304	12303
	agg send recv pkt count	0	0	0	0	0
	agg fullRDMA sent count	0	747025	748148	751836	751836
	agg halfRDMA sent count	0	0	0	0	0
	agg smallRDMA sent count	0	0	0	0	0
	agg ip pkt sent count	31	30	33	41	31
	agg cau pkt sent count	0	0	0	0	0
	agg gups pkt sent count	0	0	0	0	0

2.4.4 Considerations for using CAU

The CAU is a unit that accelerates the processing of collectives within a system that is made up of multiple nodes that are interconnected via a cluster network.

Collectives are distributed operations that operate across a tree. The following primary collective operations are used:

- Reduce: Gathers packets from all nodes expect the root node in a tree and reduces the nodes to a single packet that is delivered to the root node. The root node is any node in the tree (not necessarily the node at the top of the tree).

> **Note:** The root node might further reduce the value that it receives from the tree with a local value before processing, but that reduction in value occurs outside the CAU.

- Multi-cast: Broadcasts a packet to the entire tree via a CAU multi-cast packet that begins at any node.

A node is a quad POWER7 chip (forming a 32-way SMP), each main memory of POWER7, and a Torrent chip. Each Torrent chip contains one CAU. The cluster network is a non-coherent communication network among the nodes of a multi-node system. The cluster network is formed by the interconnection of links between the Torrent chips of the system. For more information, see Figure 2-6 on page 121.

When an application selects a set of processors to run a multi-processor program, and these processors are on different (non-coherently connected) nodes, the application forms a tree (subnetwork) for communication and (limited) processing among these processors. A tree is made up of the processors that are used by that program and a set of CAUs (the CAUs might not be on the same nodes as the processors). The CAUs facilitate communication among the processors of a tree and each CAU has an ALU that performs the (limited) processing.

The system supports many trees simultaneously and each CAU supports 64 independent trees. Software defines the tree to the hardware by programming the MMIO registers of each CAU within the tree with the locations (ISR_IDs, and so on) of all of its neighbors. Software also tells each processor within the tree (entirely within software) the location of the CAU to and from which it sends and receives CAU packets.

There are many ways to form a tree that interconnects a set of processors. Figure 2-9 shows an example of a tree which interconnects eight processors (P0 - P7). In this example, a single CAU (C0) is used, which has eight neighbors (maximum is nine).

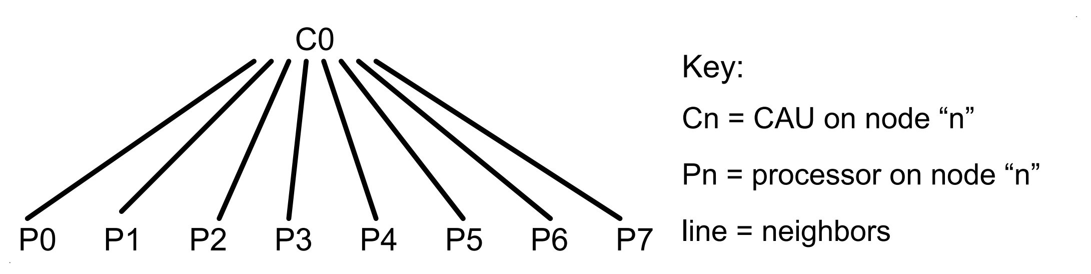


Figure 2-9 CAU example tree 1

Figure 2-10 shows an example of a tree which interconnects eight processors (P0 - P7), but uses two CAUs (C0 and C4), which results in each CAU having five neighbors.

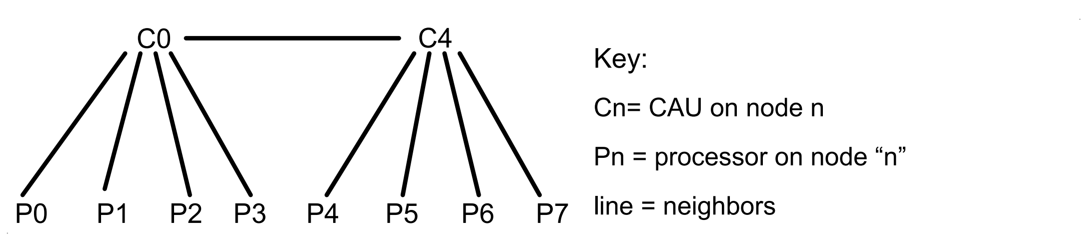


Figure 2-10 CAU example tree 2

Figure 2-11 shows an example of a tree which interconnects eight processors (P0 - P7) by using seven CAUs (C0 - C6), which results in each CAU with three neighbors (C3 includes only two). This configuration is a binary tree because no CAU has more than three neighbors.

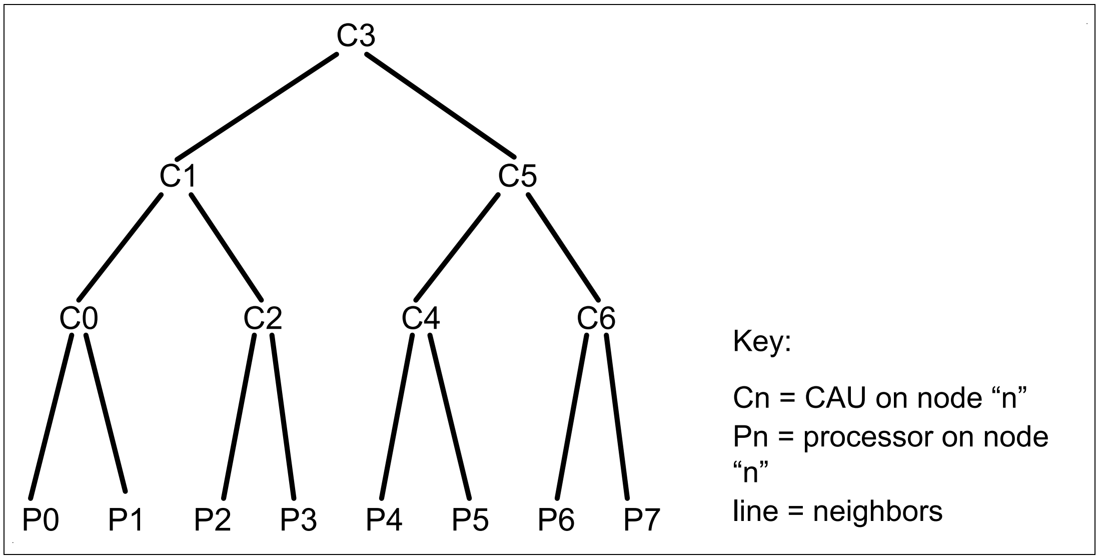


Figure 2-11 CAU example tree 3

Figure 2-12 shows an example of a tree which interconnects eight processors (P0a - P0g, P1) where the first seven processors are on the same node but are associated with different partitions. The eighth processor (P1) is on a different node. All processors use CAU C0, but they are organized into two separate indexes (a and b). This type of tree structure in which one tree features segments at multiple indexes of the same CAU, is useful when a tree includes many neighbors that are different partitions within the same node.

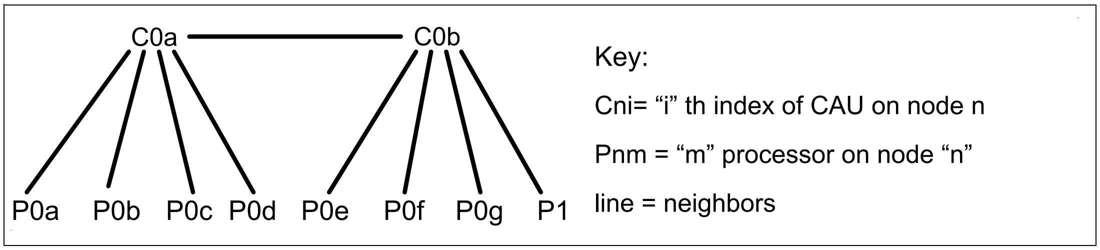

C0a
C0b
P0a
P0b
P0c
P0d
P0e
P0f
P0g
P1
Key:
Cni= “i” th index of CAU on node n
Pnm = “m” processor on node “n”
line = neighbors

Figure 2-12 CAU Example Tree 4

For optimal performance, CAUs that have one or more processors for a neighbor often reside on the same node as one of the processors. CAUs that do not have any processors as a neighbor (such as C1, C3, and C5, as shown in Figure 2-11), are on any node, including a node that does not have a processor that is part of the tree.

> **Important:** In the example trees that are shown in the previous figures, there are more processors than CAUs. Because of this condition, it is possible to define a skinny tree in which multiple CAUs have only two neighbors. This configuration results in more CAUs than processors.

Processors send CAU packets by putting them on the Send FIFO of an HFI. Processors receive CAU packets by pulling them from a Receive FIFO of an HFI. CAUs send and receive CAU packets via an interface to both HFIs on the same node. Figure 2-13 on page 132 shows the physical connections of the CAUs and processors to the HFIs and the Cluster Network. In

Figure 2-13, Node 0 and Node 1 include a CAU and a processor. Node 2 does not have a CAU and node 3 does not have a processor (uncommon).

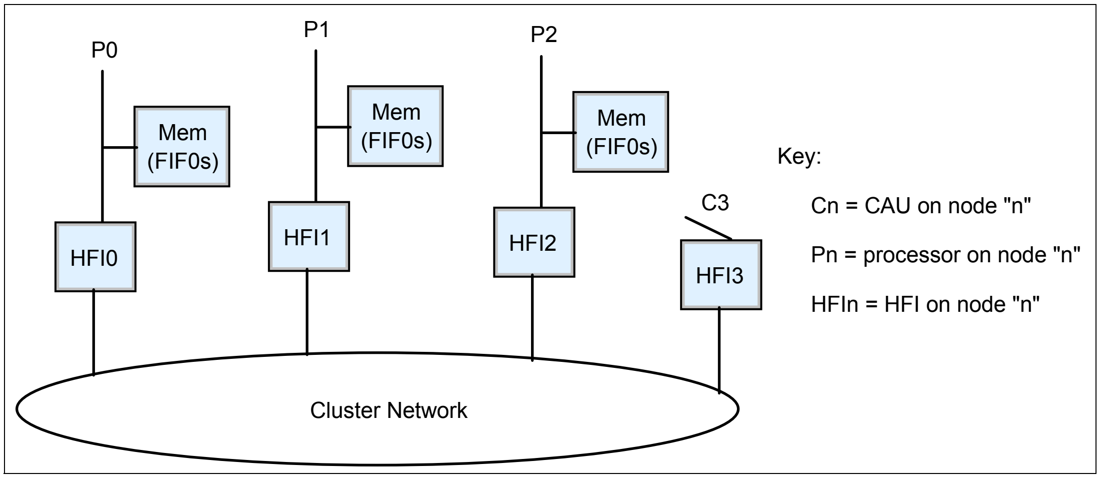


Figure 2-13 CAU tree physical connectivity example

Specifying the number of CAU group

The performance of small broadcast and reduction operations is significantly enhanced with the use of the CAU resources of the HFI. You specify the number of CAU groups for an application by using the MP_COLLECTIVE_GROUPS environment variable or the -collective_groups command line flag. Allowable values include any number that is greater than zero and less than the number of available CAU groups. The default value is zero.

Important: The CAU is not used with operations that are larger than 64 bytes.

2.4.5 Managing jobs with large numbers of tasks

In some situations, you might want to run large jobs that involve large numbers of tasks. PE can support a maximum of 1024 K tasks in a job, with the following exceptions:

- Because of constraints that are imposed by node architecture and interconnect technology, 1024 K task support is limited to 64-bit applications.
- When mixed User Space and shared memory jobs are run, architectural limits for other components (for example, HFI windows) might reduce the number of tasks supported.

To prevent performance degradation, jobs that include large numbers of tasks need special considerations. For example, you must use a different method for specifying the hosts on the host list file. Also, the debuggers and other tools you use must query and receive task information differently, and ensure that they attach to jobs before they start.

Managing task information when large jobs are run

When large jobs of up to 1024 K tasks are run, the amount of information that is generated about the tasks and the operations between them is a large. Writing such a large amount of information to a file degrades performance. To avoid this affect on performance, it is important for a tool (such as a debugger) to minimize the task information that is generated by requesting only the task information it needs. Also, the tool requests that it provide task information to, and receive notifications from, POE by using a socket connection rather than writing the task information to a file.

When a tool or debugger (including PDB) with large-scale jobs is used, it is recommended that you complete the following tasks:

- Ensure that the MP_DBG_TASKINFO environment variable is set to yes. This setting indicates that the debugger exchanges task information with POE by way of a socket connection. MP_DBG_TASKINFO is set to yes by default.
- Ensure that the MP_DEBUG_ATTACH environment variable is set to no. This setting indicates that debugger attachment files are not created or updated, which degrades the performance of a large-scale job.
- Create a socket connection that requests the specific task information that is needed by the tool or debugger, by using the `poe.socket.taskinfo` API. The socket request specifies the following types of task information:
  - Task ID
  - World ID
  - Rank in the world
  - PID of the task
  - Host name
  - IP address
  - Executable path name
  - Node ID (for PDB use only)
  - Session ID (for PDB use only)

For more information about creating the socket request, see to the main page for poe.socket.taskinfo in *Parallel Environment Runtime Edition for AIX V1.1: MPI Programming Guide*, SC23-6783.

When MP_DBG_TASKINFO=yes, POE and PMD each create UNIX socket files whose host names are the same as the names returned by gethostname() system call. By default, these files are written to `/tmp`, but the system administrator optionally uses the MP_DBG_TASKINFO_DIR entry in the `/etc/poe.limits` file to change the directory into which these files are stored.

The tool or debugger connects to the sockets created by POE and the PMD and sends the task information request. POE and PMD respond to the query by sending the requested task information to the tool by way of the same socket.

Specifying hosts for jobs with large numbers of tasks

For jobs that include large numbers of tasks (up to 1024 K), a simple host list file in which each host is specified on a separate line is unworkable. Instead, PE provides a shorthand for specifying the hosts for such large jobs.

Specifying hosts for jobs that include large numbers of tasks requires a different method than the method used for smaller jobs. For smaller jobs, each host is included in the POE host list file on a separate line, but the use of this configuration for large jobs is impractical. To make the host list file usable for large jobs, PE provides a shorthand for specifying the hosts.

On each line of the host list file, you specify a host name, followed by the tasks that run on it. If you do not specify a value for tasks, a default task mapping is assumed. You provide multiple hosts and multiple tasks on a single line. If both hosts and tasks are specified, the number of hosts must match the number of tasks.

The following format of the basic host list file entry is used:

```
hosts%tasks
```

In this format, hosts represents the hosts and tasks represents the tasks that will run on those hosts.

There are several ways to specify the hosts and tasks: as a range, as a stride, as a repetition count, or in free form. Example 2-12 shows some examples.

> **Important:** You need to use the parse_hfile_extension API to parse the shorthand host and task specifications on the host list file. For more information, see the *IBM Parallel Environment Runtime Edition for AIX V1.1: MPI Programming Guide*, SC23-6783.

Example 2-12 Specifying hosts for jobs with large numbers of tasks

```
Use range form

c250f10c12ap05-hf0*8%[0-7]
c250f10c12ap09-hf0*8%[8-15]

This example can be expanded as:

c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0

Use stride form

c250f10c12ap05-hf0*8%[0-14:2]
c250f10c12ap09-hf0*8%[1-15:2]

This example can be expanded as:

c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
```

```
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
```

Use Free form

```
c250f10c12ap05-hf0*6%[0,2,4,6,8,10]
c250f10c12ap09-hf0*6%[1,3,5,7,9,11]
c250f10c12ap13-hf0*4%[12,13,14,15]
```

This example can be expanded as:

```
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
```

Advanced form

```
c250f10c12ap[05-13:4]-hf0*5
```

This example can be expanded as:

```
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap05-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap09-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
c250f10c12ap13-hf0
```

2.5 IBM Parallel Environment Developer Edition for AIX

The new IBM Parallel Environment Developer Edition for AIX includes the Eclipse Parallel Tools Platform (PTP 5.0), and the IBM High Performance Computing Toolkit (IBM HPC Toolkit). These tools are described next.

2.5.1 Eclipse Parallel Tools Platform

PTP 5.0 provides an integrated Eclipse development environment for parallel application developers. This environment allows developers to perform the following tasks in the parallel application development cycle from within their Eclipse environment, for C, C++, and Fortran, interacting transparently with the HPC cluster throughout the development cycle:

- Create projects to contain application source code and associated files. All application files reside on the remote cluster and are transparently synchronized between a remote HPC cluster (or clusters) and the local Eclipse environment of the developer.
- Edit application source files transparently in the Eclipse environment of the developer.
- Use code completion features and interactive help to simplify coding of API calls for MPI, LAPI, PAMI, UPC, OpenMP, OpenACC, and OpenSHMEM.
- Use online help for MPI, LAPI, PAMI, UPC, OpenMP, OpenACC, and OpenSHMEM.
- Analyze source code to locate application artifacts (variables and function calls) for MPI and PAMI functions.
- Run source code analysis to find parallel application coding problems, such as mismatched MPI barriers.
- Compile the application.
- Run the application, choosing resource managers to run the application in Parallel Environment, LoadLeveler, OpenMPI, MPICH, and so on.
- Debug the application.
- Run performance tools, such as IBM HPC Toolkit, to analyze performance of the parallel application.

For more information about the Eclipse Parallel Tools Platform, getting started, tutorials, and help, see this website:

http://eclipse.org/ptp/doc.php

2.5.2 IBM High Performance Computing Toolkit

The IBM HPC Toolkit is a set of tools that is used to gather performance measurements for the application and to help users find potential performance problems in the application. The IBM HPC Toolkit includes an Eclipse plug-in that helps you instrument and run an application and view the performance measurement data for hardware performance counters, MPI profiling, OpenMP profiling, and application I/O profiling.

You also sort and filter performance data to help better understand the performance of the application. The IBM HPC Toolkit also includes the peekperf GUI with which you instrument and run the application. You also view the performance measurement data for hardware performance counters, MPI profiling, OpenMP profiling, and application I/O profiling, all from within peekperf. You also sort and filter performance data within peekperf to help better understand the performance of the application.

The IBM HPC Toolkit also includes the Xprof GUI, which is a viewer for `gmon.out` files that are generated by compiling the application by using the **-pg** option. Xprof is used to find hot spots in your application.

The following installation location is used for AIX and Linux (PE Developer Edition):

```
/opt/ibmhpc/ppedev.pct
```

You must ensure that several environment variables that are required by the IBM HPC Toolkit are properly set before you use the toolkit. To set these environment variables, you run the setup scripts that are in the top-level directory of your IBM HPC Toolkit installation.

Before you use the IBM HPC Toolkit, issue the following commands:

- **cd /opt/ibmhpc/ppedev.pct**
- **. ./env_sh**

These commands add a path for the HPC toolkit command to the PATH environment variable and the path for HPC toolkit library to the LIBPATH environment variable for AIX and the LD_LIBRARY_PATH environment variable for Linux.

If you use the Eclipse plug-in, you do not set these environment variables directly. Instead, you set them by using Eclipse dialogs. The IBM HPC Toolkit requires your application to be compiled and linked by using the **–g** flag.

If your application is not compiled and linked by using the **–g** flag, peekperf and hpctInst cannot instrument your application. If you plan to use Xprof to view profiling data for your application, your application must be compiled and linked by using the **–pg** flag.

If you use the IBM HPC Toolkit on Linux to collect performance data other than application profiling by using Xprof, you must compile and link the application with the —emit-stub-syms and -WI, —hash-style=sysv flags.

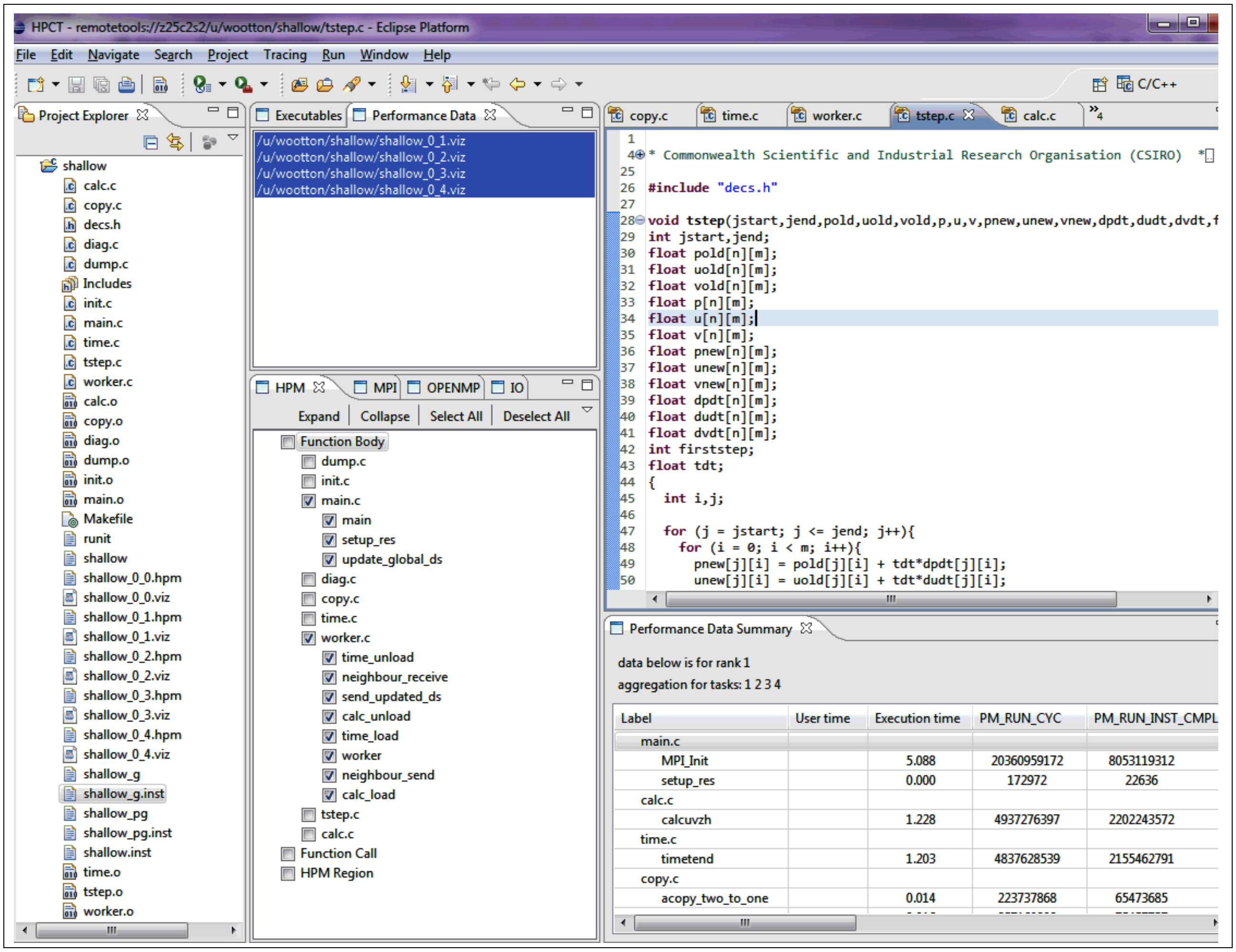
Using the IBM HPC Toolkit Eclipse plug-in

This Eclipse plug-in is a user interface for the IBM HPC Toolkit that you use for instrumenting your application, running your instrumented application, and obtaining performance measurements in the following areas:

- Hardware performance counter measurements
- MPI profiling
- OpenMP profiling
- Application I/O profiling

The plug-in maps performance measurements to your source code, and allows you to sort and filter the data.

Figure 2-14 on page 138 shows the Eclipse HPCT perspective for the IBM HPC Toolkit.



```
* Commonwealth Scientific and Industrial Research Organisation (CSIRO)  *

#include "decs.h"

void tstep(jstart,jend,pold,uold,vold,p,u,v,pnew,unew,vnew,dpdt,dudt,dvdt,f
int jstart,jend;
float pold[n][m];
float uold[n][m];
float vold[n][m];
float p[n][m];
float u[n][m];
float v[n][m];
float pnew[n][m];
float unew[n][m];
float vnew[n][m];
float dpdt[n][m];
float dudt[n][m];
float dvdt[n][m];
int firststep;
float tdt;
{
  int i,j;

  for (j = jstart; j <= jend; j++){
    for (i = 0; i < m; i++){
      pnew[j][i] = pold[j][i] + tdt*dpdt[j][i];
      unew[j][i] = uold[j][i] + tdt*dudt[j][i];
```

data below is for rank 1

aggregation for tasks: 1 2 3 4

Label	User time	Execution time	PM_RUN_CYC	PM_RUN_INST_CMPL
main.c				
MPI_Init		5.088	20360959172	8053119312
setup_res		0.000	172972	22636
calc.c				
calcuvzh		1.228	4937276397	2202243572
time.c				
timetend		1.203	4837628539	2155462791
copy.c				
acopy_two_to_one		0.014	223737868	65473685

Figure 2-14 Eclipse Open Performance Data View

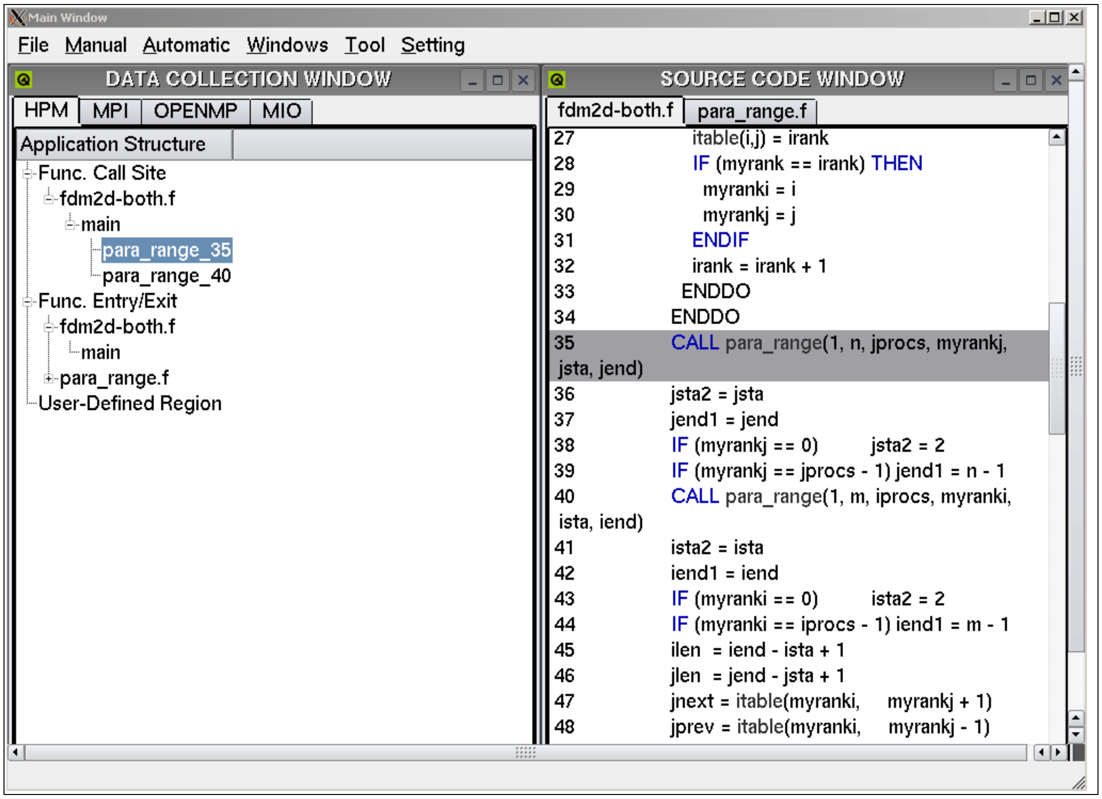
Using the Peekperf GUI

The peekperf GUI is another user interface for the IBM HPC Toolkit. You use the GUI for instrumenting your application, running your instrumented application, and obtaining performance measurements in the following areas:

- Hardware performance counter measurements
- MPI profiling
- OpenMP profiling
- Application I/O profiling

The peekperf GUI maps performance measurements to your source code, and allows you to sort and filter the data. Peekperf is started by issuing the command **`peekperf`**. When you issue the **`peekperf`** command, the peekperf GUI main window is displayed. When the main window opens, you load an application executable, open visualization data files that contain your performance measurements, and open application source files.

Figure 2-15 on page 139 shows the peekperf data collection window with expanded application structure after an application executable is loaded.

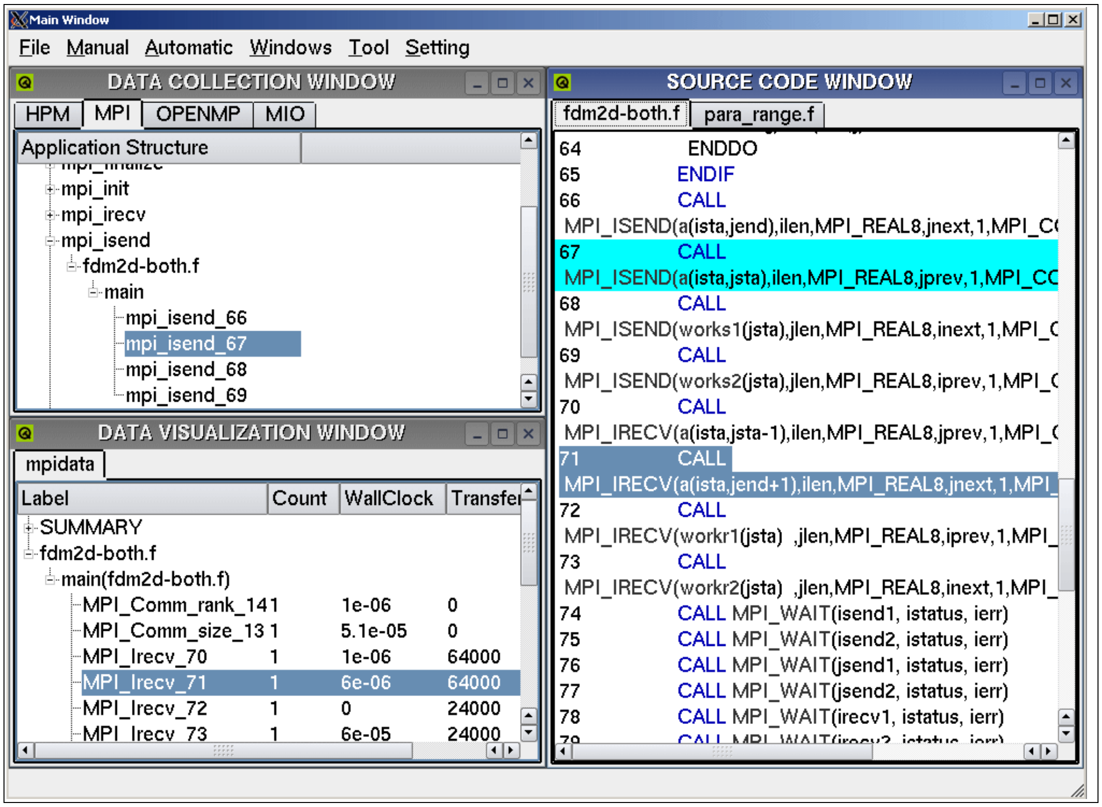


```
          itable(i,j) = irank
          IF (myrank == irank) THEN
            myranki = i
            myrankj = j
          ENDIF
          irank = irank + 1
        ENDDO
      ENDDO
      CALL para_range(1, n, jprocs, myrankj,
 jsta, jend)
      jsta2 = jsta
      jend1 = jend
      IF (myrankj == 0)        jsta2 = 2
      IF (myrankj == jprocs - 1) jend1 = n - 1
      CALL para_range(1, m, iprocs, myranki,
 ista, iend)
      ista2 = ista
      iend1 = iend
      IF (myranki == 0)        ista2 = 2
      IF (myranki == iprocs - 1) iend1 = m - 1
      ilen  = iend - ista + 1
      jlen  = jend - jsta + 1
      jnext = itable(myranki,    myrankj + 1)
      jprev = itable(myranki,    myrankj - 1)
```

Figure 2-15 Peekperf data collection window with expanded application structure

The tree in the data collection window panel presents the program structure and is created based on the type of performance data. For example, the tree in the HPM panel contains two subtrees: the Func. Entry/Exit subtree shows all the functions in the application, and the Func. Call Site subtree shows all of the call sites for each function.

Figure 2-16 on page 140 shows Peekperf MPI profiling data collection window. The data visualization window shows the number of times each function call was executed, the total time spent executing that function call, and the amount of data that is transferred by that function call. You see a more detailed view of the data you obtained by right-clicking over a leaf node in the data visualization window, which opens a metrics browser window, or by right-clicking in the white space in the data visualization window and selecting Show as Table from the pop-up menu, which opens a table view of the performance data.



```
mpi_init
mpi_irecv
mpi_isend
  fdm2d-both.f
    main
      mpi_isend_66
      mpi_isend_67
      mpi_isend_68
      mpi_isend_69
```

Label	Count	WallClock	Transfer
SUMMARY			
fdm2d-both.f			
main(fdm2d-both.f)			
MPI_Comm_rank_14	1	1e-06	0
MPI_Comm_size_13	1	5.1e-05	0
MPI_Irecv_70	1	1e-06	64000
MPI_Irecv_71	1	6e-06	64000
MPI_Irecv_72	1	0	24000
MPI_Irecv_73	1	6e-05	24000

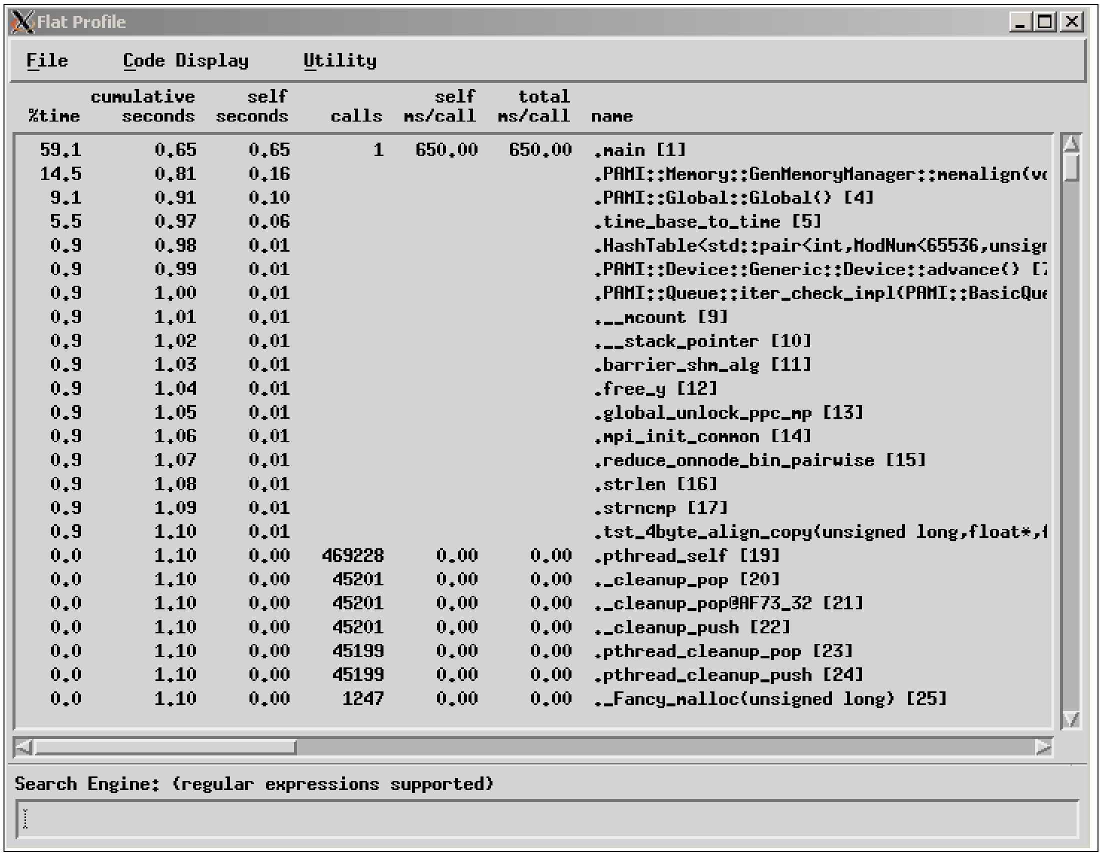
```
64          ENDDO
65        ENDIF
66        CALL
 MPI_ISEND(a(ista,jend),ilen,MPI_REAL8,jnext,1,MPI_C
67        CALL
 MPI_ISEND(a(ista,jsta),ilen,MPI_REAL8,jprev,1,MPI_CC
68        CALL
 MPI_ISEND(works1(jsta),jlen,MPI_REAL8,inext,1,MPI_C
69        CALL
 MPI_ISEND(works2(jsta),jlen,MPI_REAL8,iprev,1,MPI_C
70        CALL
 MPI_IRECV(a(ista,jsta-1),ilen,MPI_REAL8,jprev,1,MPI_C
71        CALL
 MPI_IRECV(a(ista,jend+1),ilen,MPI_REAL8,jnext,1,MPI_
72        CALL
 MPI_IRECV(workr1(jsta) ,jlen,MPI_REAL8,iprev,1,MPI_
73        CALL
 MPI_IRECV(workr2(jsta) ,jlen,MPI_REAL8,inext,1,MPI_
74        CALL MPI_WAIT(isend1, istatus, ierr)
75        CALL MPI_WAIT(isend2, istatus, ierr)
76        CALL MPI_WAIT(jsend1, istatus, ierr)
77        CALL MPI_WAIT(jsend2, istatus, ierr)
78        CALL MPI_WAIT(irecv1, istatus, ierr)
```

Figure 2-16 Peekperf MPI profiling data collection window

X Windows System Performance Profiler

The X Windows System Performance Profiler (Xprof) tool helps you analyze the performance of your parallel or serial application. The tool uses procedure-profiling information to construct a graphical display of the functions within your application. Xprof provides quick access to the profiled data, which helps you identify the functions that are the most CPU-intensive. By using the GUI, you also manipulate the display to focus on the critical areas of the application.

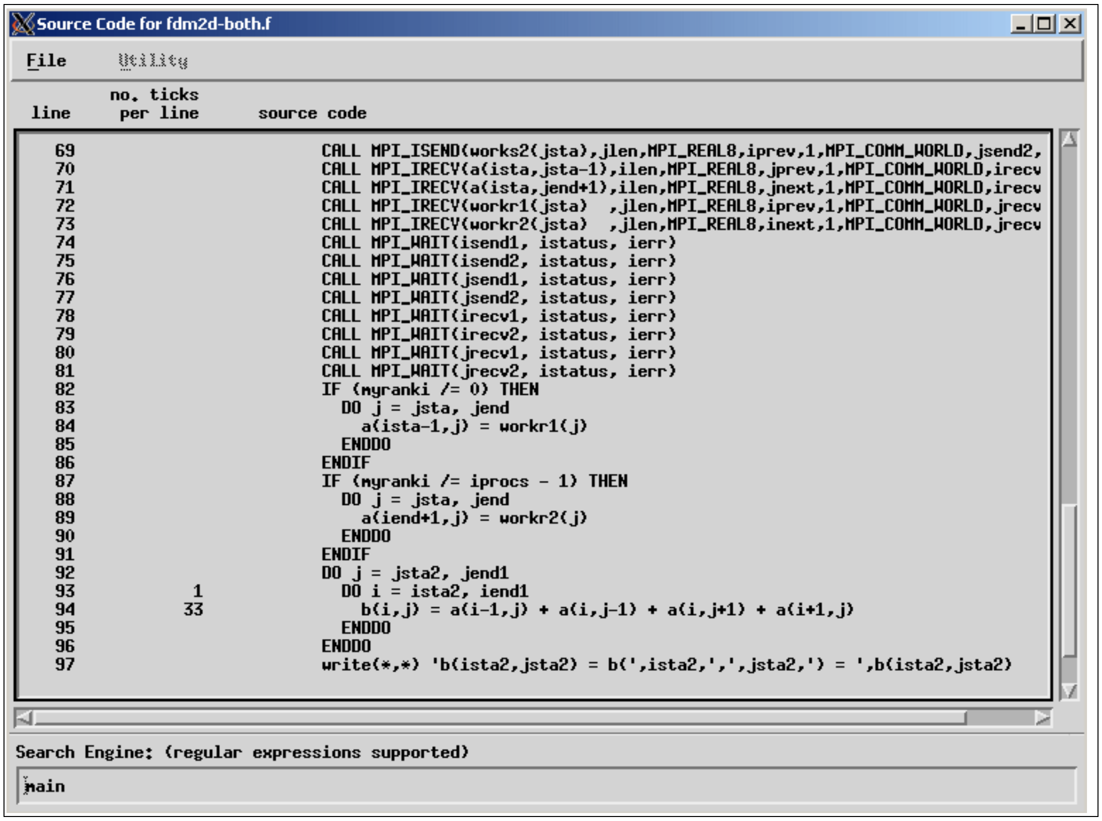
When you select the Flat Profile menu option, the Flat Profile window appears, as shown in Figure 2-17 on page 141. The Flat Profile report shows you the total execution times and call counts for each function (including shared library calls) within your application. The entries for the functions that use the greatest percentage of the total CPU usage appear at the top of the list that is based on the amount of time used.



```
       cumulative     self              self     total
 %time    seconds  seconds     calls  ms/call  ms/call  name
  59.1       0.65     0.65         1   650.00   650.00  .main [1]
  14.5       0.81     0.16                              .PAMI::Memory::GenMemoryManager::memalign(v
   9.1       0.91     0.10                              .PAMI::Global::Global() [4]
   5.5       0.97     0.06                              .time_base_to_time [5]
   0.9       0.98     0.01                              .HashTable<std::pair<int,ModNum<65536,unsign
   0.9       0.99     0.01                              .PAMI::Device::Generic::Device::advance() [7
   0.9       1.00     0.01                              .PAMI::Queue::iter_check_impl(PAMI::BasicQue
   0.9       1.01     0.01                              .__mcount [9]
   0.9       1.02     0.01                              .__stack_pointer [10]
   0.9       1.03     0.01                              .barrier_shm_alg [11]
   0.9       1.04     0.01                              .free_y [12]
   0.9       1.05     0.01                              .global_unlock_ppc_mp [13]
   0.9       1.06     0.01                              .mpi_init_common [14]
   0.9       1.07     0.01                              .reduce_onnode_bin_pairwise [15]
   0.9       1.08     0.01                              .strlen [16]
   0.9       1.09     0.01                              .strncmp [17]
   0.9       1.10     0.01                              .tst_4byte_align_copy(unsigned long,float*,f
   0.0       1.10     0.00    469228     0.00     0.00  .pthread_self [19]
   0.0       1.10     0.00     45201     0.00     0.00  ._cleanup_pop [20]
   0.0       1.10     0.00     45201     0.00     0.00  ._cleanup_pop@AF73_32 [21]
   0.0       1.10     0.00     45201     0.00     0.00  ._cleanup_push [22]
   0.0       1.10     0.00     45199     0.00     0.00  .pthread_cleanup_pop [23]
   0.0       1.10     0.00     45199     0.00     0.00  .pthread_cleanup_push [24]
   0.0       1.10     0.00      1247     0.00     0.00  ._Fancy_malloc(unsigned long) [25]
```

Figure 2-17 Xprof Flat Profile window

As shown in Figure 2-18 on page 142, the source code window shows you the source code file for the function you specified from the Flat Profile window.



Source Code for fdm2d-both.f
File
Utility
Search Engine: (regular expressions supported)
main

```
        no. ticks
 line    per line      source code

   69                         CALL MPI_ISEND(works2(jsta),jlen,MPI_REAL8,iprev,1,MPI_COMM_WORLD,jsend2,
   70                         CALL MPI_IRECV(a(ista,jsta-1),ilen,MPI_REAL8,jprev,1,MPI_COMM_WORLD,irecv
   71                         CALL MPI_IRECV(a(ista,jend+1),ilen,MPI_REAL8,jnext,1,MPI_COMM_WORLD,irecv
   72                         CALL MPI_IRECV(workr1(jsta)  ,jlen,MPI_REAL8,iprev,1,MPI_COMM_WORLD,jrecv
   73                         CALL MPI_IRECV(workr2(jsta)  ,jlen,MPI_REAL8,inext,1,MPI_COMM_WORLD,jrecv
   74                         CALL MPI_WAIT(isend1, istatus, ierr)
   75                         CALL MPI_WAIT(isend2, istatus, ierr)
   76                         CALL MPI_WAIT(jsend1, istatus, ierr)
   77                         CALL MPI_WAIT(jsend2, istatus, ierr)
   78                         CALL MPI_WAIT(irecv1, istatus, ierr)
   79                         CALL MPI_WAIT(irecv2, istatus, ierr)
   80                         CALL MPI_WAIT(jrecv1, istatus, ierr)
   81                         CALL MPI_WAIT(jrecv2, istatus, ierr)
   82                         IF (myranki /= 0) THEN
   83                           DO j = jsta, jend
   84                             a(ista-1,j) = workr1(j)
   85                           ENDDO
   86                         ENDIF
   87                         IF (myranki /= iprocs - 1) THEN
   88                           DO j = jsta, jend
   89                             a(iend+1,j) = workr2(j)
   90                           ENDDO
   91                         ENDIF
   92                         DO j = jsta2, jend1
   93            1              DO i = ista2, iend1
   94           33                b(i,j) = a(i-1,j) + a(i,j-1) + a(i,j+1) + a(i+1,j)
   95                           ENDDO
   96                         ENDDO
   97                         write(*,*) 'b(ista2,jsta2) = b(',ista2,',',jsta2,') = ',b(ista2,jsta2)
```

Figure 2-18 Xprof source code window

The source code window contains information for source code line number and number of ticks per line. Each tick represents 0.01 seconds of CPU time that is used and the value of each line represents the number of ticks that is used by the corresponding line of code. This information is used to find hot spots of source code and to optimize the applicator.

Using the hpccount command

The **hpccount** command is a command line tool included in the IBM HPC Toolkit that is used in the same manner as the time command. Performance counter data is provided to characterize the performance of the application on POWER7 hardware with the intent of finding the bottlenecks of performance and looking for opportunities to optimize the target application code. By using **hpccount**, you monitor the activity of all POWER7 subsystems, including the measurements for cache misses at all levels of cache, the number of floating point instructions that are executed, the number of load instructions that result in TLB misses, and other measurements that are supported by hardware.

The following format of **hpccount** is used most often:

```
hpccount <program>
```

If you use the **hpccount** command to measure the performance of your parallel program, you must invoke **hpccount** with the following format:

```
poe hpccount <program>
```

If you invoke **hpccount** as **hpccount poe <program>**, the performance data is shown for the **poe** command, not the performance data for your parallel program.

If the command executed correctly, **hpccount** gathers various performance data and shows them at the end of the window or stdout. In particular, resource usage statistics, hardware performance counter information, and derived hardware metrics are shown.

Example 2-13 shows the output of **hpccount** command and as it is displayed at the end of stdout.

Example 2-13 output of hpccount

```
hpccount (IBM HPC Toolkit for PE Developer Edition) summary

 ########  Resource Usage Statistics  ########

 Total amount of time in user mode             : 17.573938 seconds
 Total amount of time in system mode           : 0.037333 seconds
 Maximum resident set size                     : 212472 Kbytes
 Average shared memory use in text segment     : 2049 Kbytes*sec
 Average unshared memory use in data segment   : 4226107 Kbytes*sec
 Number of page faults without I/O activity    : 63651
 Number of page faults with I/O activity       : 20
 Number of times process was swapped out       : 0
 Number of times file system performed INPUT   : 0
 Number of times file system performed OUTPUT  : 0
 Number of IPC messages sent                   : 0
 Number of IPC messages received               : 0
 Number of signals delivered                   : 0
 Number of voluntary context switches          : 24
 Number of involuntary context switches        : 302

 #######  End of Resource Statistics  ########

 Execution time (wall clock time)      : 20.3509491559817 seconds

 PM_VSU_FSQRT_FDIV (four flops operation (fdiv,fsqrt) Scalar Instructions only!)
:          395825
 PM_VSU_FIN (VSU0 Finished an instruction)
:     27293671118
  PM_VSU_FMA (two flops operation (fmadd, fnmadd, fmsub, fnmsub) Scalar
instructions only!)                  :     15204099892
  PM_VSU_1FLOP (one flop (fadd, fmul, fsub, fcmp, fsel, fabs, fnabs, fres, fsqrte,
fneg) operation finished) :      7774667179
 PM_RUN_INST_CMPL (Run_Instructions)
:     42708056103
 PM_RUN_CYC (Run_cycles)
:     76027274960

  Utilization rate                                :          97.388 %
  Instructions per run cycle                      :           0.562
```

For more information: For more information about the IBM HPC Toolkit, see *IBM High Performance Computing Toolkit: Installation and Usage Guide* (IBM HPC Toolkit is now a part of the IBM PE Developer Edition) at this website:

http://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+High+Performance+Computing+Toolkit

2.6 Running workloads by using IBM LoadLeveler

LoadLeveler manages both serial and parallel jobs, including OpenMP, MPI, and Hybrid (MPI + OpenMP) application on a Power 775 system. The job is submitted to LoadLeveler cluster through the job command file (JCF), and each job is allocated available resources in the Power 775 cluster. It is necessary to compose the job command file at the beginning of the process to run the job in the LoadLeveler cluster.

2.6.1 Submitting jobs

You describe the job you want to submit and run within the job command file. The job command file includes some of the LoadLeveler keyword statements.

For example, to specify a binary that is executed, you use the executable keyword. To specify a shell script that is executed, the executable keyword also is used; however, if the keyword is not used, LoadLeveler assumes that the job command file is the executable.

As shown in Example 2-14, the **llclass** command is used to ascertain class information about the current LoadLeveler cluster. Example 2-14 shows that the Y_Class and the X_Class classes are available in the LoadLeveler cluster. The available class name is used to compose a LoadLeveler Job Command File (JCF) with the class keyword.

Example 2-14 llclass command

```
$ llclass
Name                 MaxJobCPU     MaxProcCPU  Free   Max Description
                    d+hh:mm:ss     d+hh:mm:ss Slots Slots
--------------- -------------- -------------- ----- ----- ---------------------
Y_Class              undefined      undefined   190   190
X_Class              undefined      undefined   190   190
```

Job Command File for a serial job

Figure 2-19 is a sample job command file for a serial job that shows that jcf keyword #@rset_support=rset_mcm_affinity is used instead of export MEMORY_AFFINITY=MCM.

```
#!/bin/ksh
# @ job_name = myjob.serial
# @ job_type = serial
# @ class = X_Class
# @ resources = ConsumableCpus(1)
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ queue
export MEMORY_AFFINITY=MCM
./serial.exe
```

Figure 2-19 Job Command File for serial job

Job Command File for OpenMP job

Figure 2-20 shows a sample job command file for an OpenMP job. This job is requesting four separate cores on four CPUs. You set the number of threads for the OpenMP job by using parallel_threads and OMP_NUM_THREADS environment variable.

```
#!/bin/ksh
# @ job_name = myjob.openmp
# @ job_type = parallel
# @ class = X_Class
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ task_affinity = core(4)
# @ cpus_per_core = 1
# @ parallel_threads = 4
# @ queue
export OMP_NUM_THREADS=4
export MEMORY_AFFINITY=MCM
./openmp.exe
```

Figure 2-20 Job Command File for OpenMP job

Job Command File for MPI job

Figure 2-21 shows a sample job command file for an MPI job. This job is requesting four CPUs on four separate cores. The export of OMP_NUM_THREADS and MEMORY_AFFINITY is not needed.

```
#!/bin/ksh
# @ job_name = myjob.mpi
# @ job_type = parallel
# @ class = X_Class
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ env_copy = all
# @ bulkxfer = yes
# @ network.MPI = sn_all,shared,US,,4
# @ rset = RSET_MCM_AFFINITY
# @ mcm_affinity_options = mcm_mem_req mcm_distribute mcm_sni_none
# @ task_affinity=core
# @ cpus_per_core=1
# @ node = 2
# @ tasks_per_node = 32
# @ collective_groups = 64
# @ queue
export MP_LABELIO=yes
export LANG=en_US
export MP_SHARED_MEMORY=yes
export MP_SINGLE_THREAD=yes
export MP_DEVTYPE=hfi
export MP_USE_BULK_XFER=yes
export MP_RDMA_ROUTE_MODE="hw_indirect"
export MP_FIFO_ROUTE_MODE="hw_indirect"
poe ./mpi.exe
```

Figure 2-21 Job Command File for an MPI job

Job Command File for Hybrid (MPI + OpenMP) job

Figure 2-22 shows a sample job command file for a Hybrid (MPI + OpenMP) job. This job is requesting two nodes, eight tasks per node and four threads per task. The total requested cores are 64.

```
#!/bin/ksh
# @ job_name = myjob.hybrid
# @ job_type = parallel
# @ class = X_Class
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ env_copy = all
# @ bulkxfer = yes
# @ network.MPI = sn_all,shared,US
# @ rset = RSET_MCM_AFFINITY
# @ mcm_affinity_options = mcm_mem_req mcm_distribute mcm_sni_none
# @ task_affinity = core(4)
# @ cpus_per_core = 1
# @ parallel_threads = 4
# @ node = 2
# @ tasks_per_node = 8
# @ collective_groups = 64
# @ queue
export MP_LABELIO=yes
export LANG=en_US
export MP_SHARED_MEMORY=yes
export MP_SINGLE_THREAD=yes
export MP_DEVTYPE=hfi
export MP_USE_BULK_XFER=yes
export MP_RDMA_ROUTE_MODE="hw_indirect"
export MP_FIFO_ROUTE_MODE="hw_indirect"
poe ./hybrid.exe
```

Figure 2-22 Job Command File for Hybrid (MPI + OpenMP) job

If the job command file is composed correctly and as you intended, you submit the job by using the **llsubmit** command. Example 2-15 shows the usage of the **llsubmit** command and the message that is received after the command is issued.

Example 2-15 llsubmit command

```
$ llsubmit ll_mpi.cmd
llsubmit: The job "c250f10c12ap02-hf0.ppd.pok.ibm.com.49" has been submitted.
```

2.6.2 Querying and managing jobs

Querying and managing jobs is described in this section.

Querying job status

After a job is submitted by using the **llsubmit** command, you use the **llq** command to query and display the LoadLeveler job queue. Example 2-16 on page 147 shows the usage of the **llq** command and the message that is received after the command is issued.

Example 2-16   llq command

```
$ llq
Id                       Owner      Submitted   ST PRI Class        Running On
------------------------ ---------- ----------- -- --- ------------ -----------
c250f10c12ap02-hf0.49.0  itsohpc    10/28 10:10 R  50  X_Class
c250f10c12ap09-hf0

1 job step(s) in queue, 0 waiting, 0 pending, 1 running, 0 held, 0 preempted
```

You also use the **llq** command with the **-l** flag to ascertain the detail information for the job, as shown in Example 2-17. The job shows the Resource Set information, the output for task instances, and allocated hosts if the job requested MCM affinity.

Example 2-17   Detail information of job

```
$ llq -l c250f10c12ap02-hf0.49.0
===== Job Step c250f10c12ap02-hf0.ppd.pok.ibm.com.49.0 =====
        Job Step Id: c250f10c12ap02-hf0.ppd.pok.ibm.com.49.0
           Job Name: myjob.mpi
          Step Name: 0
  Structure Version: 10
              Owner: itsohpc
         Queue Date: Fri Oct 28 10:10:12 2011
             Status: Running
     Reservation ID:
  Requested Res. ID:
   Flexible Res. ID:
          Recurring: False
 Scheduling Cluster:
 Submitting Cluster:
    Sending Cluster:
  Requested Cluster:
     Schedd History:
   Outbound Schedds:
    Submitting User:
   Eligibility Time: Fri Oct 28 10:10:12 2011
      Dispatch Time: Fri Oct 28 10:10:12 2011
    Completion Date:
    Completion Code:
        Favored Job: No
      User Priority: 50
       user_sysprio: 0
      class_sysprio: 0
      group_sysprio: 0
    System Priority: -160052
          q_sysprio: -160052
 Previous q_sysprio: 0
      Notifications: Complete
 Virtual Image Size: 1 kb
         Large Page: N
              Trace: no
         Coschedule: no
       SMT required: as_is
    MetaCluster Job: no
     Checkpointable: no
```

```
            Ckpt Start Time:
        Good Ckpt Time/Date:
           Ckpt Elapse Time: 0 seconds
        Fail Ckpt Time/Date:
            Ckpt Accum Time: 0 seconds
            Checkpoint File:
           Ckpt Execute Dir:
          Restart From Ckpt: no
         Restart Same Nodes: no
                    Restart: yes
                Preemptable: yes
         Preempt Wait Count: 0
             Hold Job Until:
             User Hold Time: 00:00:00 (0 seconds)
                       RSet: RSET_MCM_AFFINITY
        Mcm Affinity Option: MCM_DISTRIBUTE MCM_MEM_REQ MCM_SNI_NONE
              Task Affinity:
              Cpus Per Core:  0
           Parallel Threads:  0
                        Env:
                         In: /dev/null
                        Out: myjob.mpi.out
                        Err: myjob.mpi.err
        Initial Working Dir: /sn_local/home/itsohpc/buha/llcmd/03_mpi
                 Dependency:
        Data Stg Dependency:
                  Resources: ConsumableCpus(1)
             Node Resources: CollectiveGroups(64)
             Step Resources:
                  Step Type: General Parallel
                 Node Usage: shared
            Submitting Host: c250f10c12ap02-hf0.ppd.pok.ibm.com
                Schedd Host: c250f10c12ap02-hf0.ppd.pok.ibm.com
              Job Queue Key:
                Notify User: itsohpc@c250f10c12ap02-hf0.ppd.pok.ibm.com
                      Shell: /bin/ksh
          LoadLeveler Group: No_Group
                      Class: X_Class
            Ckpt Hard Limit: undefined
            Ckpt Soft Limit: undefined
             Cpu Hard Limit: undefined
             Cpu Soft Limit: undefined
            Data Hard Limit: undefined
            Data Soft Limit: undefined
              As Hard Limit: undefined
              As Soft Limit: undefined
           Nproc Hard Limit: undefined
           Nproc Soft Limit: undefined
         Memlock Hard Limit: undefined
         Memlock Soft Limit: undefined
           Locks Hard Limit: undefined
           Locks Soft Limit: undefined
          Nofile Hard Limit: undefined
          Nofile Soft Limit: undefined
            Core Hard Limit: undefined
```

```
    Core Soft Limit: undefined
    File Hard Limit: undefined
    File Soft Limit: undefined
   Stack Hard Limit: undefined
   Stack Soft Limit: undefined
     Rss Hard Limit: undefined
     Rss Soft Limit: undefined
Step Cpu Hard Limit: undefined
Step Cpu Soft Limit: undefined
Wall Clk Hard Limit: 2+00:00:00 (172800 seconds)
Wall Clk Soft Limit: 2+00:00:00 (172800 seconds)
            Comment:
            Account:
         Unix Group: usr
Negotiator Messages:
      Bulk Transfer: Yes
Adapter Requirement:
(sn_all,mpi,US,shared,AVERAGE,instances=1,imm_send_buffers=1,collective_groups=64)
          Step Cpus: 64
Step Virtual Memory: 0.000 mb
   Step Real Memory: 0.000 mb
Step Large Page Mem: 0.000 mb
     Cluster Option: none
     Topology Group:
Topology Requirement: none
     Network Usages: 0(1,mpi,US,1,0,1,64),

Stripe Min Networks: False
    Monitor Program:
--------------------------------------------------------------------------------
Node
----

   Name            :
   Requirements    :
   Preferences     :
   Node minimum    : 2
   Node maximum    : 2
   Node actual     : 2
   Allocated Hosts : c250f10c12ap09-hf0.ppd.pok.ibm.com::,MCM0:CPU< 0 >,MCM0:CPU<
1 >,MCM0:CPU< 2 >,MCM0:CPU< 3 >,MCM0:CPU< 4 >,MCM0:CPU< 5 >,MCM0:CPU< 6
>,MCM0:CPU< 7 >,MCM0:CPU< 8 >,MCM0:CPU< 9 >,MCM0:CPU< 10 >,MCM0:CPU< 11
>,MCM0:CPU< 12 >,MCM0:CPU< 13 >,MCM0:CPU< 14 >,MCM0:CPU< 15 >,MCM0:CPU< 16
>,MCM0:CPU< 17 >,MCM0:CPU< 18 >,MCM0:CPU< 19 >,MCM0:CPU< 20 >,MCM0:CPU< 21
>,MCM0:CPU< 22 >,MCM0:CPU< 23 >,MCM0:CPU< 24 >,MCM0:CPU< 25 >,MCM0:CPU< 26
>,MCM0:CPU< 27 >,MCM0:CPU< 28 >,MCM0:CPU< 29 >,MCM0:CPU< 30 >,MCM0:CPU< 31 >
                   + c250f10c12ap13-hf0.ppd.pok.ibm.com::,MCM0:CPU< 0 >,MCM1:CPU<
16 >,MCM2:CPU< 32 >,MCM3:CPU< 48 >,MCM0:CPU< 1 >,MCM1:CPU< 17 >,MCM2:CPU< 33
>,MCM3:CPU< 49 >,MCM0:CPU< 2 >,MCM1:CPU< 18 >,MCM2:CPU< 34 >,MCM3:CPU< 50
>,MCM0:CPU< 3 >,MCM1:CPU< 19 >,MCM2:CPU< 35 >,MCM3:CPU< 51 >,MCM0:CPU< 4
>,MCM1:CPU< 20 >,MCM2:CPU< 36 >,MCM3:CPU< 52 >,MCM0:CPU< 5 >,MCM1:CPU< 21
>,MCM2:CPU< 37 >,MCM3:CPU< 53 >,MCM0:CPU< 6 >,MCM1:CPU< 22 >,MCM2:CPU< 38
>,MCM3:CPU< 54 >,MCM0:CPU< 7 >,MCM1:CPU< 23 >,MCM2:CPU< 39 >,MCM3:CPU< 55 >

   Master Task
```

```
-----------

   Executable   : /sn_local/home/itsohpc/buha/llcmd/03_mpi/ll_mpi.cmd
   Exec Args    :
   Num Task Inst: 1
   Task Instance: c250f10c12ap09-hf0:-1:,

Task
----

   Num Task Inst: 64
   Task Instance: c250f10c12ap09-hf0:0:,MCM0:CPU< 0 >
   Task Instance: c250f10c12ap09-hf0:1:,MCM0:CPU< 1 >
   Task Instance: c250f10c12ap09-hf0:2:,MCM0:CPU< 2 >
   Task Instance: c250f10c12ap09-hf0:3:,MCM0:CPU< 3 >
   Task Instance: c250f10c12ap09-hf0:4:,MCM0:CPU< 4 >
   Task Instance: c250f10c12ap09-hf0:5:,MCM0:CPU< 5 >
   Task Instance: c250f10c12ap09-hf0:6:,MCM0:CPU< 6 >
   Task Instance: c250f10c12ap09-hf0:7:,MCM0:CPU< 7 >
   Task Instance: c250f10c12ap09-hf0:8:,MCM0:CPU< 8 >
   Task Instance: c250f10c12ap09-hf0:9:,MCM0:CPU< 9 >
   Task Instance: c250f10c12ap09-hf0:10:,MCM0:CPU< 10 >
   Task Instance: c250f10c12ap09-hf0:11:,MCM0:CPU< 11 >
   Task Instance: c250f10c12ap09-hf0:12:,MCM0:CPU< 12 >
   Task Instance: c250f10c12ap09-hf0:13:,MCM0:CPU< 13 >
   Task Instance: c250f10c12ap09-hf0:14:,MCM0:CPU< 14 >
   Task Instance: c250f10c12ap09-hf0:15:,MCM0:CPU< 15 >
   Task Instance: c250f10c12ap09-hf0:16:,MCM0:CPU< 16 >
   Task Instance: c250f10c12ap09-hf0:17:,MCM0:CPU< 17 >
   Task Instance: c250f10c12ap09-hf0:18:,MCM0:CPU< 18 >
   Task Instance: c250f10c12ap09-hf0:19:,MCM0:CPU< 19 >
   Task Instance: c250f10c12ap09-hf0:20:,MCM0:CPU< 20 >
   Task Instance: c250f10c12ap09-hf0:21:,MCM0:CPU< 21 >
   Task Instance: c250f10c12ap09-hf0:22:,MCM0:CPU< 22 >
   Task Instance: c250f10c12ap09-hf0:23:,MCM0:CPU< 23 >
   Task Instance: c250f10c12ap09-hf0:24:,MCM0:CPU< 24 >
   Task Instance: c250f10c12ap09-hf0:25:,MCM0:CPU< 25 >
   Task Instance: c250f10c12ap09-hf0:26:,MCM0:CPU< 26 >
   Task Instance: c250f10c12ap09-hf0:27:,MCM0:CPU< 27 >
   Task Instance: c250f10c12ap09-hf0:28:,MCM0:CPU< 28 >
   Task Instance: c250f10c12ap09-hf0:29:,MCM0:CPU< 29 >
   Task Instance: c250f10c12ap09-hf0:30:,MCM0:CPU< 30 >
   Task Instance: c250f10c12ap09-hf0:31:,MCM0:CPU< 31 >
   Task Instance: c250f10c12ap13-hf0:32:,MCM0:CPU< 0 >
   Task Instance: c250f10c12ap13-hf0:33:,MCM1:CPU< 16 >
   Task Instance: c250f10c12ap13-hf0:34:,MCM2:CPU< 32 >
   Task Instance: c250f10c12ap13-hf0:35:,MCM3:CPU< 48 >
   Task Instance: c250f10c12ap13-hf0:36:,MCM0:CPU< 1 >
   Task Instance: c250f10c12ap13-hf0:37:,MCM1:CPU< 17 >
   Task Instance: c250f10c12ap13-hf0:38:,MCM2:CPU< 33 >
   Task Instance: c250f10c12ap13-hf0:39:,MCM3:CPU< 49 >
   Task Instance: c250f10c12ap13-hf0:40:,MCM0:CPU< 2 >
   Task Instance: c250f10c12ap13-hf0:41:,MCM1:CPU< 18 >
   Task Instance: c250f10c12ap13-hf0:42:,MCM2:CPU< 34 >
   Task Instance: c250f10c12ap13-hf0:43:,MCM3:CPU< 50 >
```

```
      Task Instance: c250f10c12ap13-hf0:44:,MCM0:CPU< 3 >
      Task Instance: c250f10c12ap13-hf0:45:,MCM1:CPU< 19 >
      Task Instance: c250f10c12ap13-hf0:46:,MCM2:CPU< 35 >
      Task Instance: c250f10c12ap13-hf0:47:,MCM3:CPU< 51 >
      Task Instance: c250f10c12ap13-hf0:48:,MCM0:CPU< 4 >
      Task Instance: c250f10c12ap13-hf0:49:,MCM1:CPU< 20 >
      Task Instance: c250f10c12ap13-hf0:50:,MCM2:CPU< 36 >
      Task Instance: c250f10c12ap13-hf0:51:,MCM3:CPU< 52 >
      Task Instance: c250f10c12ap13-hf0:52:,MCM0:CPU< 5 >
      Task Instance: c250f10c12ap13-hf0:53:,MCM1:CPU< 21 >
      Task Instance: c250f10c12ap13-hf0:54:,MCM2:CPU< 37 >
      Task Instance: c250f10c12ap13-hf0:55:,MCM3:CPU< 53 >
      Task Instance: c250f10c12ap13-hf0:56:,MCM0:CPU< 6 >
      Task Instance: c250f10c12ap13-hf0:57:,MCM1:CPU< 22 >
      Task Instance: c250f10c12ap13-hf0:58:,MCM2:CPU< 38 >
      Task Instance: c250f10c12ap13-hf0:59:,MCM3:CPU< 54 >
      Task Instance: c250f10c12ap13-hf0:60:,MCM0:CPU< 7 >
      Task Instance: c250f10c12ap13-hf0:61:,MCM1:CPU< 23 >
      Task Instance: c250f10c12ap13-hf0:62:,MCM2:CPU< 39 >
      Task Instance: c250f10c12ap13-hf0:63:,MCM3:CPU< 55 >
--------------------------------------------------------------------------------

1 job step(s) in query, 0 waiting, 0 pending, 1 running, 0 held, 0 preempted
```

Querying machine status

You use the **llstatus** command to ascertain the status information about machines in the LoadLeveler cluster.

The **llstatus** command features three levels of output: cluster, machine_group, and machine. The default output is cluster, which gives a summary for the machines in the cluster. The level is set by exporting the environment variable LOAD_STATUS_LEVEL or is specified on the command line.

You receive the output similar to the output shown in Example 2-18 when you issue the **llstatus** command.

Example 2-18 llstatus command

```
$ llstatus
Active         5/6
Schedd         3/4                  1 job steps
Startd         3/4                 64 running tasks


The Central Manager is defined on c250f10c12ap01-hf0.ppd.pok.ibm.com

Absent:         1
Startd:      Down   Drained   Draining      Flush    Suspend
                0          0          0          0          0
Schedd:      Down   Drained   Draining
                0          0          0
```

The output in Example 2-18 indicates that there are six machines that are defined in the administration file or database, and five machines reported their status to the Resource Manager daemon. Of the four machines that are known by the Central Manager to be Schedd machines, three machines are active and able to accept jobs. Among these Schedd machines, there is one machine queued job step in various states. Of the four machines that are known to be Startd machines, three are active and already running jobs or able to run jobs. Among these three Startd machines, there are 64 running tasks that belong to various jobs. Finally, the Central Manager is identified.

Example 2-19 shows the output for the machine level by using the **`llstatus -L`** machine command.

Example 2-19 llstatus -L machine

```
$ llstatus -L machine
Name                      Schedd InQ  Act Startd Run LdAvg Idle Arch      OpSys
c250f10c12ap01-hf0.ppd.po Avail     0    0 Down     0 0.00     0 R6000     AIX71
c250f10c12ap02-hf0.ppd.po Avail     1    1 Down     0 0.00     0 R6000     AIX71
c250f10c12ap05-hf0.ppd.po Avail     0    0 Idle     0 2.00   194 R6000     AIX71
c250f10c12ap09-hf0.ppd.po Down      0    0 Run     32 0.01  9999 R6000     AIX71
c250f10c12ap13-hf0.ppd.po Down      0    0 Run     32 0.03  9999 R6000     AIX71

R6000/AIX71                 5 machines      1  jobs     64  running tasks
Total Machines              5 machines      1  jobs     64  running tasks

The Central Manager is defined on c250f10c12ap01-hf0.ppd.pok.ibm.com

The BACKFILL scheduler is in use

The following machine is absent
c250mgrs40-itso.ppd.pok.ibm.com
```

Example 2-20 shows the status of the consumable resources that are associated with all of the machines in the LoadLeveler cluster.

Example 2-20 Status of the consumable resources

```
llstatus -R
Machine                        Consumable Resource(Available, Total)
------------------------------ -------------------------------------------------
c250f10c12ap01-hf0.ppd.pok.i-       #
c250f10c12ap02-hf0.ppd.pok.i-       #
c250f10c12ap05-hf0.ppd.pok.ib- CollectiveGroups(64,64)+< ConsumableCpus< 0-61 ><
0-61 >
c250f10c12ap09-hf0.ppd.pok.ib- CollectiveGroups(0,64)+< ConsumableCpus< 32-63 ><
0-63 >
c250f10c12ap13-hf0.ppd.pok.ib- CollectiveGroups(0,64)+< ConsumableCpus< 8-15 24-31
40-47 56-63 >< 0-63 >

Resources with "+" appended to their names have the Total value reported from
Startd.
Resources with "<" appended to their names were created automatically.
LoadL_startd daemons of machines with "#" appended to their names are down.
```

CPU ID < > notation: The individual CPU ID < > notation is used to list individual CPU IDs instead of the CPU count ( ) notation for machines in which the RSET_SUPPORT configuration file keyword is set to RSET_MCM_AFFINITY.

Canceling the job

The **llcancel** command is used to cancel one or more jobs from the LoadLeveler, as shown in Example 2-21. You receive a response similar to the response shown in Example 2-21.

Example 2-21 llcancel command to cancel jobs

```
$ llcancel c250f10c12ap02-hf0.49.0
llcancel: Cancel command has been sent to the central manager.
```

After the **llcancel** command is issued, you check whether the job is canceled by using the **llq** command, as shown in Example 2-22.

Example 2-22 llq command

```
$ llq
llq: There is currently no job status to report.
```

2.6.3 Specific issues for LoadLeveler

The following issues must be considered when you are planning to use LoadLeveler.

Task assignment for an MPI job

You use the keywords that are listed in Table 2-3 to specify how LoadLeveler assigns tasks to nodes. Various task assignment keywords are used in combination, and other keywords are mutually exclusive.

Table 2-3 Valid combinations of task assignment keywords are listed in each column

Keyword	Valid Combination				
total_tasks	X	X			
tasks_per_node			X	X	
node = <min,max>			X		
node = <number>	X			X	
task_geometry					X
blocking		X			

For more information about task assignments for an MPI LoadLeveler job, see *IBM LoadLeveler Using and Administering*, SC23-6792-03 at this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp

In addition, you use the host_file keyword to assign MPI tasks to nodes as shown in the following example:

```
# @ host_file = host_list_file_name
```

The host_list_file_name file contains the host list for task allocations, and LoadLeveler finds this file under the current working directory if a full path name is not specified.

Important: For host file usage, the following considerations are present:

- Leading and trailing tabs and spaces are removed.
- Blank lines are deleted.
- Comment lines that start with a # symbol are skipped.
- Tabs or spaces before the first comment line are allowed.
- No more than one word per line is allowed; otherwise, the entire line is treated as a host name.

LoadLeveler affinity settings

LoadLeveler provides affinity scheduling functions when users run the jobs through LoadLeveler. To use the affinity scheduling functions, LoadLeveler must set the rset_support keyword to the value rset_mcm_affinity in the LoadLeveler configuration file.

The rset keyword also must be set to rset_mcm_affinity to force the creation of a rset for each MPI task in the following job command file:

```
# @ rset = rset_mcm_affinity
```

Each task is constrained to run in an MCM affinity domain if no other options are set, but the mcm_affinity_options keyword must be set so that performance is not affected. By using the default value, tasks accumulate on the same MCM, which leaves other MCMs idle. This condition might be good for a small job that does not need more resources than a single MCM provides, but scheduling tasks round-robin across MCMs is preferable. The following value for mcm_affinity_options for MPI jobs that run on Power 775 might be used:

```
#@mcm_affinity_options=mcm_distribute mcm_mem_req mcm_sni_none
```

The mcm_affinity_options keyword must define one of the types of affinity options: task_mcm_allocation, memory_affinity, and adapter_affinity, which result in the following values:

- task_mcm_allocation options:
  - mcm_accumulate (tasks are placed on the same MCM, when possible)
  - mcm_distribute (tasks are distributed round-robin across available MCMs)
- memory_affinity options:
  - The default value must suffice here, but on AIX, it is suggested that MEMORY_AFFINITY=MCM is exported into the environment of the job so that local memory requests are made
  - mcm_mem_none (the job has no memory affinity requirement)
  - mcm_mem_pref (memory affinity preferred)
  - mcm_mem_req (memory affinity required)
- adapter_affinity options:
  - mcm_sni_none (no network adapter affinity requirement)
  - mcm_sni_pref (network adapter affinity preferred)
  - mcm_sni_req (network adapter affinity required)
  - The recommended setting for adapter_affinity on Power 775 system is the default ’mcm_sni_none“ option. The “mcm_sni_pref” or the “mcm_sni_req” option is not suitable on IBM Power Systems.

HFI consideration

The IBM Power 775 system that uses the HFI has more features for communication with collective_groups and imm_send_buffers. The administrator must set the configuration keyword SCHEDULE_BY_RESOURCES to include CollectiveGroups when collective_groups are used.

collective_groups

The collective_groups requests the CAU groups for the specified protocol instances of the job by using the following format:

```
Syntax: # @ collective_groups = number
```

The range of the collective_groups is 0 - 64. If the system is dedicated to your job, the value of collective_groups is set to 64. If there are other jobs that run on the same nodes that are in use, you reduce the setting so other jobs are able to use the nodes. You check the current available collective_groups on the machine in the LoadLeveler cluster by using the **llstatus-R** command, as shown in Example 2-20 on page 152. For more information, see 2.4.4, “Considerations for using CAU” on page 129.

imm_send_buffers

The imm_send_buffers requests a number of immediate send buffers for each window that is allocated for each protocol instance of the job by using the following format:

```
Syntax: # @ imm_send_buffers=number
```

The value of the imm_send_buffers must be greater than or equal to zero, and it is inherited from all the protocol instances of the job step unless the individual protocol instances are specified with their own imm_send_buffers. For more information, see 2.4.1, “Considerations for using HFI” on page 119.

3

Monitoring assets

In this chapter, the monitoring assets available for IBM Power Systems 775, AIX, and Linux HPC solution are described. The key features of the new monitoring software that are introduced with this cluster type also are described. In addition, we demonstrate how to run general and key component tests, list configurations, and access monitored data for post-processing in external systems.

This chapter describes the monitoring tools for the following components:

- LoadLeveler
- General Parallel File System
- xCAT
- DB2
- AIX and Linux systems
- Integrated Switch Network Manager
- HFI
- Reliable Scalable Cluster Technology
- Compilers environment
- Diskless resources

We also introduce a new monitoring tool that is called Toolkit for Event Analysis and Logging (TEAL) and demonstrate how to quickly check all system components.

3.1 Component monitoring

This section describes the available monitoring commands for each specific component that is used in the IBM Power Systems 775 AIX and Linux HPC solution, as shown in Table 3-2 on page 160. Some of these command outputs are analyzed and discussed to determine whether the system is experiencing a problem. If problems are persisting, the actions to take are described in Chapter 4, “Troubleshooting problems” on page 243.

For monitoring the IBM Power Systems 775 cluster, all of the highlighted commands in Table 3-2 on page 160 include examples and detailed command descriptions. For more information about monitoring, see the component document links in Table 3-1.

Table 3-1 Monitoring-related software component documentation

Component	Document	Format type	Link or Description
LoadLeveler	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_up_LoadLeveler_in_a_Stateful_Cluster#Initialize_and_Configure_LoadLeveler	HTML	SourceForge (Setting up LoadLeveler in a Statefull Cluster)
	Tivoli Workload Scheduler LoadLeveler: Using and administering	HTML, PDF	Information Center (Cluster Products): http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp
	Tivoli Workload Scheduler LoadLeveler: Command and API Reference		
General Parallel File System (GPFS)	GPFS: Administration and Programming Reference		
xCAT (General)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Documentation	HTML	SourceForge
xCAT (Linux)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_775_Startup_Procedure		
	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_Up_a_Linux_Hierarchical_Cluster		
xCAT (AIX)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Power_775_Hardware_Management		
DB2 9.7	System commands (under Commands and Database reference)	HTML	Information Center (IBM DB2 Database for Linux, UNIX, and Windows Information Center): http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp

Component	Document	Format type	Link or Description
DB2 9.7 (for xCAT)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_Up_DB2_as_the_xCAT_DB	HTML	SourceForge (Main page)
	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_Up_DB2_as_the_xCAT_DB#Verify_DB2_setup		SourceForge (Verify DB2 Setup)
	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_Up_DB2_as_the_xCAT_DB#Useful_DB2_Commands		SourceForge (Useful DB2 Commands)
NMON	http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon	HTML	developerWorks®
	http://www.ibm.com/developerworks/aix/library/au-analyze_aix/		developerWorks
	http://nmon.sourceforge.net/pmwiki.php		SourceForge
AIX 7.1	http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cre%2Fcre-kickoff.htm	HTML, PDF	Information Center (AIX 7.1 Information Center)
Linux (Red Hat)	http://docs.redhat.com/docs/en-US/index.html	HTML, PDF	RedHat (all products)
ISNM	Management Guide	PDF	High performance clustering that uses the 9125-F2C
RSCT	RSCT for AIX: Technical Reference	HTML, PDF	Information Center (Cluster Products): http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp
PE RE	Parallel Environment Runtime Edition for AIX: Operation and Use V1.1		
ESSL	ESSL for AIX V5.1 ESSL for Linux on POWER V5.1 Guide and Reference		
Parallel ESSL	Parallel ESSL for AIX V4.1 Guide and Reference		
Diskless resources (AIX) - (iSCSI)	http://www.ibm.com/developerworks/aix/library/au-iscsi.html	HTML	developerWorks
	http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.commadmn%2Fdoc%2Fcommadmndita%2Fiscsi_config.htm	HTML	Information Center (AIX 7.1 Information Center)

Component	Document	Format type	Link or Description
Diskless resources (AIX) - (NIM)	http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.cmds/doc/aixcmds3/lsnim.htm	HTML	Information Center (AIX 7.1 Information Center)
Diskless resources (AIX)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_AIX_Diskless_Nodes	HTML	SourceForge
Diskless resources (Linux)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Linux_Statelite		
TEAL	Check Table 3-4 on page 218 for TEAL documentation.		

Command overview

Table 3-2 shows the monitoring commands overview.

Table 3-2 Software component monitoring commands

Component	Command (AIX/Linux)	Description
LoadLeveler	**`llclass`**	Queries class information.
	`llfs`	Fair share scheduling queries and operations.
	llq	Queries job status.
	`llqres`	Queries a reservation.
	llstatus	Queries machine status.
	`llsummary`	Returns a job resource information for accounting.
	`lltrace`	Queries or controls trace messages.

Component	Command (AIX/Linux)	Description
GPFS	**mmcheckquota** (check)	Checks file system user, group, and fileset quotas.
	mmdf	Queries available file space on a GPFS file system.
	mmdiag	Displays diagnostic information about the internal GPFS state on the current node.
	mmfsck	Checks and repairs a GPFS file system.
	mmgetacl	Displays the GPFS access control list of a file or directory.
	mmgetstate	Displays the state of the GPFS daemon on one or more nodes.
	mmkerninfo	Displays the current kernel bit mode.
	mmlsattr	Queries file attributes.
	mmlscallback	Lists callbacks that are currently registered in the GPFS system.
	mmlsdisk	Displays the current configuration and state of disks in a file system.
	mmlsfs	Displays file system attributes.
	mmlsmgr (check)	Displays which node is the file System Manager for the specified file systems or which node is the cluster manager.
	mmlsmount	Lists the nodes that have a given GPFS file system mounted.
	mmlsnsd	Displays NSD information for the GPFS cluster.
	mmlspolicy	Displays policy information.
	mmlsquota	Displays quota information for a user, group, or fileset.
	mmlsvdisk	Displays the current configuration and state of VDisks in the current node.
	mmgetpdisktopology	Lists VDisk topology that is found on all attached Disk Enclosures (DE).
	mmpmon (check)	Monitors GPFS performance on a per-node basis.
	mmrepquota (check)	Displays file system user, group, and fileset quotas.
	topsummary (CHECK)	Analyzer for the output of the **mmgetpdisktopology** command. Informs about the presence of Disk Enclosures (DE).

Component	Command (AIX/Linux)	Description
xCAT	`lshwconn`	Displays the connection status for FSP and BPA nodes.
	`lsslp`	Discovers selected network services information within the same subnet.
	`lsvm`	Lists partition information for the selected nodes.
	`nodestat`	Displays the running status of a node range.
	`renergy`	Displays energy information from nodes.
	`rpower`	Remote power control of nodes. Also displays information about the status of the nodes.
	`rscan`	Collects node information from one or more hardware control points.
DB2	`db2_local_ps`	Display DB2 processes status.
	`db2ilist`	List DB2 instances.
	`db2level`	Displays information about the code level of DB2 and their instances.
	`db2 connect to <DB2_instance>`	Tests (connecting) DB2 database instance (often called xcatdb).
	`db2 get database configuration for <DB2_instance>`	Displays details for an instance (often called xcatdb).
	`/usr/local/bin/isql -v <DB2_instance>`	Tests (connecting) ODBC support for a DB2 database instance (often called xcatdb).
AIX and Linux Systems	For more information, see Table 3-3 on page 194.	

Component	Command (AIX/Linux)	Description
ISNM	`service hdwr_svr status`	(Linux only) Reports the status of the Hardware Server daemon.
	`ps -ef │ grep hdwr_svr`	Displays whether the Hardware Server daemon process is running.
	`ps -ef │ grep cnmd`	Displays whether the CNM daemon process is running.
	`lsnwcomponents`	Displays information about cluster components: FSPs and BPAs.
	`lsnwdownhw`	Lists faulty hardware: Links, Hot Fabric Interfaces (HFIs), and ISRs.
	`lsnwexpnbrs`	Lists the expected neighbors of the links in a specified server in the cluster.
	`lsnwgc`	Displays current global counter information.
	`lsnwlinkinfo`	Displays information about the ISR links.
	`lsnwloc`	Displays information about frame-cage and supernode-drawer locations.
	`lsnwmiswire`	Displays information about miswired links.
	`lsnwtopo`	Displays cluster network topology information.
	`nwlinkdiag`	Diagnoses an ISR network link.
HFI (775 only)	`hfi_read`	Reads data from HFI logical device.
	`hfi_read_regs`	Displays HFI registers values.
	`hfi_dump`	Dumps HFI logical device (hfi).
	`ifhfi_dump`	Dumps HFI device (hf).
RSCT	`lssrc -s ctrmc`	Displays information about the status of the RMC daemon.
	`lssrc -l -s ctrmc`	Displays information about the RMC subsystem.
PE Runtime Edition	`rset_query`	Displays information about the memory affinity assignments that are performed.
ESSL	NA	IBM Engineering and Scientific Subroutine Library for AIX and Linux on POWER.
Parallel ESSL	NA	Parallel Engineering and Scientific Subroutine Library for AIX.
Diskless resources (NIM)	`lsnim`	Lists resources that are configured on NIM Server.
Diskless resource (iSCSI)	`service iscsi status`	Reports the status of the iSCSI daemon (Linux only).

Component	Command (AIX/Linux)	Description
Diskless resource (Network File System)	**service nfs status**	Reports the status of the Network File System (NFS) daemons (Linux only).
	lssrc -g nfs	Reports the status of the NFS daemons (AIX only).
	showmount -e	Lists the exported mount points and their access permissions.
Diskless resource (Trivial File Transfer Protocol)	**service tftpd status**	Reports the status of the Trivial File Transfer Protocol (TFTP) daemon (Linux only).
	lssrc -s tftpd	Reports the status of the TFTP daemon (AIX only).
TEAL	For more information, see Table 3-6 on page 219.	

3.1.1 LoadLeveler

In this section, we describe commonly used LoadLeveler commands. For more information, see Table 3-1 on page 158.

The following commands are described:

- **llq**
- **llstatus**

llq

For this command, we list the help description, as shown in Figure 3-1. Typical output examples are shown in Example 2-15 on page 146, and Example 2-16 on page 147.

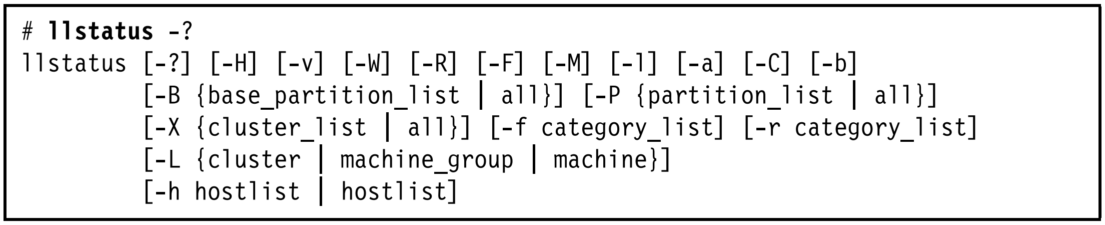
```
# llq -?
Usage: llq [ -? ] [ -H ] [ -v ] [ -W ] [ -x [ -d ] ] [ -s ] [ -p ] [ -l ] [ -m
] [ -w ] [ -X {cluster_list | all} ] [ -j joblist | joblist ] [ -u userlist ] [
-h hostlist ] [ -c classlist ] [ -R reservationlist ] [ -f category_list ] [ -r
category_list ] [ -b ] [ -S <total | user | group | class> ]
```

Figure 3-1 llq command flag description

The **llq** command queries information about jobs in LoadLeveler queues. For more information about the command reference details, see “Tivoli Workload Scheduler LoadLeveler: Command and API Reference” in Table 3-1 on page 158.

llstatus

For this command, we list the help description as shown in Figure 3-2 on page 165. Typical output examples are shown in Example 2-18 on page 151, Example 2-18 on page 151, and Example 2-19 on page 152.



```
# llstatus -?
llstatus [-?] [-H] [-v] [-W] [-R] [-F] [-M] [-l] [-a] [-C] [-b]
         [-B {base_partition_list | all}] [-P {partition_list | all}]
         [-X {cluster_list | all}] [-f category_list] [-r category_list]
         [-L {cluster | machine_group | machine}]
         [-h hostlist | hostlist]
```

Figure 3-2 llstatus command flag description

The **llstatus** command returns status information about machines in the LoadLeveler cluster. For more information about the command reference details, see “Tivoli Workload Scheduler LoadLeveler: Command and API Reference” in Table 3-1 on page 158.

3.1.2 General Parallel File System

In this section, we present two distinct General Parallel File System (GPFS) monitoring areas. For the first part, examples illustrate the general GPFS monitoring point of view. In the second part, we present specific IBM Power Systems 775 cluster examples that are introduced with the GPFS Native RAID (GNR). For for information, see Table 3-1 on page 158.

GPFS general commands

The following general GPFS commands are used to manage the GPFS file system:

- **mmdf**
- **mmgetstate**
- **mmlsdisk**
- **mmlsfs**
- **mmlsmount**
- **mmlsnsd**

These commands are described in the GPFS information in Table 3-1 on page 158. Although these commands belong to the GPFS base product, the output from the commands is not the same. In this instance, the commands are mixed with specific IBM Power Systems 775 cluster GNR support. The following sections show the example outputs from these commands.

mmdf

Example 3-1 shows the **mmdf** command output.

Example 3-1 mmdf command output

```
# mmdf gpfs1
disk              disk size  failure holds    holds            free KB
free KB
name                  in KB    group metadata data        in full blocks
in fragments
-------------- ------------ -------- -------- ----- -------------------
-------------------
Disks in storage pool: system (Maximum disk size allowed is 10 TB)
000DE22BOTDA1META   1048702976      -1 yes     no      1048541184 (100%)
1984 ( 0%)
000DE22BOTDA2META   1048702976      -1 yes     no      1048543232 (100%)
3968 ( 0%)
000DE22BOTDA3META   1048702976      -1 yes     no      1048541184 (100%)
1984 ( 0%)
```

```
000DE22BOTDA4META    1048702976      -1 yes      no      1048543232 (100%)
3968 ( 0%)
000DE22TOPDA1META    1048702976      -1 yes      no      1048541184 (100%)
1984 ( 0%)
000DE22TOPDA2META    1048702976      -1 yes      no      1048541184 (100%)
3968 ( 0%)
000DE22TOPDA3META    1048702976      -1 yes      no      1048545280 (100%)
3968 ( 0%)
000DE22TOPDA4META    1048702976      -1 yes      no      1048543232 (100%)
3968 ( 0%)
                -------------                        --------------------
-------------------
(pool total)       8389623808                           8388339712 (100%)
25792 ( 0%)

Disks in storage pool: data (Maximum disk size allowed is 64 TB)
000DE22BOTDA1DATA    6231867392      -1 no       yes      6231834624 (100%)
29696 ( 0%)
000DE22BOTDA2DATA    6442024960      -1 no       yes      6442008576 (100%)
13824 ( 0%)
000DE22BOTDA3DATA    6442024960      -1 no       yes      6442008576 (100%)
13824 ( 0%)
000DE22BOTDA4DATA    6442024960      -1 no       yes      6442008576 (100%)
13824 ( 0%)
000DE22TOPDA1DATA    6442024960      -1 no       yes      6442008576 (100%)
13824 ( 0%)
000DE22TOPDA2DATA    6442024960      -1 no       yes      6441992192 (100%)
13824 ( 0%)
000DE22TOPDA3DATA    6442024960      -1 no       yes      6441992192 (100%)
29696 ( 0%)
000DE22TOPDA4DATA    6442024960      -1 no       yes      6442008576 (100%)
13824 ( 0%)
                -------------                        --------------------
-------------------
(pool total)      51326042112                          51325861888 (100%)
142336 ( 0%)

                =============                        ====================
===================
(data)            51326042112                          51325861888 (100%)
142336 ( 0%)
(metadata)         8389623808                           8388339712 (100%)
25792 ( 0%)
                =============                        ====================
===================
(total)           59715665920                          59714201600 (100%)
168128 ( 0%)

Inode Information
-----------------
Number of used inodes:            4161
Number of free inodes:         1011647
Number of allocated inodes:    1015808
Maximum number of inodes:     58318848
```

mmgetstate

Example 3-2 shows the **mmgetstate** command output.

Example 3-2 mmgetstate command output

```
# mmgetstate -a -L -s

 Node number  Node name       Quorum  Nodes up  Total nodes  GPFS state  Remarks
----------------------------------------------------------------------------------
--
       1      c250f10c12ap05-ml0   2        2          3       active      quorum
node
       2      c250f10c12ap09-ml0   2        2          3       active      quorum
node
       3      c250f10c12ap13-ml0   2        2          3       active

 Summary information
---------------------
Number of nodes defined in the cluster:            3
Number of local nodes active in the cluster:       3
Number of remote nodes joined in this cluster:     0
Number of quorum nodes defined in the cluster:     2
Number of quorum nodes active in the cluster:      2
Quorum = 2, Quorum achieved
```

mmlsdisk

Example 3-3 shows the **mmlsdisk** command output.

Example 3-3 mmlsdisk command output

```
# mmlsdisk gpfs1
disk         driver   sector failure holds    holds
storage
name         type       size   group metadata data  status        availability
pool
------------ -------- ------ ------- -------- ----- ------------- ------------
------------
000DE22BOTDA1META nsd         512      -1 yes      no    ready         up
system
000DE22BOTDA1DATA nsd         512      -1 no       yes   ready         up
data
000DE22BOTDA2META nsd         512      -1 yes      no    ready         up
system
000DE22BOTDA2DATA nsd         512      -1 no       yes   ready         up
data
000DE22BOTDA3META nsd         512      -1 yes      no    ready         up
system
000DE22BOTDA3DATA nsd         512      -1 no       yes   ready         up
data
000DE22BOTDA4META nsd         512      -1 yes      no    ready         up
system
000DE22BOTDA4DATA nsd         512      -1 no       yes   ready         up
data
000DE22TOPDA1META nsd         512      -1 yes      no    ready         up
system
```

```
000DE22TOPDA1DATA nsd         512     -1 no       yes   ready         up
data
000DE22TOPDA2META nsd         512     -1 yes      no    ready         up
system
000DE22TOPDA2DATA nsd         512     -1 no       yes   ready         up
data
000DE22TOPDA3META nsd         512     -1 yes      no    ready         up
system
000DE22TOPDA3DATA nsd         512     -1 no       yes   ready         up
data
000DE22TOPDA4META nsd         512     -1 yes      no    ready         up
system
000DE22TOPDA4DATA nsd         512     -1 no       yes   ready         up
data
```

mmlsfs

Example 3-4 shows the **mmlsfs** command output.

Example 3-4 mmlsfs command output

```
# mmlsfs gpfs1
flag                value                    description
------------------- ------------------------ -----------------------------------
 -f                 65536                    Minimum fragment size in bytes
(system pool)
                   524288                   Minimum fragment size in bytes (other
pools)
 -i                 512                      Inode size in bytes
 -I                 32768                    Indirect block size in bytes
 -m                 1                        Default number of metadata replicas
 -M                 2                        Maximum number of metadata replicas
 -r                 1                        Default number of data replicas
 -R                 2                        Maximum number of data replicas
 -j                 scatter                  Block allocation type
 -D                 nfs4                     File locking semantics in effect
 -k                 all                      ACL semantics in effect
 -n                 150                      Estimated number of nodes that will
mount file system
 -B                 2097152                  Block size (system pool)
                    16777216                 Block size (other pools)
 -Q                 none                     Quotas enforced
                    none                     Default quotas enabled
 --filesetdf        no                       Fileset df enabled?
 -V                 12.10 (3.4.0.7)          File system version
 --create-time      Thu Oct 20 17:09:34 2011 File system creation time
 -u                 yes                      Support for large LUNs?
 -z                 no                       Is DMAPI enabled?
 -L                 4194304                  Logfile size
 -E                 yes                      Exact mtime mount option
 -S                 no                       Suppress atime mount option
 -K                 whenpossible             Strict replica allocation option
 --fastea           yes                      Fast external attributes enabled?
 --inode-limit      58318848                 Maximum number of inodes
 -P                 system;data              Disk storage pools in file system
 -d
000DE22BOTDA1META;000DE22BOTDA1DATA;000DE22BOTDA2META;000DE22BOTDA2DATA;000DE22BOT
DA3META;000DE22BOTDA3DATA;000DE22BOTDA4META;000DE22BOTDA4DATA;000DE22TOPDA1META;
 -d
000DE22TOPDA1DATA;000DE22TOPDA2META;000DE22TOPDA2DATA;000DE22TOPDA3META;000DE22TOP
DA3DATA;000DE22TOPDA4META;000DE22TOPDA4DATA   Disks in file system
 -A                 yes                       Automatic mount option
 -o                 none                      Additional mount options
 -T                 /gpfs1                    Default mount point
 --mount-priority   0                         Mount priority
```

mmlsmount

Example 3-5 shows the **mmlsmount** command output.

Example 3-5 mmlsmount command output

```
# mmlsmount gpfs1 -L -C all

File system gpfs1 is mounted on 3 nodes:
  30.10.12.5      c250f10c12ap05-ml0        c250f10c12ap13-ml0.ppd.pok.ibm.com
  30.10.12.13     c250f10c12ap13-ml0        c250f10c12ap13-ml0.ppd.pok.ibm.com
  30.10.12.9      c250f10c12ap09-ml0        c250f10c12ap13-ml0.ppd.pok.ibm.com
```

mmlsnsd

Example 3-6 shows the **mmlsnds** command output.

Example 3-6 mmlsnsd command output

```
# mmlsnsd -a -L

 File system   Disk name    NSD volume ID      NSD servers
-------------------------------------------------------------------------------------
-----------
 gpfs1         000DE22BOTDA1META 140A0C0D4EA07BA2
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA1DATA 140A0C0D4EA07C5A
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA2META 140A0C0D4EA07ECA
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA2DATA 140A0C0D4EA07FB5
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA3META 140A0C0D4EA08096
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA3DATA 140A0C0D4EA080F4
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA4META 140A0C0D4EA0818D
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22BOTDA4DATA 140A0C0D4EA081DA
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA1META 140A0C0D4EA083F3
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA1DATA 140A0C0D4EA084C0
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA2META 140A0C0D4EA086F0
c250f10c12ap13-ml0.ppd.pok.ibm.com
```

```
 gpfs1         000DE22TOPDA2DATA 140A0C0D4EA0876E
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA3META 140A0C0D4EA088A5
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA3DATA 140A0C0D4EA088EE
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA4META 140A0C0D4EA089B5
c250f10c12ap13-ml0.ppd.pok.ibm.com
 gpfs1         000DE22TOPDA4DATA 140A0C0D4EA089F2
c250f10c12ap13-ml0.ppd.pok.ibm.com
```

GNR commands

Support and VDisk implementation GPFS features new commands for the GNR. Some commands deliver the following monitor capabilities for the new subsystem components:

- **mmlsvdisk**
- **mmgetpdisktopology**
- **topsummary**

mmlsvdisk

For this command, we list the help description in Figure 3-3. Typical output examples are shown in Example 3-7 on page 171, Example 3-8 on page 172, Example 3-9 on page 172, and Example 3-10 on page 174.

```
# mmlsvdisk -h
[E] Usage:
  mmlsvdisk [--vdisk "VdiskName[;VdiskName...]" | --non-nsd]
    or
  mmlsvdisk --recovery-group RecoveryGroupName
     [--declustered-array DeclusteredArrayName]
```

Figure 3-3 mmlsvdisk command flag description

The **mmlsvdisk** command lists the GNR topology configuration. The following input and output values are shown:

- Command: **mmlsvdisk**
- Flags:
  - --non-nsd: Displays non-NSD type disks and LOG disks.
  - --vdisk “vdisk_name”: Displays VDisk detailed information about the <vdisk_name>.
  - --recovery-group “recovery_group_name”: Displays VDisk information about the <recovery_group_name>.
- Outputs:

  [no-flags]
  <vdisk_name> <RAID_code> <recovery_group_name> <declustered_array>
                    <block_size>
  [--non-nsd]
  Same as with [no-flags] but only for LOG VDisks, as shown in Example 3-8 on page 172.
  [--recovery-group]
  vdisk:
    name = "<vdisk_name>"
    raidCode = "<RAID_code>"
    recoveryGroup = "<recovery_group_name>"

```
declusteredArray = "<declustered_array>"
blockSizeInKib = <block_size>
size = "<vdisk_size>"
state = "[ok | ]"
remarks = "[log]"
```

- Designations:
  - State: State of the VDisk.
  - Remarks: Optional attribute for log device.
- Attributes:
  - <vdisk_name>: Logical name of the VDisk.
  - <RAID_code>: Type of RAID used for the VDisk, which is three-way or four-way replication or two-fault or three-fault tolerant.
  - <recovery_group_name>: Name of the recovery group out of the two groups available.
  - <declustered_array>: Name of the Disk Array (DA), which ranges from 2 - 8 arrays per disk enclosure.
  - <block_size>: Block size of the VDisk in kilobytes.
  - <vdisk_size>: Size of VDisk in [T,G,M,K]Bytes.

> **For more information:** For more information about the overall concepts of GNR, see “GPFS Native RAID” on page 77.

Example 3-7 mmlsvdisk output example

```
# mmlsvdisk
                                                           declustered  block size
 vdisk name          RAID code        recovery group         array       in KiB
remarks
 ------------------  ---------------  ------------------  -----------  ----------
-------
 000DE22BOTDA1DATA   8+3p             000DE22BOT           DA1                16384
 000DE22BOTDA1META   4WayReplication  000DE22BOT           DA1                 2048
 000DE22BOTDA2DATA   8+3p             000DE22BOT           DA2                16384
 000DE22BOTDA2META   4WayReplication  000DE22BOT           DA2                 2048
 000DE22BOTDA3DATA   8+3p             000DE22BOT           DA3                16384
 000DE22BOTDA3META   4WayReplication  000DE22BOT           DA3                 2048
 000DE22BOTDA4DATA   8+3p             000DE22BOT           DA4                16384
 000DE22BOTDA4META   4WayReplication  000DE22BOT           DA4                 2048
 000DE22BOTLOG       3WayReplication  000DE22BOT          LOG                   512
log
 000DE22TOPDA1DATA   8+3p             000DE22TOP           DA1                16384
 000DE22TOPDA1META   4WayReplication  000DE22TOP           DA1                 2048
 000DE22TOPDA2DATA   8+3p             000DE22TOP           DA2                16384
 000DE22TOPDA2META   4WayReplication  000DE22TOP           DA2                 2048
 000DE22TOPDA3DATA   8+3p             000DE22TOP           DA3                16384
 000DE22TOPDA3META   4WayReplication  000DE22TOP           DA3                 2048
 000DE22TOPDA4DATA   8+3p             000DE22TOP           DA4                16384
 000DE22TOPDA4META   4WayReplication  000DE22TOP           DA4                 2048
 000DE22TOPLOG       3WayReplication  000DE22TOP          LOG                   512
log
```

Example 3-8 mmlsvdisk output example without DATA vdisks

```
# mmlsvdisk --non-nsd
                                                          declustered  block size
 vdisk name          RAID code        recovery group         array       in KiB
remarks
 ------------------  ---------------  ------------------  -----------  ----------
-------
 000DE22BOTLOG       3WayReplication  000DE22BOT          LOG                 512
log
 000DE22TOPLOG       3WayReplication  000DE22TOP          LOG                 512
log
```

Example 3-9 mmlsvdisk output example with recovery group detailed information

```
# mmlsvdisk --recovery-group 000DE22TOP
vdisk:
  name = "000DE22TOPLOG"
  raidCode = "3WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "LOG"
  blockSizeInKib = 512
  size = "8240 MiB"
  state = "ok"
  remarks = "log"

vdisk:
  name = "000DE22TOPDA1META"
  raidCode = "4WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA1"
  blockSizeInKib = 2048
  size = "1000 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA1DATA"
  raidCode = "8+3p"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA1"
  blockSizeInKib = 16384
  size = "6143 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA2META"
  raidCode = "4WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA2"
  blockSizeInKib = 2048
  size = "1000 GiB"
  state = "ok"
  remarks = ""
```

```
vdisk:
  name = "000DE22TOPDA2DATA"
  raidCode = "8+3p"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA2"
  blockSizeInKib = 16384
  size = "6143 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA3META"
  raidCode = "4WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA3"
  blockSizeInKib = 2048
  size = "1000 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA3DATA"
  raidCode = "8+3p"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA3"
  blockSizeInKib = 16384
  size = "6143 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA4META"
  raidCode = "4WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA4"
  blockSizeInKib = 2048
  size = "1000 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA4DATA"
  raidCode = "8+3p"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA4"
  blockSizeInKib = 16384
  size = "6143 GiB"
  state = "ok"
  remarks = ""
```

Example 3-10 mmlsvdisk output example with recovery group (one declustered array)

```
# mmlsvdisk --recovery-group 000DE22TOP --declustered-array DA1
vdisk:
  name = "000DE22TOPDA1META"
  raidCode = "4WayReplication"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA1"
  blockSizeInKib = 2048
  size = "1000 GiB"
  state = "ok"
  remarks = ""

vdisk:
  name = "000DE22TOPDA1DATA"
  raidCode = "8+3p"
  recoveryGroup = "000DE22TOP"
  declusteredArray = "DA1"
  blockSizeInKib = 16384
  size = "6143 GiB"
  state = "ok"
  remarks = ""
```

mmgetpdisktopology

For this command, there is no help description, arguments, or flags. A typical usage example is shown in Example 3-11.

The **mmgetpdisktopology** command gathers information about all of the connected DEs to the LPAR where the command is executed. The command generates the following input and output values:

- Command: **mmgetpdisktopology**
- Flags: None
- Outputs: See Example B-1 on page 324.

> **Important:** The contents for the file “/mmgetpdisktopology_file.output” in Example 3-11 are summarized in Example B-1 on page 324.

Example 3-11 mmgetpdisktopology usage example

```
# mmgetpdisktopology > /mmgetpdisktopology_file.output
```

topsummary

For this command, there is no help description or flags. The command takes only a file as an input or stdin (standard input). A typical output example is shown in Example 3-12 on page 175.

The **topsummary** command compiles all of the information from the **mmgetpdisktopology** command and generates a readable output. The command generates the following input and output values:

- Command: **topsummary** <stdin_or_file>
- Flags: NONE
- Outputs:

```
P7IH-DE enclosures found: <enclosure_name>
{
Enclosure <enclosure_name>:
{{
Enclosure <enclosure_name> STOR <physical_location_portcard> sees <portcards_list>
Portcard <portcard>: <ses>[code_A]/<mpt_sas>/<disks> diskset “<diskset_id>”
Enclosure <enclosure_name> STOR <physical_location_portcard> sees <total_disks>
[...]
}}
{{
[depends] Carrier location <physical_location> appears <description>
[...]
}}
Enclosure <enclosure_name> sees <total_enclosure_disks>
[...]
}
{
<mpt_sas>[code_B] <systemp_physical_location> <enclosure_name> {STOR <id>
                              <portcard> (<ses_connectors>) [...]}
[...]
}
```

- Attributes:
  - <stdin_or_file>: Output from the **mmgetpdisktopology** command, either by file or stdin output redirection.
  - <enclosure_name>: Name of the Disk Enclosure (DE).
  - <physical_location_portcard>: Portcards location identification.
  - <portcards_list>: Portcards detected.
  - <ses>: Portcard logical SAS device controller, one for each row of disks (four rows total).
  - <mpt_sas>: Power 775 SAS Card Link (logical).
  - <disks>: Disks visible in a specific path.
  - <diskset_id>: Disk ID representing a diskset visible in a specific path.
  - <total_disks>: Total number of disks visible in one STOR area by one or two portcards.
  - <physical_location>: Power 775 physical location for SAS cards.
  - <description>: Information about the detected problem.
  - <total_enclosure_disks>: Total number of disks visible in the DE.
  - <systemp_physical_location>: Physical location for the SAS card in the 775 CEC.
  - <id>: STOR location area ID.
  - <ses_connectors>: Logical connections to corresponding portcard.

> **Important:** The contents for the file `/output_from_mmgetpdisktopology.file` in Example 3-12 are summarized in Example B-1 on page 324.

Example 3-12 topsummary output example

```
# topsummary /output_from_mmgetpdisktopology.file
P7IH-DE enclosures found: 000DE22
Enclosure 000DE22:
```

```
Enclosure 000DE22 STOR P1-C4/P1-C5 sees both portcards: P1-C4 P1-C5
Portcard P1-C4: ses20[0154]/mpt2sas0/24 diskset "37993" ses21[0154]/mpt2sas0/24
diskset "18793"
Portcard P1-C5: ses28[0154]/mpt2sas2,mpt2sas1/24 diskset "37993"
ses29[0154]/mpt2sas2,mpt2sas1/24 diskset "18793"
Enclosure 000DE22 STOR P1-C4/P1-C5 sees 48 disks
Enclosure 000DE22 STOR P1-C12/P1-C13 sees both portcards: P1-C12 P1-C13
Portcard P1-C12: ses22[0154]/mpt2sas0/24 diskset "26285" ses23[0154]/mpt2sas0/23
diskset "44382"
Portcard P1-C13: ses30[0154]/mpt2sas2,mpt2sas1/24 diskset "26285"
ses31[0154]/mpt2sas2,mpt2sas1/23 diskset "44382"
Enclosure 000DE22 STOR P1-C12/P1-C13 sees 47 disks
Enclosure 000DE22 STOR P1-C20/P1-C21 sees both portcards: P1-C20 P1-C21
Portcard P1-C20: ses36[0154]/mpt2sas2,mpt2sas1/24 diskset "04091"
ses37[0154]/mpt2sas2,mpt2sas1/24 diskset "31579"
Portcard P1-C21: ses44[0154]/mpt2sas3/24 diskset "04091" ses45[0154]/mpt2sas3/24
diskset "31579"
Enclosure 000DE22 STOR P1-C20/P1-C21 sees 48 disks
Enclosure 000DE22 STOR P1-C28/P1-C29 sees both portcards: P1-C28 P1-C29
Portcard P1-C28: ses38[0154]/mpt2sas2,mpt2sas1/24 diskset "64504"
ses39[0154]/mpt2sas2,mpt2sas1/24 diskset "52307"
Portcard P1-C29: ses46[0154]/mpt2sas3/24 diskset "64504" ses47[0154]/mpt2sas3/24
diskset "52307"
Enclosure 000DE22 STOR P1-C28/P1-C29 sees 48 disks
Enclosure 000DE22 STOR P1-C60/P1-C61 sees both portcards: P1-C60 P1-C61
Portcard P1-C60: ses32[0154]/mpt2sas2,mpt2sas1/24 diskset "27327"
ses33[0154]/mpt2sas2,mpt2sas1/24 diskset "43826"
Portcard P1-C61: ses40[0154]/mpt2sas3/24 diskset "27327" ses41[0154]/mpt2sas3/24
diskset "43826"
Enclosure 000DE22 STOR P1-C60/P1-C61 sees 48 disks
Enclosure 000DE22 STOR P1-C68/P1-C69 sees both portcards: P1-C68 P1-C69
Portcard P1-C68: ses34[0154]/mpt2sas2,mpt2sas1/24 diskset "05822"
ses35[0154]/mpt2sas2,mpt2sas1/24 diskset "59472"
Portcard P1-C69: ses42[0154]/mpt2sas3/24 diskset "05822" ses43[0154]/mpt2sas3/24
diskset "59472"
Enclosure 000DE22 STOR P1-C68/P1-C69 sees 48 disks
Enclosure 000DE22 STOR P1-C76/P1-C77 sees both portcards: P1-C76 P1-C77
Portcard P1-C76: ses16[0154]/mpt2sas0/24 diskset "37499" ses17[0154]/mpt2sas0/24
diskset "34848"
Portcard P1-C77: ses24[0154]/mpt2sas2,mpt2sas1/24 diskset "37499"
ses25[0154]/mpt2sas2,mpt2sas1/24 diskset "34848"
Enclosure 000DE22 STOR P1-C76/P1-C77 sees 48 disks
Enclosure 000DE22 STOR P1-C84/P1-C85 sees both portcards: P1-C84 P1-C85
Portcard P1-C84: ses18[0154]/mpt2sas0/24 diskset "33798" ses19[0154]/mpt2sas0/24
diskset "40494"
Portcard P1-C85: ses26[0154]/mpt2sas2,mpt2sas1/23 diskset "56527"
ses27[0154]/mpt2sas2,mpt2sas1/24 diskset "40494"
Enclosure 000DE22 STOR P1-C84/P1-C85 sees 48 disks
Carrier location P1-C40-D1 appears empty but should have an HDD
Carrier location P1-C86-D3 appears only on the portcard P1-C84 path
Enclosure 000DE22 sees 383 disks

mpt2sas3[1005480000] U78A9.001.1122233-P1-C9-T1 000DE22 STOR 3 P1-C21 (ses44
ses45) STOR 4 P1-C29 (ses46 ses47) STOR 5 P1-C61 (ses40 ses41) STOR 6 P1-C69
(ses42 ses43)
```

```
mpt2sas2[1005480000] U78A9.001.1122233-P1-C10-T1 000DE22 STOR 3 P1-C20 (ses36
ses37) STOR 4 P1-C28 (ses38 ses39) STOR 5 P1-C60 (ses32 ses33) STOR 6 P1-C68
(ses34 ses35)
mpt2sas1[1005480000] U78A9.001.1122233-P1-C11-T1 000DE22 STOR 1 P1-C5 (ses28
ses29) STOR 2 P1-C13 (ses30 ses31) STOR 7 P1-C77 (ses24 ses25) STOR 8 P1-C85
(ses26 ses27)
mpt2sas0[1005480000] U78A9.001.1122233-P1-C12-T1 000DE22 STOR 1 P1-C4 (ses20
ses21) STOR 2 P1-C12 (ses22 ses23) STOR 7 P1-C76 (ses16 ses17) STOR 8 P1-C84
(ses18 ses19)
```

3.1.3 xCAT

For the xCAT component and its subcategories, this section presents examples of monitoring commands that are commonly used to get the status of some services, nodes, and hardware availability. We also introduce a new component specific to the IBM Power 775 cluster power consumption statistics. We also present hardware discovery commands that are important when performing problem determination tasks. For more information, see Table 3-1 on page 158.

The commands shown in this section are organized into the following distinct areas:

- General view and power management
- Hardware discovery
- Hardware connectivity

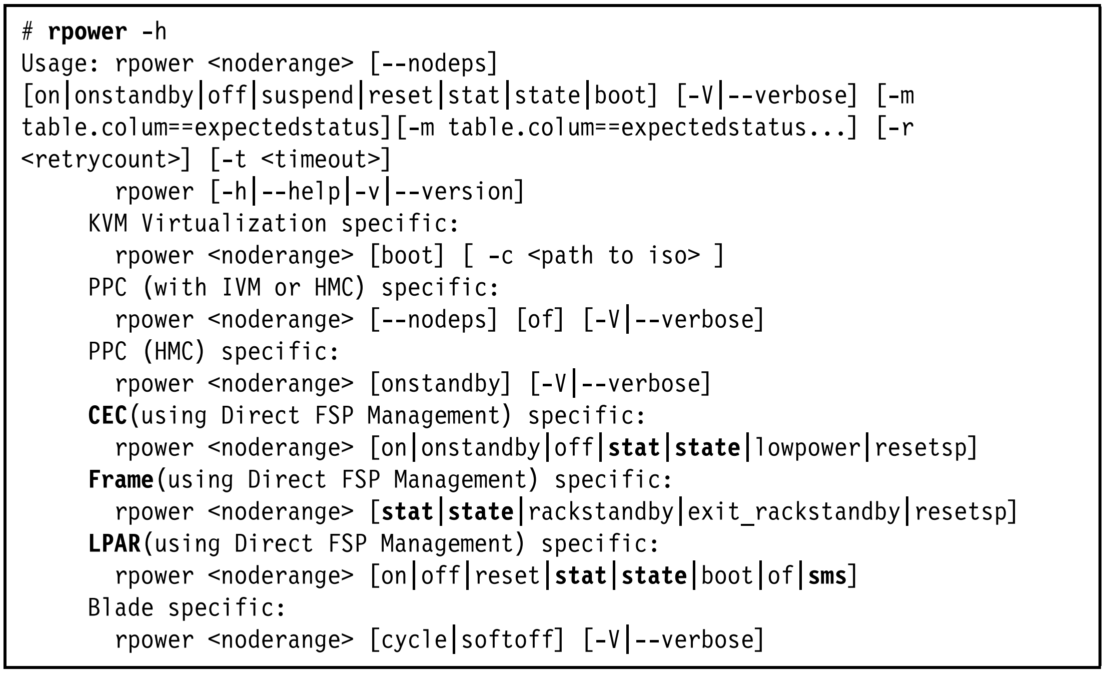
General view and power management

The following commands are used for general view and power management:

- **`rpower`**
- **`nodestat`**

rpower

For this command, we list the help description as shown in Figure 3-4 on page 178. Typical output examples are shown in Example 3-13 on page 178 and Example 3-14 on page 179.



```
# rpower -h
Usage: rpower <noderange> [--nodeps]
[on|onstandby|off|suspend|reset|stat|state|boot] [-V|--verbose] [-m
table.colum==expectedstatus][-m table.colum==expectedstatus...] [-r
<retrycount>] [-t <timeout>]
       rpower [-h|--help|-v|--version]
     KVM Virtualization specific:
       rpower <noderange> [boot] [ -c <path to iso> ]
     PPC (with IVM or HMC) specific:
       rpower <noderange> [--nodeps] [of] [-V|--verbose]
     PPC (HMC) specific:
       rpower <noderange> [onstandby] [-V|--verbose]
     CEC(using Direct FSP Management) specific:
       rpower <noderange> [on|onstandby|off|stat|state|lowpower|resetsp]
     Frame(using Direct FSP Management) specific:
       rpower <noderange> [stat|state|rackstandby|exit_rackstandby|resetsp]
     LPAR(using Direct FSP Management) specific:
       rpower <noderange> [on|off|reset|stat|state|boot|of|sms]
     Blade specific:
       rpower <noderange> [cycle|softoff] [-V|--verbose]
```

Figure 3-4 rpower command flag description

The **rpower** command remotely controls nodes and retrieves their status. Here we are focusing in the stat and state actions available for FRAME, CEC, and LPAR node types. The input and output values are:

- Command: rpower <noderange> <action>
- Outputs:

  [no-flags]
  <noderange>: <status>
- Attributes:
  - <noderange>: Nodes that are listed in xCAT database that belong to a FRAME/CEC/LPAR hardware type.
  - <action>: For monitoring cases that are one of these two options (more rpower options are available, but the options do not give monitoring information): [stat | state]
  - <status>: Status of the node. Possible values depend of noderange target type:
    - FRAME: [BPA state - <actual_state_of_BPAs> | Error: <error_description> | ...]
    - CEC: [operating | power off | IPL-in-process | Error: <error_description> | ...]
    - LPAR: [Open Firmware | Not Activated | Running | Node not found | ...]
  - <error_description>: “The connection state for the primary FSP/BPA are NOT LINE UP. The command requires that the connection for the primary FSP/BPA should be LINE UP.”

Example 3-13 rpower output example for all xcat “cec” group nodes (stat)

```
# rpower cec stat
f06cec01: operating
f06cec02: operating
f06cec03: operating
f06cec04: operating
f06cec05: operating
f06cec06: operating
f06cec07: operating
```

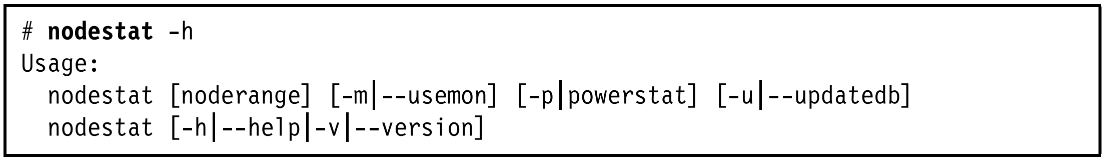
```
f06cec08: operating
f06cec09: operating
f06cec10: operating
f06cec11: operating
f06cec12: operating
```

Example 3-14 rpower output example for “m601” group nodes (state)

```
# rpower m601 stat
c250f06c01ap01-hf0: Running
c250f06c01ap05-hf0: Running
c250f06c01ap09-hf0: Running
c250f06c01ap13-hf0: Running
c250f06c01ap17-hf0: Running
c250f06c01ap21-hf0: Running
c250f06c01ap25-hf0: Running
c250f06c01ap29-hf0: Running
```

nodestat

For this command, we list the help description as shown in Figure 3-5. Typical output examples are shows in Example 3-15 on page 180 and Example 3-16 on page 180.



```
# nodestat -h
Usage:
  nodestat [noderange] [-m|--usemon] [-p|powerstat] [-u|--updatedb]
  nodestat [-h|--help|-v|--version]
```

Figure 3-5 nodestat command flag description

The **nodestat** command returns information about the status of the nodes operating system and power state and updates the xCAT database with information. This command also supports custom application status through a xCAT table and flag. The following input and output values are used:

- Command: **nodestat** <noderange>
- Flags:
  - -p: Also informs about power state
  - -m: Uses xCAT monsetting table to monitor more services with customized outputs
  - -u: Updates xCAT database with returned values
- Outputs:

  [no-flag]
  <noderange>: <status>
  [-p]
  <noderange>: <status>(<power_state>)
- Attributes:
  - <noderange>: Nodes that are listed in the xCAT database that belong to a FRAME/CEC/LPAR hardware type.
  - <status>: Information about the level of detail of the checked item.

nodestat: By default, **nodestat** completes the following steps:

1. Gets the sshd,pbs_mom, xend port status.
2. If none of the ports are open, **nodestat** gets the fping status.
3. For pingable nodes that are in the middle of deployment, **nodestat** gets the deployment status.
4. For non-pingable nodes, **nodestat** shows 'noping'.

– <power_state>: When the sate is not on or running, reports back the output of **rpower <noderange> stat** command.

Example 3-15 “nodestat” output example with power option and all nodes that are turned on

```
# nodestat m601 -p
c250f06c01ap01-hf0: sshd
c250f06c01ap05-hf0: sshd
c250f06c01ap09-hf0: sshd
c250f06c01ap13-hf0: sshd
c250f06c01ap17-hf0: sshd
c250f06c01ap21-hf0: sshd
c250f06c01ap25-hf0: sshd
c250f06c01ap29-hf0: sshd
```

Example 3-16 “nodestat” output example with power option and all nodes are off

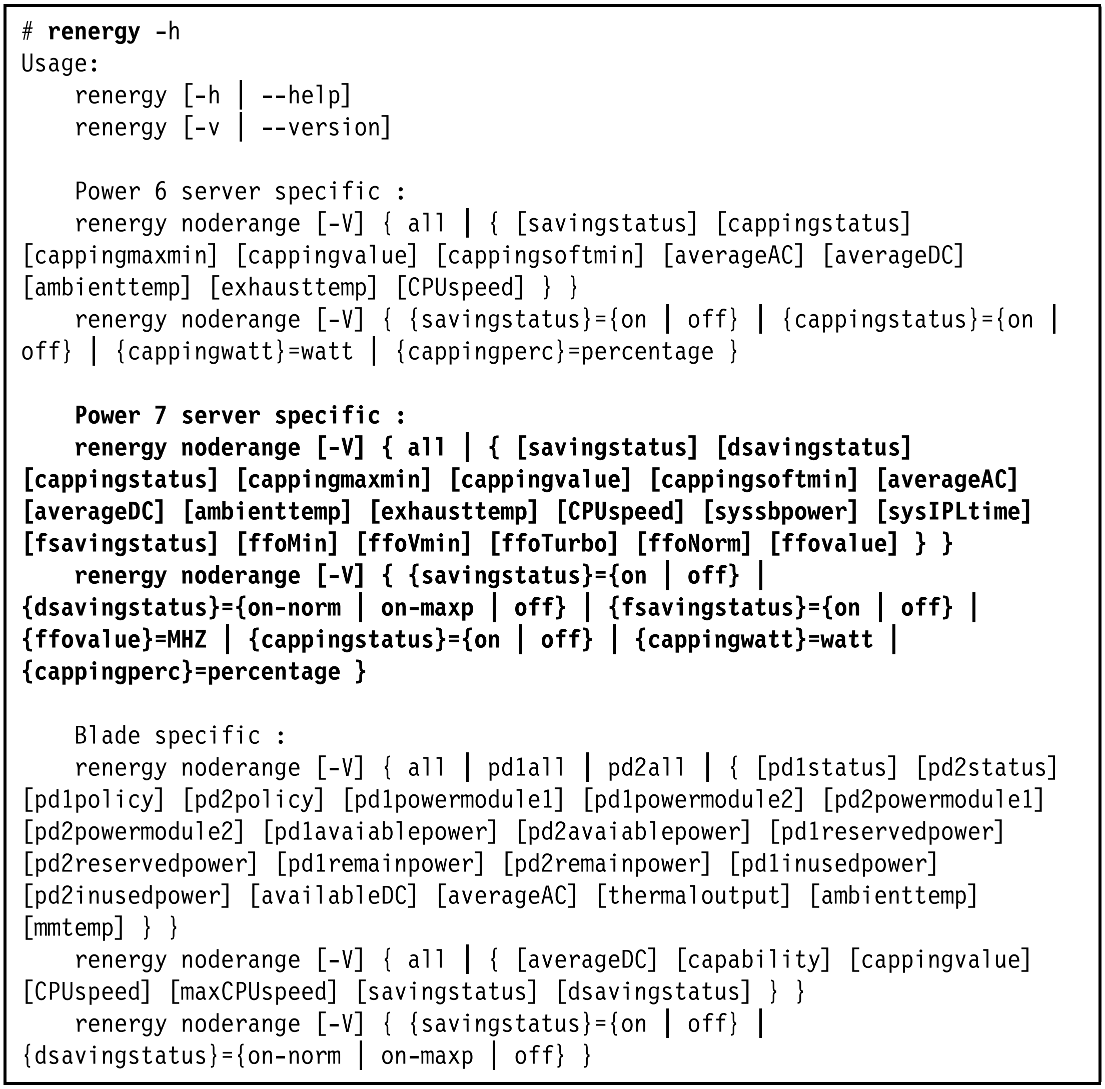
```
# nodestat m602 -p
c250f06c02ap01-hf0: noping(Not Activated)
c250f06c02ap05-hf0: noping(Not Activated)
c250f06c02ap09-hf0: noping(Not Activated)
c250f06c02ap13-hf0: noping(Not Activated)
c250f06c02ap17-hf0: noping(Not Activated)
c250f06c02ap21-hf0: noping(Not Activated)
c250f06c02ap25-hf0: noping(Not Activated)
c250f06c02ap29-hf0: noping(Not Activated)
```

Power management

The **renergy** power command is described in this section.

renergy

For this command, we list the help description as shown in Figure 3-6 on page 181. Typical output examples are shown in Example 3-17 on page 183, Example 3-18 on page 183, Example 3-19 on page 184, and Example 3-20 on page 184.



```
# renergy -h
Usage:
    renergy [-h | --help]
    renergy [-v | --version]

    Power 6 server specific :
    renergy noderange [-V] { all | { [savingstatus] [cappingstatus]
[cappingmaxmin] [cappingvalue] [cappingsoftmin] [averageAC] [averageDC]
[ambienttemp] [exhausttemp] [CPUspeed] } }
    renergy noderange [-V] { {savingstatus}={on | off} | {cappingstatus}={on |
off} | {cappingwatt}=watt | {cappingperc}=percentage }

    Power 7 server specific :
    renergy noderange [-V] { all | { [savingstatus] [dsavingstatus]
[cappingstatus] [cappingmaxmin] [cappingvalue] [cappingsoftmin] [averageAC]
[averageDC] [ambienttemp] [exhausttemp] [CPUspeed] [syssbpower] [sysIPLtime]
[fsavingstatus] [ffoMin] [ffoVmin] [ffoTurbo] [ffoNorm] [ffovalue] } }
    renergy noderange [-V] { {savingstatus}={on | off} |
{dsavingstatus}={on-norm | on-maxp | off} | {fsavingstatus}={on | off} |
{ffovalue}=MHZ | {cappingstatus}={on | off} | {cappingwatt}=watt |
{cappingperc}=percentage }

    Blade specific :
    renergy noderange [-V] { all | pd1all | pd2all | { [pd1status] [pd2status]
[pd1policy] [pd2policy] [pd1powermodule1] [pd1powermodule2] [pd2powermodule1]
[pd2powermodule2] [pd1avaiablepower] [pd2avaiablepower] [pd1reservedpower]
[pd2reservedpower] [pd1remainpower] [pd2remainpower] [pd1inusedpower]
[pd2inusedpower] [availableDC] [averageAC] [thermaloutput] [ambienttemp]
[mmtemp] } }
    renergy noderange [-V] { all | { [averageDC] [capability] [cappingvalue]
[CPUspeed] [maxCPUspeed] [savingstatus] [dsavingstatus] } }
    renergy noderange [-V] { {savingstatus}={on | off} |
{dsavingstatus}={on-norm | on-maxp | off} }
```

Figure 3-6 renergy command flag description

The **renergy** command remotely enables or disables Power7 energy features and displays energy consumption statistics and features status. The following input and output values are used:

- Command: **renergy <noderange> all**
- Flags:
  - cappingmaxmin: Maximum and minimum power consumption values; is shown as “cappingmax” and “cappingmin” values.
  - cappingvalue: Power capping value setting; must be between the cappingmaxmin values.
  - averageAC: Average AC power consumption.
  - averageDC: Average DC power consumption.
  - ambienttemp: Ambient temperature in degrees Celsius near the CEC.
  - exhausttemp: Exhaust temperature in degrees Celsius.
  - CPUspeed: Actual CPU frequency in MHz.

  - syssbpower: Service processor power consumption.
  - sysIPLtime: Time that is needed to IPL the CEC in seconds.
  - all: Show all the flags.
  - savingstatus=[on | off]: Static savings enable or disable (to enable, dsavingstatus must be off).
  - dsavingstatus=[on-norm | on-maxp | off]: Dynamic savings enable or disable (to enable, savingstatus must be off). Turning on as two modes. Normal and maximum power usage if needed.
  - fsavingstatus=[on | off]: Firmware overwrite savings enable or disable (if enabled, savingstatus and dsavingstatus are off).
  - ffovalue=[ffoMin-ffoTurbo]: Firmware overwrite value. Must be between (and include) the ffoMin and ffoTurbo values for the fsavingstatus to enable.
  - cappingstatus=[on | off]: Capping energy consumption enable or disable.
  - (cappingvalue) cappingwatt=[cappingmin-cappingmax]: Capping energy consumption value in Watts between (and including) cappingmin and cappingmax values permitted.
  - (cappingvalue) cappingperc=[0-100]: Capping energy consumption value in percentage of the value between the cappingmin and cappingmax values permitted.

- Outputs:

  [all]
  <noderange>: savingstatus: [on | off]
  <noderange>: dsavingstatus: [on-norm | on-maxp | off]
  <noderange>: cappingstatus: [on | off]
  <noderange>: cappingmin: W
  <noderange>: cappingmax: W
  <noderange>: cappingvalue: [cappingmin...cappingmax] W
  <noderange>: cappingsoftmin: W
  <noderange>: averageAC: W
  <noderange>: averageDC: W
  <noderange>: ambienttemp: C
  <noderange>: exhausttemp: C
  <noderange>: CPUspeed: MHz
  <noderange>: syssbpower: W
  <noderange>: sysIPLtime: S
  <noderange>: fsavingstatus: [on | off]
  <noderange>: ffoMin: MHz
  <noderange>: ffoVmin: MHz
  <noderange>: ffoTurbo: MHz
  <noderange>: ffoNorm: MHz
  <noderange>: ffovalue: [ffoMin-ffoTurbo] MHz

- Attributes:
  - <noderange>: Nodes that are listed in the xCAT database that belong to a CEC/FSP hardware type.

Note: When “savingstatus”, “dsavingstatus”, “fsavingstatus”, or “cappingstatus” changes, some time is needed for the remaining values to update. A message that indicates this need is shown.

Example 3-17 renergy output example for all the energy values of a 775 FSP - all options OFF

```
# renergy fsp all
40.10.12.1: savingstatus: off
40.10.12.1: dsavingstatus: off
40.10.12.1: cappingstatus: off
40.10.12.1: cappingmin: 18217 W
40.10.12.1: cappingmax: 18289 W
40.10.12.1: cappingvalue: na
40.10.12.1: cappingsoftmin: 5001 W
40.10.12.1: averageAC: 3144 W
40.10.12.1: averageDC: 8596 W
40.10.12.1: ambienttemp: 25 C
40.10.12.1: exhausttemp: 25 C
40.10.12.1: CPUspeed: 3836 MHz
40.10.12.1: syssbpower: 20 W
40.10.12.1: sysIPLtime: 900 S
40.10.12.1: fsavingstatus: off
40.10.12.1: ffoMin: 2856 MHz
40.10.12.1: ffoVmin: 2856 MHz
40.10.12.1: ffoTurbo: 3836 MHz
40.10.12.1: ffoNorm: 3836 MHz
40.10.12.1: ffovalue: 3836 MHz
```

Example 3-18 renergy output example for all the energy values - turning “savingstatus” on

```
# renergy fsp savingstatus=on
40.10.12.1: Set savingstatus succeeded.
40.10.12.1: This setting may need several minutes to take effect.
# renergy fsp all
40.10.12.1: savingstatus: on
40.10.12.1: dsavingstatus: off
40.10.12.1: cappingstatus: off
40.10.12.1: cappingmin: 18217 W
40.10.12.1: cappingmax: 18289 W
40.10.12.1: cappingvalue: na
40.10.12.1: cappingsoftmin: 5001 W
40.10.12.1: averageAC: 3144 W
40.10.12.1: averageDC: 7080 W
40.10.12.1: ambienttemp: 25 C
40.10.12.1: exhausttemp: 26 C
40.10.12.1: CPUspeed: 2856 MHz
40.10.12.1: syssbpower: 20 W
40.10.12.1: sysIPLtime: 900 S
40.10.12.1: fsavingstatus: off
40.10.12.1: ffoMin: 2856 MHz
40.10.12.1: ffoVmin: 2856 MHz
40.10.12.1: ffoTurbo: 3836 MHz
40.10.12.1: ffoNorm: 3836 MHz
40.10.12.1: ffovalue: 3836 MHz
```

Example 3-19 renergy output example - turning “fsavingstatus” on (enforcing ffo values)

```
# renergy fsp fsavingstatus=on
40.10.12.1: Set fsavingstatus succeeded.
# renergy fsp all
40.10.12.1: savingstatus: off
40.10.12.1: dsavingstatus: off
40.10.12.1: cappingstatus: off
40.10.12.1: cappingmin: 18217 W
40.10.12.1: cappingmax: 18289 W
40.10.12.1: cappingvalue: na
40.10.12.1: cappingsoftmin: 5001 W
40.10.12.1: averageAC: 3144 W
40.10.12.1: averageDC: 8696 W
40.10.12.1: ambienttemp: 25 C
40.10.12.1: exhausttemp: 26 C
40.10.12.1: CPUspeed: 3836 MHz
40.10.12.1: syssbpower: 20 W
40.10.12.1: sysIPLtime: 900 S
40.10.12.1: fsavingstatus: on
40.10.12.1: ffoMin: 2856 MHz
40.10.12.1: ffoVmin: 2856 MHz
40.10.12.1: ffoTurbo: 3836 MHz
40.10.12.1: ffoNorm: 3836 MHz
40.10.12.1: ffovalue: 3836 MHz
```

Example 3-20 renergy output example - setting up “cappingvalue” and turning “cappingstatus” on

```
# renergy fsp cappingperc=0
40.10.12.1: Set cappingperc succeeded.
40.10.12.1: cappingvalue: 18217 W
# renergy fsp cappingstatus=on
40.10.12.1: Set cappingstatus succeeded.
40.10.12.1: This setting may need several minutes to take effect.
# renergy fsp all
40.10.12.1: savingstatus: off
40.10.12.1: dsavingstatus: off
40.10.12.1: cappingstatus: on
40.10.12.1: cappingmin: 18217 W
40.10.12.1: cappingmax: 18289 W
40.10.12.1: cappingvalue: 18217
40.10.12.1: cappingsoftmin: 5001 W
40.10.12.1: averageAC: 3144 W
40.10.12.1: averageDC: 8696 W
40.10.12.1: ambienttemp: 25 C
40.10.12.1: exhausttemp: 26 C
40.10.12.1: CPUspeed: 3836 MHz
40.10.12.1: syssbpower: 20 W
40.10.12.1: sysIPLtime: 900 S
40.10.12.1: fsavingstatus: on
40.10.12.1: ffoMin: 2856 MHz
40.10.12.1: ffoVmin: 2856 MHz
40.10.12.1: ffoTurbo: 3836 MHz
40.10.12.1: ffoNorm: 3836 MHz
40.10.12.1: ffovalue: 3836 MHz
```

Hardware discovery

This section describes the following hardware discovery commands:

- **lsslp**
- **rscan**
- **lsvm**

lsslp

For this command, we list the help description that is shown in Figure 3-7. Typical output examples are shown in Example 3-21 and Example 3-22 on page 186.

```
# lsslp -h
Usage: lsslp [-h|--help|-v|--version]
       lsslp [<noderange>][-V|--verbose][-i ip[,ip..]][-w][-r|-x|-z][-n][-I][-s
FRAME|CEC|MM|IVM|RSA|HMC][-C counts][-T timeout]
             [-t tries][-m][-e cmd][-c [timeinterval[interval,..]]][--vpdtable]
             [-M vpd|switchport][--makedhcp][--updatehost][--resetnet]
```

Figure 3-7 lsslp command flag description

The **lsslp** command uses SLP protocol to discover the hardware components of the following types: BPA, CEC, FSP, and HMC. The following input and output values are used:

- Command: **lsslp -m [-s FRAME | CEC | HMC]**
- Flags:
  - [-s FRAME | CEC | HMC]: Limits the discovery to one of these types.
  - -m: Uses multicast communication for the discovery procedure.
  - -z: Converts the output into xCAT “stanza” format.
- Outputs:

  [no-flags]
  Device type-model serial-number side ip-addresses hostname
  <device> <type-model> <serial-number> <side> <ip-addresses> <hostname>
- Attributes:
  - <device>: [BPA | FSP | CEC | HMC].
  - <type-model>: XXXX-YYY model type.
  - <serial-number>: IBM 7-”hexadecimal” serial number.
  - <side>: [A-0 | A-1 | B-0 | B-1].
  - <ip-addresses>: IP for the hardware component.
  - <hostname>: Hostname for the same IP (if not present is equal to the <ip-addresses>).

Example 3-21 lsslp output example for a scan with only one FSP connected (out of two)

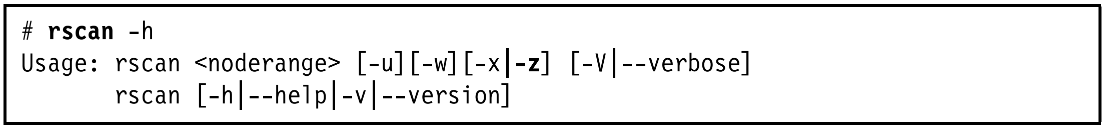
```
# lsslp -m
device  type-model  serial-number  side  ip-addresses             hostname
FSP     9125-F2C    02D7695        A-1   10.0.0.63 10.0.0.63
FSP     9125-F2C    02D7695        A-0   40.10.12.1 40.10.12.1
CEC     9125-F2C    02D7695                                       cec12
```

Example 3-22 lsslp output example for a complete frame scan with twelve 775 including HMC

```
# lsslp -m
device  type-model  serial-number  side  ip-addresses  hostname
BPA     78AC-100    992003H        A-0   40.6.0.1      40.6.0.1
BPA     78AC-100    992003H        B-0   40.6.0.2      40.6.0.2
FSP     9125-F2C    02C68B6        A-0   40.6.1.1      40.6.1.1
FSP     9125-F2C    02C68B6        B-0   40.6.1.2      40.6.1.2
FSP     9125-F2C    02C6A46        A-0   40.6.10.1     40.6.10.1
FSP     9125-F2C    02C6A46        B-0   40.6.10.2     40.6.10.2
FSP     9125-F2C    02C6A66        A-0   40.6.11.1     40.6.11.1
FSP     9125-F2C    02C6A66        B-0   40.6.11.2     40.6.11.2
FSP     9125-F2C    02C6A86        A-0   40.6.12.1     40.6.12.1
FSP     9125-F2C    02C6A86        B-0   40.6.12.2     40.6.12.2
FSP     9125-F2C    02C68D6        A-0   40.6.2.1      40.6.2.1
FSP     9125-F2C    02C68D6        B-0   40.6.2.2      40.6.2.2
FSP     9125-F2C    02C6906        A-0   40.6.3.1      40.6.3.1
FSP     9125-F2C    02C6906        B-0   40.6.3.2      40.6.3.2
FSP     9125-F2C    02C6946        A-0   40.6.4.1      40.6.4.1
FSP     9125-F2C    02C6946        B-0   40.6.4.2      40.6.4.2
FSP     9125-F2C    02C6986        A-0   40.6.5.1      40.6.5.1
FSP     9125-F2C    02C6986        B-0   40.6.5.2      40.6.5.2
FSP     9125-F2C    02C69B6        A-0   40.6.6.1      40.6.6.1
FSP     9125-F2C    02C69B6        B-0   40.6.6.2      40.6.6.2
FSP     9125-F2C    02C69D6        A-0   40.6.7.1      40.6.7.1
FSP     9125-F2C    02C69D6        B-0   40.6.7.2      40.6.7.2
FSP     9125-F2C    02C6A06        A-0   40.6.8.1      40.6.8.1
FSP     9125-F2C    02C6A06        B-0   40.6.8.2      40.6.8.2
FSP     9125-F2C    02C6A26        A-0   40.6.9.1      40.6.9.1
FSP     9125-F2C    02C6A26        B-0   40.6.9.2      40.6.9.2
HMC     7042CR5     KQZHRKK        N/A   40.0.0.231    c250hmc10(40.0.0.231)
CEC     9125-F2C    02C68B6                            f06cec01
CEC     9125-F2C    02C68D6                            f06cec02
CEC     9125-F2C    02C6906                            f06cec03
CEC     9125-F2C    02C6946                            f06cec04
CEC     9125-F2C    02C6986                            f06cec05
CEC     9125-F2C    02C69B6                            f06cec06
CEC     9125-F2C    02C69D6                            f06cec07
CEC     9125-F2C    02C6A06                            f06cec08
CEC     9125-F2C    02C6A26                            f06cec09
CEC     9125-F2C    02C6A46                            f06cec10
CEC     9125-F2C    02C6A66                            f06cec11
CEC     9125-F2C    02C6A86                            f06cec12
FRAME   78AC-100    992003H                            frame06
```

rscan

For this command, we list the help description in Figure 3-8. A typical output example is shown in Example 3-23.



```
# rscan -h
Usage: rscan <noderange> [-u][-w][-x|-z] [-V|--verbose]
       rscan [-h|--help|-v|--version]
```

Figure 3-8 rscan command flag description

The **rscan** command collects node information from one or more hardware control points. The following input and output values are used:

- Command: **rscan <noderange>**
- Flags:
  - -z: Converts the output into xCAT “stanza” format.
- Outputs:

  [no-flags]
  type name id type-model serial-number side address
  <type> <name> <id> <type-model> <serial-number> <side> <address>
- Attributes:
  - <noderange>: Nodes that are listed in xCAT database that belong to an FSP hardware type.
  - <name>: Device name type that is scanned.
  - <id>: ID for LPAR of the device that is scanned.
  - <type-model>: XXXX-YYY model type.
  - <serial-number>: IBM 7-hexadecimal serial number. The serial number associates all the LPARs belonging to a determined FSP.
  - <side>: [A | B].
  - <address>: IP for the hardware component.

Example 3-23 rscan output example

```
# rscan fsp
type    name        id      type-model  serial-number  side    address
fsp     40.10.12.1          9125-F2C    02D7695        A       40.10.12.1;
lpar    02d76951    1       9125-F2C    02D7695
lpar    02d76952    2       9125-F2C    02D7695
lpar    02d76955    5       9125-F2C    02D7695
lpar    02d76959    9       9125-F2C    02D7695
lpar    02d769513   13      9125-F2C    02D7695
lpar    02d769517   17      9125-F2C    02D7695
lpar    02d769518   18      9125-F2C    02D7695
lpar    02d769521   21      9125-F2C    02D7695
lpar    02d769525   25      9125-F2C    02D7695
lpar    02d769529   29      9125-F2C    02D7695
```

lsvm

For this command, we list the help description that is shown in Figure 3-9 on page 188. A typical output example is shown in Example 3-24 on page 189.

```
# lsvm -h
Usage:
   Common:
       lsvm <noderange> [-V|--verbose]
       lsvm [-h|--help|-v|--version]
   PPC (with HMC) specific:
       lsvm <noderange> [-a|--all]
   PPC (using Direct FSP Management) specific:
       lsvm <noderange> [-l|--long]
```

Figure 3-9 lsvm command flag description

The `lsvm` command lists the LPARs I/O slots information and CEC configuration. The following input and output values are used:

- Command: **`lsvm <noderange>`**
- **Flags**:
  - -l: Detailed output.
- Outputs:

  [no-flags]
  {
  <id_vm>: <number2>/<physical_location_slot>/<reference>/<U_size>/<id_vm>
  [...]
  }
  <noderange>: PendingPumpMode=<pending_pump_mode>,
  CurrentPumpMode=<active_pump_mode>,
  OctantCount=<installed_QPM>:
  {
  OctantID=<lpar_id>, PendingOctCfg=<pending_octant_config>,
  CurrentOctCfg=<active_octant_config>,
  PendingMemoryInterleaveMode=<pending_interleave_mode>,
  CurrentMemoryInterleaveMode=<active_interleave_mode>;
  [...]
  }
- Designations:
  - PumpMode: The pump mode value includes the following valid options:
    - 1 - Node Pump Mode
    - 2 - Chip Pump Mode
  - OctCfg: LPAR configuration per octant. The octant configuration includes the following valid options:
    - 1 - 100% to 1 LPAR
    - 2 - 50% to 2LPARs
    - 3 - 25% to 1 LPAR and 75% to 1 LPAR
    - 4 - 25% to 4 LPARs
    - 5 - 25% to 2 LPAR and 50% to 1 LPAR
  - MemoryInterleaveMode: Memory configuration setup per octant. The Memory Interleaving Mode includes the following valid options:
    - 0 - not Applicable
    - 1 - interleaved
    - 2 - non-interleaved

- Attributes:
  - <noderange>: Nodes that are listed in xCAT database that belong to a FSP/CEC hardware type.
  - <id_vm>: LPAR id such as the first lpar=1on OctantID=0, the first lpar=5 on
    OctantID=1
  - <physical_location_slot>: Physical location of PCI ports.
  - <pending_pump_mode>: Pending configuration for pump mode (effective after IPL).
  - <active_pump_mode>: Currently active pump mode.
  - <installed_QPM>: Number of QPM physically installed.
  - <pending_octant_config>: Pending octant configuration (effective after IPL).
  - <active_octant_config>: Currently active octant configuration.
  - <pending_interleave_mode>: Pending memory interleave mode (effective after IPL).
  - <active_interleave_mode>: Currently active memory interleave mode.

Example 3-24 lsvm output example

```
# lsvm fsp
1: 520/U78A9.001.1122233-P1-C14/0x21010208/2/1
1: 514/U78A9.001.1122233-P1-C17/0x21010202/2/1
1: 513/U78A9.001.1122233-P1-C15/0x21010201/2/1
1: 512/U78A9.001.1122233-P1-C16/0x21010200/2/1
13: 537/U78A9.001.1122233-P1-C9/0x21010219/2/13
13: 536/U78A9.001.1122233-P1-C10/0x21010218/2/13
13: 529/U78A9.001.1122233-P1-C11/0x21010211/2/13
13: 528/U78A9.001.1122233-P1-C12/0x21010210/2/13
17: 553/U78A9.001.1122233-P1-C5/0x21010229/2/17
17: 552/U78A9.001.1122233-P1-C6/0x21010228/2/17
17: 545/U78A9.001.1122233-P1-C7/0x21010221/2/17
17: 544/U78A9.001.1122233-P1-C8/0x21010220/2/17
2: 521/U78A9.001.1122233-P1-C13/0x21010209/2/2
29: 569/U78A9.001.1122233-P1-C1/0x21010239/0/0
29: 568/U78A9.001.1122233-P1-C2/0x21010238/0/0
29: 561/U78A9.001.1122233-P1-C3/0x21010231/0/0
29: 560/U78A9.001.1122233-P1-C4/0x21010230/0/0
40.10.12.1: PendingPumpMode=1,CurrentPumpMode=1,OctantCount=8:
OctantID=0,PendingOctCfg=5,CurrentOctCfg=5,PendingMemoryInterleaveMode=2,CurrentMe
moryInterleaveMode=2;
OctantID=1,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,CurrentMe
moryInterleaveMode=1;
OctantID=2,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,CurrentMe
moryInterleaveMode=1;
OctantID=3,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,CurrentMe
moryInterleaveMode=2;
OctantID=4,PendingOctCfg=5,CurrentOctCfg=5,PendingMemoryInterleaveMode=2,CurrentMe
moryInterleaveMode=2;
OctantID=5,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,CurrentMe
moryInterleaveMode=1;
OctantID=6,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,CurrentMe
moryInterleaveMode=1;
OctantID=7,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,CurrentMe
moryInterleaveMode=2;
```

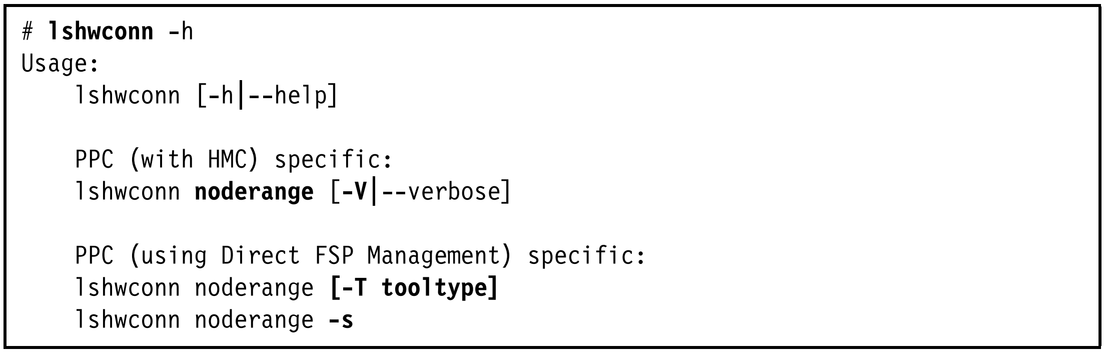
Hardware connectivity

For hardware connectivity, the following command is described:

- **lshwconn**

lshwconn

For this command, we list the help description that is shown in Figure 3-10. Typical output examples are shown in Example 3-25 on page 191, Example 3-26 on page 191, Example 3-27 on page 191, Example 3-28 on page 191, and Example 3-29 on page 192.



```
# lshwconn -h
Usage:
    lshwconn [-h|--help]

    PPC (with HMC) specific:
    lshwconn noderange [-V|--verbose]

    PPC (using Direct FSP Management) specific:
    lshwconn noderange [-T tooltype]
    lshwconn noderange -s
```

Figure 3-10 lshwconn command flag description

The **lshwconn** command displays hardware connections between the EMS and FRAME/CEC/FSP. The following input and output values are used:

- Command: **lshwconn <noderange>**
- Flags:
  - [-T tooltype]: Choose the target for communication. The accepted values for <tooltype> are <fnm> or <lpar>. Default is <lpar>.
  - -s: Displays the connection with HMC.
  - -V: Displays the time that is needed to retrieve the information.
- Outputs:

  [-T]
  <noderange>: <fsp>: sp=[primary | secondary],ipadd=<A.A.A.A>,
  alt_ipadd=<B.B.B.B>,state=[LINE UP |]
  [-T,-V]
  <noderange>: <fsp>: sp=[primary | secondary],ipadd=<A.A.A.A>,
  alt_ipadd=<B.B.B.B>,state=[LINE UP |]
  <HH:MM:SS> <X> Total Elapsed Time: <Y> sec
- Designations:
  - sp: Displays which service processor is in use.
  - state: Displays the last state of the service processor in use.
  - ipadd: IP address for side A.
  - alt_ipadd: IP address for side B.
  - state: Connection status.
- Attributes:
  - <noderange>: Nodes that are listed in xCAT database that belong to a FRAME/CEC/FSP hardware type.
  - <fsp>: xCAT FSP name that is associated to <noderage>.

– <A.A.A.A>: IP address of side A of the FSP.

– <B.B.B.B>: IP address of side B of the FSP.

– <HH:MM:SS>: Current time from machine, hour, minute, and second format.

– <Y>: Elapsed time in seconds.

Example 3-25 lshwconn output example that lists hardware connections for all CEC type hardware

```
# lshwconn cec
cec12: 40.10.12.1: sp=primary,ipadd=40.10.12.1,alt_ipadd=unavailable,state=LINE UP
```

Example 3-26 lshwconn output when listing hardware connections for all FSP type hardware

```
# lshwconn fsp
40.10.12.1: 40.10.12.1:
sp=primary,ipadd=40.10.12.1,alt_ipadd=unavailable,state=LINE UP
```

Example 3-27 lshwconn output example for a FRAME example (one 775 cluster frame)

```
# lshwconn frame |sort
frame06: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
frame06: 40.6.0.2: side=b,ipadd=40.6.0.2,alt_ipadd=unavailable,state=LINE UP
```

Example 3-28 lshwconn output example for a complete CEC example (12 775 cecs)

```
# lshwconn cec |sort
f06cec01: 40.6.1.1: sp=primary,ipadd=40.6.1.1,alt_ipadd=unavailable,state=LINE UP
f06cec01: 40.6.1.2: sp=secondary,ipadd=40.6.1.2,alt_ipadd=unavailable,state=LINE
UP
f06cec02: 40.6.2.1: sp=secondary,ipadd=40.6.2.1,alt_ipadd=unavailable,state=LINE
UP
f06cec02: 40.6.2.2: sp=primary,ipadd=40.6.2.2,alt_ipadd=unavailable,state=LINE UP
f06cec03: 40.6.3.1: sp=secondary,ipadd=40.6.3.1,alt_ipadd=unavailable,state=LINE
UP
f06cec03: 40.6.3.2: sp=primary,ipadd=40.6.3.2,alt_ipadd=unavailable,state=LINE UP
f06cec04: 40.6.4.1: sp=secondary,ipadd=40.6.4.1,alt_ipadd=unavailable,state=LINE
UP
f06cec04: 40.6.4.2: sp=primary,ipadd=40.6.4.2,alt_ipadd=unavailable,state=LINE UP
f06cec05: 40.6.5.1: sp=secondary,ipadd=40.6.5.1,alt_ipadd=unavailable,state=LINE
UP
f06cec05: 40.6.5.2: sp=primary,ipadd=40.6.5.2,alt_ipadd=unavailable,state=LINE UP
f06cec06: 40.6.6.1: sp=secondary,ipadd=40.6.6.1,alt_ipadd=unavailable,state=LINE
UP
f06cec06: 40.6.6.2: sp=primary,ipadd=40.6.6.2,alt_ipadd=unavailable,state=LINE UP
f06cec07: 40.6.7.1: sp=secondary,ipadd=40.6.7.1,alt_ipadd=unavailable,state=LINE
UP
f06cec07: 40.6.7.2: sp=primary,ipadd=40.6.7.2,alt_ipadd=unavailable,state=LINE UP
f06cec08: 40.6.8.1: sp=secondary,ipadd=40.6.8.1,alt_ipadd=unavailable,state=LINE
UP
f06cec08: 40.6.8.2: sp=primary,ipadd=40.6.8.2,alt_ipadd=unavailable,state=LINE UP
f06cec09: 40.6.9.1: sp=secondary,ipadd=40.6.9.1,alt_ipadd=unavailable,state=LINE
UP
f06cec09: 40.6.9.2: sp=primary,ipadd=40.6.9.2,alt_ipadd=unavailable,state=LINE UP
f06cec10: 40.6.10.1: sp=secondary,ipadd=40.6.10.1,alt_ipadd=unavailable,state=LINE
UP
```

```
f06cec10: 40.6.10.2: sp=primary,ipadd=40.6.10.2,alt_ipadd=unavailable,state=LINE
UP
f06cec11: 40.6.11.1: sp=secondary,ipadd=40.6.11.1,alt_ipadd=unavailable,state=LINE
UP
f06cec11: 40.6.11.2: sp=primary,ipadd=40.6.11.2,alt_ipadd=unavailable,state=LINE
UP
f06cec12: 40.6.12.1: sp=secondary,ipadd=40.6.12.1,alt_ipadd=unavailable,state=LINE
UP
f06cec12: 40.6.12.2: sp=primary,ipadd=40.6.12.2,alt_ipadd=unavailable,state=LINE
UP
```

Example 3-29 lshwconn output example for a complete FSP example (12 775 cecs)

```
# lshwconn fsp |sort
40.6.1.1: 40.6.1.1: sp=primary,ipadd=40.6.1.1,alt_ipadd=unavailable,state=LINE UP
40.6.1.2: 40.6.1.2: sp=secondary,ipadd=40.6.1.2,alt_ipadd=unavailable,state=LINE
UP
40.6.10.1: 40.6.10.1:
sp=secondary,ipadd=40.6.10.1,alt_ipadd=unavailable,state=LINE UP
40.6.10.2: 40.6.10.2: sp=primary,ipadd=40.6.10.2,alt_ipadd=unavailable,state=LINE
UP
40.6.11.1: 40.6.11.1:
sp=secondary,ipadd=40.6.11.1,alt_ipadd=unavailable,state=LINE UP
40.6.11.2: 40.6.11.2: sp=primary,ipadd=40.6.11.2,alt_ipadd=unavailable,state=LINE
UP
40.6.12.1: 40.6.12.1:
sp=secondary,ipadd=40.6.12.1,alt_ipadd=unavailable,state=LINE UP
40.6.12.2: 40.6.12.2: sp=primary,ipadd=40.6.12.2,alt_ipadd=unavailable,state=LINE
UP
40.6.2.1: 40.6.2.1: sp=secondary,ipadd=40.6.2.1,alt_ipadd=unavailable,state=LINE
UP
40.6.2.2: 40.6.2.2: sp=primary,ipadd=40.6.2.2,alt_ipadd=unavailable,state=LINE UP
40.6.3.1: 40.6.3.1: sp=secondary,ipadd=40.6.3.1,alt_ipadd=unavailable,state=LINE
UP
40.6.3.2: 40.6.3.2: sp=primary,ipadd=40.6.3.2,alt_ipadd=unavailable,state=LINE UP
40.6.4.1: 40.6.4.1: sp=secondary,ipadd=40.6.4.1,alt_ipadd=unavailable,state=LINE
UP
40.6.4.2: 40.6.4.2: sp=primary,ipadd=40.6.4.2,alt_ipadd=unavailable,state=LINE UP
40.6.5.1: 40.6.5.1: sp=secondary,ipadd=40.6.5.1,alt_ipadd=unavailable,state=LINE
UP
40.6.5.2: 40.6.5.2: sp=primary,ipadd=40.6.5.2,alt_ipadd=unavailable,state=LINE UP
40.6.6.1: 40.6.6.1: sp=secondary,ipadd=40.6.6.1,alt_ipadd=unavailable,state=LINE
UP
40.6.6.2: 40.6.6.2: sp=primary,ipadd=40.6.6.2,alt_ipadd=unavailable,state=LINE UP
40.6.7.1: 40.6.7.1: sp=secondary,ipadd=40.6.7.1,alt_ipadd=unavailable,state=LINE
UP
40.6.7.2: 40.6.7.2: sp=primary,ipadd=40.6.7.2,alt_ipadd=unavailable,state=LINE UP
40.6.8.1: 40.6.8.1: sp=secondary,ipadd=40.6.8.1,alt_ipadd=unavailable,state=LINE
UP
40.6.8.2: 40.6.8.2: sp=primary,ipadd=40.6.8.2,alt_ipadd=unavailable,state=LINE UP
40.6.9.1: 40.6.9.1: sp=secondary,ipadd=40.6.9.1,alt_ipadd=unavailable,state=LINE
UP
40.6.9.2: 40.6.9.2: sp=primary,ipadd=40.6.9.2,alt_ipadd=unavailable,state=LINE UP
```

3.1.4 DB2

In the IBM Power Systems 775 cluster setup, DB2 is used as a database engine for xCAT (with ODBC support). The commands and procedures that are used to check the status of the DB2 subsystem and their readiness to other components are shown in the following examples:

- Example 3-30
- Example 3-31
- Example 3-32
- Example 3-33 on page 194
- Example 3-34 on page 194
- Example B-2 on page 325

For more information about monitoring the DB2 subsystem, see Table 3-1 on page 158.

The following commands are described in this section:

- **db2ilist**
- **db2level**
- **db2_local_ps**
- **db2 connect to <DB2_instance>**
- **db2 get database configuration for <DB2_instance>**
- **/usr/local/bin/isql -v <DB2_instance>**

Example 3-30 db2ilist output example when DB2 instances are listed

```
# db2ilist
xcatdb
```

Example 3-31 db2level output when DB2 version details and product installation directory are listed

```
# db2level
DB21085I  Instance "xcatdb" uses "64" bits and DB2 code release "SQL09074" with
level identifier "08050107".
Informational tokens are "DB2 v9.7.0.4", "s110330", "IP23236", and Fix Pack
"4".
Product is installed at "/opt/IBM/db2/V9.7".
```

Example 3-32 db2_local_ps output example when DB2 running processes are listed

```
# db2_local_ps
Node 0
     UID        PID       PPID    C     STIME     TTY     TIME CMD
  xcatdb    6160442   13631654   21       Oct      04        - 27:29 db2sysc
    root   10616864    6160442    0       Oct      04        - 0:41 db2ckpwd
    root   10682410    6160442    0       Oct      04        - 0:41 db2ckpwd
    root   13434964    6160442    0       Oct      04        - 0:41 db2ckpwd
  xcatdb   15269970    6160442    0       Oct      04        - 0:00 db2vend (PD
Vendor Process - 258)
```

Example 3-33 Command to check the connectivity status of DB2 database instance

```
# db2 connect to xcatdb

   Database Connection Information

 Database server        = DB2/AIX64 9.7.4
 SQL authorization ID   = ROOT
 Local database alias   = XCATDB
```

Example 3-34 Command to check the connectivity status of ODBC in DB2 database instance

```
# /usr/local/bin/isql -v xcatdb
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL> quit
```

3.1.5 AIX and Linux systems

In this section, we describe commonly used commands to monitor system activity, as shown in Table 3-3. For more information, see Table 3-1 on page 158.

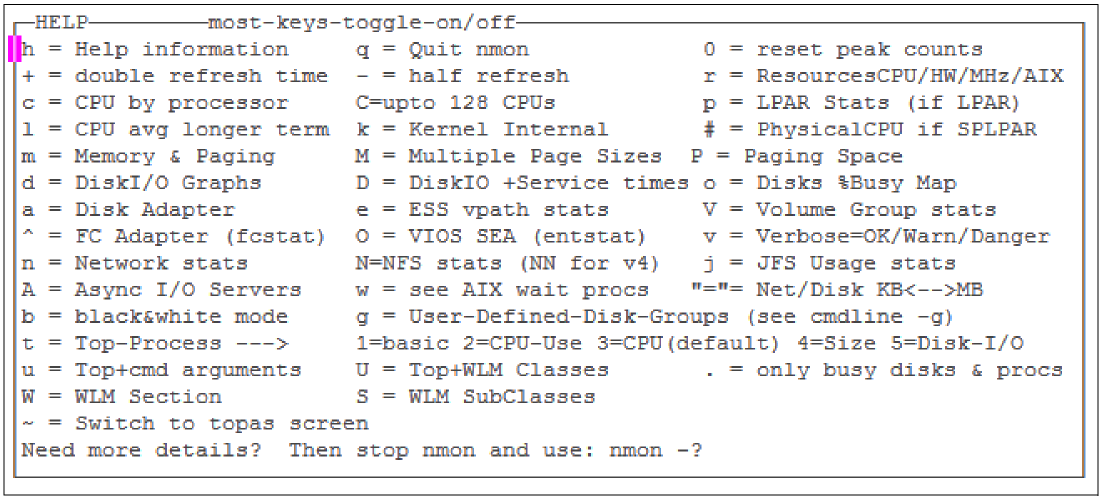
Table 3-3 AIX and Linux system monitoring commands equivalencies

AIX	Linux	Command Description
nmon	**nmon**	Nigel Monitor for General system overview.
topas	**top**	OS-specific system overview.
ps	**ps**	Displays processes status information.
alog -t console -o	-	Dumps console Log.
iostat	**iostat**	Reports IO statistics for loaded devices (physical and logical).
istat	-	Displays i-node information for a particular file.
-	stat	Displays file or file system status information.
mount	**mount**	Lists mounted file systems.
vmstat	**vmstat**	Reports virtual memory statistics.
pstat	-	Interprets and displays the contents of systems tables: process, kernel thread, file, i-node, swap, processor, tty structures, and variables.
netstat	**netstat**	Displays logical network routing information, status, and statistics.
entstat	**ethtool**	Displays physical and logical network status and statistics.

> **Important:** Although some commands feature the same string and purpose, their internal arguments and flags might differ in AIX and Linux.

NMON Tool

This tool is used for general system monitoring and it is supported in AIX and Linux. You check all of its options by pressing **H**, as shown in Figure 3-11.

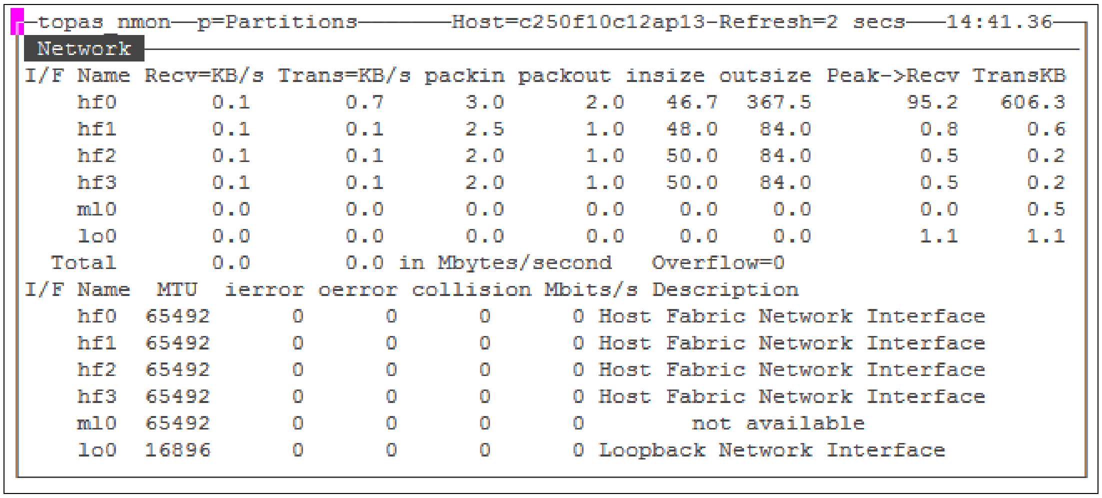


```
HELP            most-keys-toggle-on/off
h = Help information     q = Quit nmon              0 = reset peak counts
+ = double refresh time  - = half refresh           r = ResourcesCPU/HW/MHz/AIX
c = CPU by processor     C=upto 128 CPUs            p = LPAR Stats (if LPAR)
l = CPU avg longer term  k = Kernel Internal        # = PhysicalCPU if SPLPAR
m = Memory & Paging      M = Multiple Page Sizes  P = Paging Space
d = DiskI/O Graphs       D = DiskIO +Service times o = Disks %Busy Map
a = Disk Adapter         e = ESS vpath stats        V = Volume Group stats
^ = FC Adapter (fcstat)  O = VIOS SEA (entstat)     v = Verbose=OK/Warn/Danger
n = Network stats        N=NFS stats (NN for v4)    j = JFS Usage stats
A = Async I/O Servers    w = see AIX wait procs    "="= Net/Disk KB<-->MB
b = black&white mode     g = User-Defined-Disk-Groups (see cmdline -g)
t = Top-Process --->     1=basic 2=CPU-Use 3=CPU(default) 4=Size 5=Disk-I/O
u = Top+cmd arguments    U = Top+WLM Classes        . = only busy disks & procs
W = WLM Section          S = WLM SubClasses
~ = Switch to topas screen
Need more details?  Then stop nmon and use: nmon -?
```

Figure 3-11   NMON help panel (AIX panel)

AIX

NMON views Hot Fabric Interface (HFI) logical networks for IP communication only in AIX, as shown in Figure 3-12.

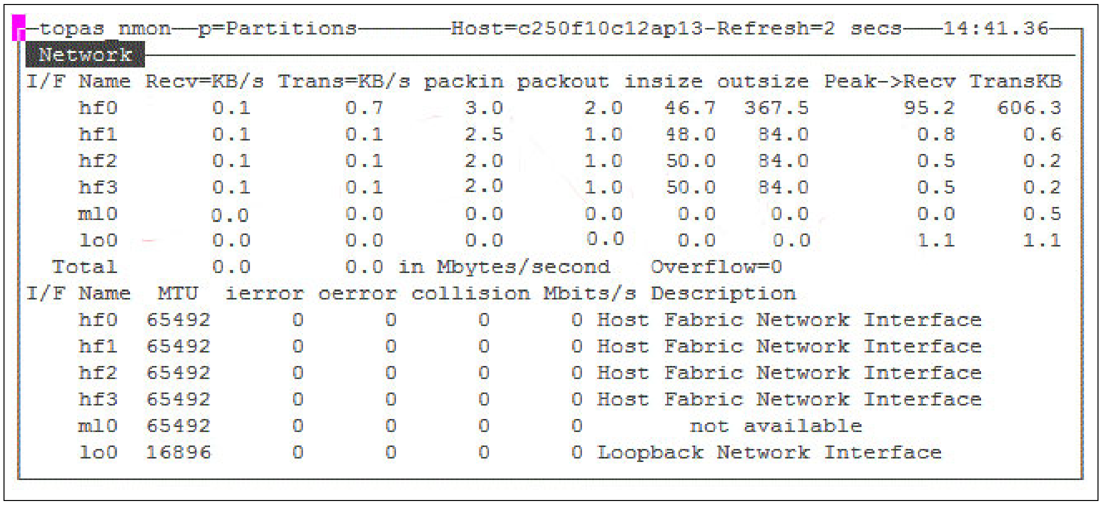


```
topas_nmon--p=Partitions--------Host=c250f10c12ap13-Refresh=2 secs---14:41.36
 Network
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv TransKB
    hf0       0.1       0.7      3.0      2.0   46.7  367.5       95.2   606.3
    hf1       0.1       0.1      2.5      1.0   48.0   84.0        0.8     0.6
    hf2       0.1       0.1      2.0      1.0   50.0   84.0        0.5     0.2
    hf3       0.1       0.1      2.0      1.0   50.0   84.0        0.5     0.2
    ml0       0.0       0.0      0.0      0.0    0.0    0.0        0.0     0.5
    lo0       0.0       0.0      0.0      0.0    0.0    0.0        1.1     1.1
  Total       0.0       0.0 in Mbytes/second   Overflow=0
I/F Name  MTU  ierror oerror collision Mbits/s Description
    hf0  65492      0      0      0       0 Host Fabric Network Interface
    hf1  65492      0      0      0       0 Host Fabric Network Interface
    hf2  65492      0      0      0       0 Host Fabric Network Interface
    hf3  65492      0      0      0       0 Host Fabric Network Interface
    ml0  65492      0      0      0       0        not available
    lo0  16896      0      0      0       0 Loopback Network Interface
```

Figure 3-12   NMON example for IP over HFI monitoring in AIX

Linux

NMON views HFI logical networks for IP communication only in Linux (Red Hat), as shown in Figure 3-13 on page 196.



```
-topas nmon--p=Partitions---------Host=c250f10c12ap13-Refresh=2 secs---14:41.36--
 Network ---------------------------------------------------------------------------
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv TransKB
    hf0       0.1        0.7      3.0      2.0   46.7  367.5       95.2    606.3
    hf1       0.1        0.1      2.5      1.0   48.0   84.0        0.8      0.6
    hf2       0.1        0.1      2.0      1.0   50.0   84.0        0.5      0.2
    hf3       0.1        0.1      2.0      1.0   50.0   84.0        0.5      0.2
    ml0       0.0        0.0      0.0      0.0    0.0    0.0        0.0      0.5
    lo0       0.0        0.0      0.0      0.0    0.0    0.0        1.1      1.1
  Total       0.0        0.0 in Mbytes/second   Overflow=0
I/F Name  MTU  ierror oerror collision Mbits/s Description
    hf0  65492      0      0        0       0 Host Fabric Network Interface
    hf1  65492      0      0        0       0 Host Fabric Network Interface
    hf2  65492      0      0        0       0 Host Fabric Network Interface
    hf3  65492      0      0        0       0 Host Fabric Network Interface
    ml0  65492      0      0        0       0         not available
    lo0  16896      0      0        0       0 Loopback Network Interface
```

Figure 3-13 NMON example for IP over HFI monitoring in Linux (Red Hat)

AIX Commands

As shown in Example 3-35, Example 3-36 on page 197, Example 3-37 on page 197, Example 3-38 on page 197, and Example 3-39 on page 198, for the following commands, we show output data that helps in identifying problems or specific debugging cases:

- **`iostat`**
- **`mount`**
- **`vmstat`**
- **`netstat`**
- **`entstat`**

Example 3-35 iostat output example

```
# iostat

System configuration: lcpu=32 drives=4 paths=3 vdisks=1

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait
          0.2         21.7                0.1    0.2   99.6      0.1

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk0           0.8      11.6       1.9    1056751  24978277
hdisk1           0.8      11.3       1.8     179528  24986205
cd0              0.0       2.9       0.1    6387824         0
hdisk2           3.3     114.5       2.7  162761054  93333336
```

Example 3-36 mount output example

```
# mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs2   Sep 30 15:43 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs2   Sep 30 15:43 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs2   Sep 30 15:43 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs2   Sep 30 15:43 rw,log=/dev/hd8
         /dev/hd1         /home            jfs2   Sep 30 15:44 rw,log=/dev/hd8
         /dev/hd11admin   /admin           jfs2   Sep 30 15:44 rw,log=/dev/hd8
         /proc            /proc            procfs Sep 30 15:44 rw
         /dev/hd10opt     /opt             jfs2   Sep 30 15:44 rw,log=/dev/hd8
         /dev/livedump    /var/adm/ras/livedump jfs2   Sep 30 15:44
rw,log=/dev/hd8
         /dev/xcatlv      /install         jfs2   Oct 04 05:39 rw,log=/dev/loglv00
         /dev/db2lv       /db2database     jfs2   Oct 04 10:58 rw,log=/dev/loglv00
         /dev/mntdb2lv    /mntdb2          jfs2   Oct 04 10:59 rw,log=/dev/loglv00
```

Example 3-37 vmstat output example

```
# vmstat

System configuration: lcpu=32 mem=31616MB

kthr    memory              page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 1  1 1595797 1004966   0   0   0   3    6   0  13 1919 460  0  0 99  0
```

Example 3-38 netstat output example

```
# netstat -ni
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      0.21.5e.8a.b3.fa 14398127     0  4879323     3     0
en0   1500  10          10.0.0.103       14398127     0  4879323     3     0
en2   1500  link#3      0.21.5e.8a.ad.4c  2870925     0   257771     4     0
en2   1500  40          40.0.0.103        2870925     0   257771     4     0
en4   1500  link#4      0.21.5e.ad.16.80  6605484     0 18745290     0     0
en4   1500  192.168.0   192.168.0.103     6605484     0 18745290     0     0
lo0   16896 link#1                        3339828     0  3339822     0     0
lo0   16896 127         127.0.0.1         3339828     0  3339822     0     0
lo0   16896 ::1%1                         3339828     0  3339822     0     0
```

Example 3-39   entstat output example

```
# entstat en0
-------------------------------------------------------------
ETHERNET STATISTICS (en0) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:b3:fa
Elapsed Time: 25 days 21 hours 8 minutes 8 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 4879540                              Packets: 14398719
Bytes: 551682847                              Bytes: 15732347906
Interrupts: 0                                 Interrupts: 5370419
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 5
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Broadcast Packets: 10314                      Broadcast Packets: 2980396
Multicast Packets: 10605                      Multicast Packets: 3028
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 0

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 0
Adapter Data Rate: 2000
Driver Flags: Up Broadcast Running
        Simplex 64BitSupport ChecksumOffload
        LargeSend DataRateSet
```

3.1.6 Integrated Switch Network Manager

In this section, a new set of commands specific for the IBM Power Systems 775 cluster are described. The Integrated Switch Network Manager (ISNM) integrates the Cluster Network Manager (CNM), and the hardware server daemons. For more information, see Table 3-1 on page 158.

The following commands are described in this section:

- **lsnwcomponents**
- **lsnwdownhw**
- **lsnwexpnbrs**
- **lsnwgc**
- **lsnwlinkinfo**
- **lsnwloc**
- **lsnwmiswire**
- **lsnwtopo**

lsnwcomponents

For this command, we list the help description that is shown in Figure 3-14. A typical output is shown in Example 3-40 on page 200.

```
# lsnwcomponents -h

Usage:  lsnwcomponents [ -B | --BPA | -F | --FSP ] [ -p | --page ] [ -h |
--help ]

The options can be given either with '-' or '--'.
```

*Figure 3-14 **lsnwcomponents** command flag description*

The **lsnwcomponents** command lists integrated switch network components. The following input and output values are used:

- Command: **lsnwcomponents**
- Flags:
  - -B: Filters BPA output only
  - -F: Filters FSP output only
- Outputs:

  [no-flags]
  [BPA | FSP] [Primary | Backup] ip=<ip_active> MTMS= <model>-<type>*<serial_number>
  <active_service_processor>
- Designations:
  - ip: Hardware component IP
  - MTMS: Hardware part, Model, Type and Serial Number
- Attributes:
  - <ip_active>: X.X.X.X (IP)
  - <model>: Model number
  - <type>: Type 3 hexadecimal code
  - <serial_number>: Seven hexadecimal code

Example 3-40 lsnwcomponents output example for a single 775 CEC

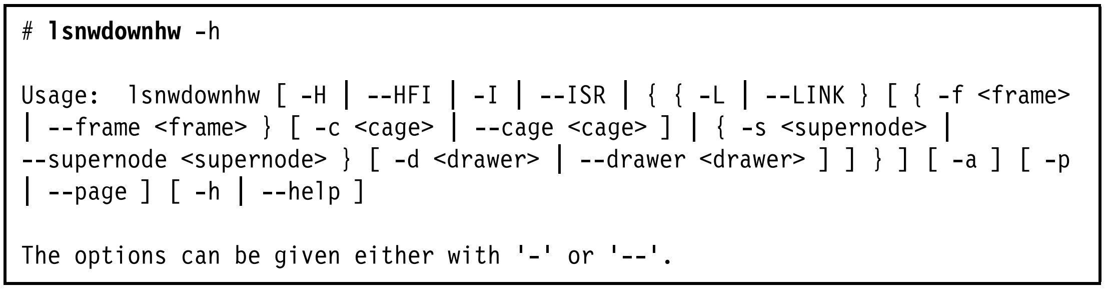
```
# lsnwcomponents
FSP Primary ip=40.10.12.1 MTMS=9125-F2C*02D7695 FR010-CG14-SN008-DR3
```

Example 3-41 lsnwcomponents output example for twelve 775 CEC on a single FRAME

```
lsnwcomponents |sort
BPA Backup  ip=40.6.0.2 MTMS=78AC-100*992003H FR006
BPA Primary ip=40.6.0.1 MTMS=78AC-100*992003H FR006
FSP Backup  ip=40.6.1.2 MTMS=9125-F2C*02C68B6
FSP Backup  ip=40.6.10.1 MTMS=9125-F2C*02C6A46
FSP Backup  ip=40.6.11.1 MTMS=9125-F2C*02C6A66
FSP Backup  ip=40.6.12.1 MTMS=9125-F2C*02C6A86
FSP Backup  ip=40.6.2.1 MTMS=9125-F2C*02C68D6
FSP Backup  ip=40.6.3.1 MTMS=9125-F2C*02C6906
FSP Backup  ip=40.6.4.1 MTMS=9125-F2C*02C6946
FSP Backup  ip=40.6.5.1 MTMS=9125-F2C*02C6986
FSP Backup  ip=40.6.6.1 MTMS=9125-F2C*02C69B6
FSP Backup  ip=40.6.7.1 MTMS=9125-F2C*02C69D6
FSP Backup  ip=40.6.8.1 MTMS=9125-F2C*02C6A06
FSP Backup  ip=40.6.9.1 MTMS=9125-F2C*02C6A26
FSP Primary ip=40.6.1.1 MTMS=9125-F2C*02C68B6 FR006-CG03-SN051-DR0
FSP Primary ip=40.6.10.2 MTMS=9125-F2C*02C6A46 FR006-CG12-SN005-DR1
FSP Primary ip=40.6.11.2 MTMS=9125-F2C*02C6A66 FR006-CG13-SN005-DR2
FSP Primary ip=40.6.12.2 MTMS=9125-F2C*02C6A86 FR006-CG14-SN005-DR3
FSP Primary ip=40.6.2.2 MTMS=9125-F2C*02C68D6 FR006-CG04-SN051-DR1
FSP Primary ip=40.6.3.2 MTMS=9125-F2C*02C6906 FR006-CG05-SN051-DR2
FSP Primary ip=40.6.4.2 MTMS=9125-F2C*02C6946 FR006-CG06-SN051-DR3
FSP Primary ip=40.6.5.2 MTMS=9125-F2C*02C6986 FR006-CG07-SN004-DR0
FSP Primary ip=40.6.6.2 MTMS=9125-F2C*02C69B6 FR006-CG08-SN004-DR1
FSP Primary ip=40.6.7.2 MTMS=9125-F2C*02C69D6 FR006-CG09-SN004-DR2
FSP Primary ip=40.6.8.2 MTMS=9125-F2C*02C6A06 FR006-CG10-SN004-DR3
FSP Primary ip=40.6.9.2 MTMS=9125-F2C*02C6A26 FR006-CG11-SN005-DR0
```

lsnwdownhw

For this command, we list the help description that is shown in Figure 3-15. Typical output is shown in Example 3-42 on page 201.



```
# lsnwdownhw -h

Usage:  lsnwdownhw [ -H | --HFI | -I | --ISR | { { -L | --LINK } [ { -f <frame>
| --frame <frame> } [ -c <cage> | --cage <cage> ] | { -s <supernode> |
--supernode <supernode> } [ -d <drawer> | --drawer <drawer> ] ] } ] [ -a ] [ -p
| --page ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-15 lsnwdownhw command flag description

The **lsnwdownhw** command lists faulty hardware in the network as, D and L links, HFIs, and ISRs. The input and output values are:

- Command: **lsnwdownhw**

- Flags:
  - -H: Filters HFI output only
  - -I: Filters ISR output only
  - -L: Filters D and L LINKs output only
- Outputs:

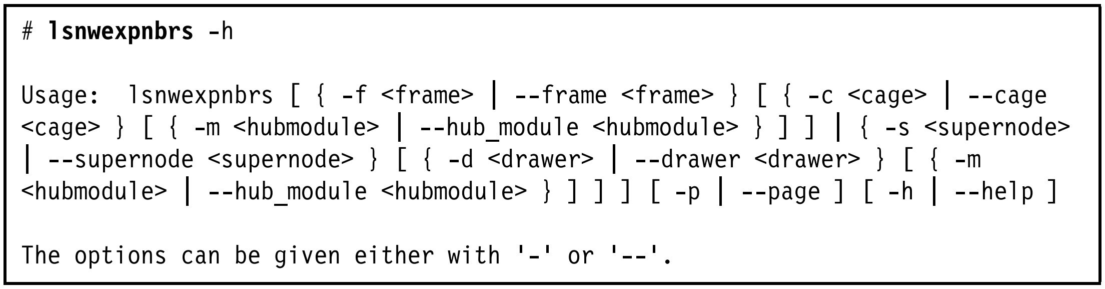
  [no-flags]
  Link <connection_physical_location> <status> Service_Location: <physical_location>
- Designations:
  - Service_Location: Area to service, location point of view.
- Attributes:
  - <connection_physical_location>: Physical link location.
  - <status>: Hardware status for the detected link on failure.
  - <physical_location>: Physical location within the machine where service is needed to correct the link failure.

Example 3-42 lsnwdownhw command output

```
# lsnwdownhw
Link     FR006-CG08-SN004-DR1-HB0-LR20   DOWN_FAULTY
Service_Location: U78A9.001.99200FX-P1-T9
Link     FR006-CG08-SN004-DR1-HB0-LR21   DOWN_FAULTY
Service_Location: U78A9.001.99200FX-P1-T9
Link     FR006-CG08-SN004-DR1-HB0-LR22   DOWN_FAULTY
Service_Location: U78A9.001.99200FX-P1-T9
Link     FR006-CG08-SN004-DR1-HB0-LR23   DOWN_FAULTY
Service_Location: U78A9.001.99200FX-P1-T9
Link     FR006-CG10-SN004-DR3-HB0-LR17   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG10-SN004-DR3-HB1-LR17   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG10-SN004-DR3-HB2-LR17   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG10-SN004-DR3-HB3-LR17   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG07-SN004-DR0-HB1-LR16   DOWN_FAULTY
Service_Location: U78A9.001.99200HM-P1-T9
Link     FR006-CG07-SN004-DR0-HB1-LR17   DOWN_FAULTY
Service_Location: U78A9.001.99200HM-P1-T9
Link     FR006-CG07-SN004-DR0-HB1-LR18   DOWN_FAULTY
Service_Location: U78A9.001.99200HM-P1-T9
Link     FR006-CG07-SN004-DR0-HB1-LR19   DOWN_FAULTY
Service_Location: U78A9.001.99200HM-P1-T9
Link     FR006-CG09-SN004-DR2-HB4-LR16   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG09-SN004-DR2-HB5-LR16   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG09-SN004-DR2-HB6-LR16   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
Link     FR006-CG09-SN004-DR2-HB7-LR16   DOWN_FAULTY
Service_Location: U78A9.001.#######-P1-T9
```

lsnwexpnbrs

For this command, we list the help description that is shown in Figure 3-16 on page 202. A typical output is shown in Example 3-43 on page 202.



```
# lsnwexpnbrs -h

Usage:  lsnwexpnbrs [ { -f <frame> | --frame <frame> } [ { -c <cage> | --cage
<cage> } [ { -m <hubmodule> | --hub_module <hubmodule> } ] ] | { -s <supernode>
| --supernode <supernode> } [ { -d <drawer> | --drawer <drawer> } [ { -m
<hubmodule> | --hub_module <hubmodule> } ] ] ] [ -p | --page ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-16 lsnwexpnbrs command flag description

The **lsnwexpnbrs** command lists integrated switch network expected neighbors (D and L links). The following input and output values are used:

- Command: **lsnwexpnbrs**
- Flags:
  - -f, -c, -m, -s, -d: Filter results by, frame, cage, supernode, drawer, and hub module.
- Outputs:
  [no-flags]
  Loc: <physical_location_initial_link_endpoint> ExpNbr:
  <expected_physical_location_final_link_endpoint> ActualNbr:
  <actual_physical_location_link_endpoint>
- Designations:
  - LINK <physical_location>: FRAME-CAGE-SUPERNODE-DRAWER-HUB-LINK
- Attributes:
  - <physical_location_initial_link_endpoint>: “Endpoint one” of the link connection.
  - <expected_physical_location_final_link_endpoint>: Expected “endpoint two” of the same link connection.
  - <actual_physical_location_link_endpoint>: Actual configured “endpoint two” of the same link connection.

Example 3-43 lsnwexpnbrs command output

```
# lsnwexpnbrs
Loc: FR006-CG04-SN051-DR1-HB0-LL0 ExpNbr: FR006-CG04-SN051-DR1-HB3-LL0 ActualNbr:
FR006-CG04-SN051-DR1-HB3-LL0
Loc: FR006-CG04-SN051-DR1-HB0-LL1 ExpNbr: FR006-CG04-SN051-DR1-HB5-LL0 ActualNbr:
FR006-CG04-SN051-DR1-HB5-LL0
Loc: FR006-CG04-SN051-DR1-HB0-LL2 ExpNbr: FR006-CG04-SN051-DR1-HB1-LL2 ActualNbr:
FR006-CG04-SN051-DR1-HB1-LL2
[...]
```

lsnwgc

For this command, we list the help description that is shown in Figure 3-17 on page 203. A typical output is shown in Example 3-44 on page 204.

```
# lsnwgc  -h

Usage:
  lsnwgc [ -a  | -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-17 lsnwgc command flag description

The **lsnwgc** command lists integrated switch network global counter information. The following input and output values are used:

- Command: **lsnwgc -a**
- Flags:
  - -a: Lists the global counter parameters and global counter master and backups.
- Outputs:

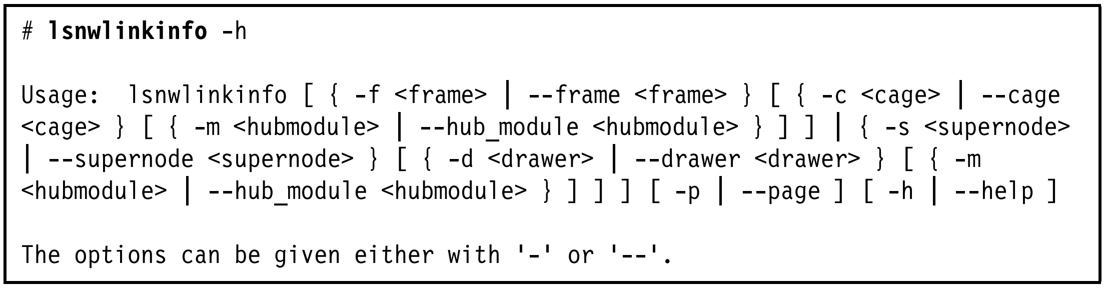
  [-a]
  Master: <FRAME>-<CAGE>-<SUPERNODE>-<DRAWER>-<HUB> Counter Id:
  <node_id>
  No. of Configured Backups: <number_backup_nodes>
  {
  <FRAME>-<CAGE>-<SUPERNODE>-<DRAWER>-<HUB>
  [...]
  }
  Broadcast Frequency (in 2 Ghz counter cycles) - <broadcast_freq>
  Takeover Trigger (Number of Broadcasts) - <takeover_trigger>
  Invalid Trigger (Number of Broadcasts) - <invalid_triger>
  Range Constant (Added when a backup is stale) - <range_constant>
- Designations:
  - <FRAME>: Frame number (FRXXX)
  - <CAGE>: Cage number (CGXX)
  - <SUPERNODE>: Supernode number (SNXXX)
  - <DRAWER>: Drawer number (DR[0-3])
- Attributes:
  - <number_backup_nodes>: Number of nodes that assume the role of Master if this node goes down.

Example 3-44 lsnwgc command output

```
# lsnwgc -a
Master: FR006-CG06-SN051-DR3-HB1 Counter Id: 0xd0
No. of Configured Backups: 11
FR006-CG12-SN005-DR1-HB1
FR006-CG04-SN051-DR1-HB1
FR006-CG13-SN005-DR2-HB1
FR006-CG05-SN051-DR2-HB1
FR006-CG10-SN004-DR3-HB1
FR006-CG08-SN004-DR1-HB1
FR006-CG07-SN004-DR0-HB1
FR006-CG09-SN004-DR2-HB1
FR006-CG11-SN005-DR0-HB1
FR006-CG14-SN005-DR3-HB1
FR006-CG03-SN051-DR0-HB1

Broadcast Frequency (in 2Ghz counter cycles) - 0xffffff
Takeover Trigger (Number of Broadcasts) - 0x10
Invalid Trigger (Number of Broadcasts) - 0xff
Range Constant (Added when a backup is stale) - 0x1000
```

lsnwlinkinfo

For this command, we list the help description that is shown in Figure 3-18. A typical output is shown in Example 3-45.



```
# lsnwlinkinfo -h

Usage:  lsnwlinkinfo [ { -f <frame> | --frame <frame> } [ { -c <cage> | --cage
<cage> } [ { -m <hubmodule> | --hub_module <hubmodule> } ] ] | { -s <supernode>
| --supernode <supernode> } [ { -d <drawer> | --drawer <drawer> } [ { -m
<hubmodule> | --hub_module <hubmodule> } ] ] ] [ -p | --page ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-18 lsnwlinkinfo command flag description

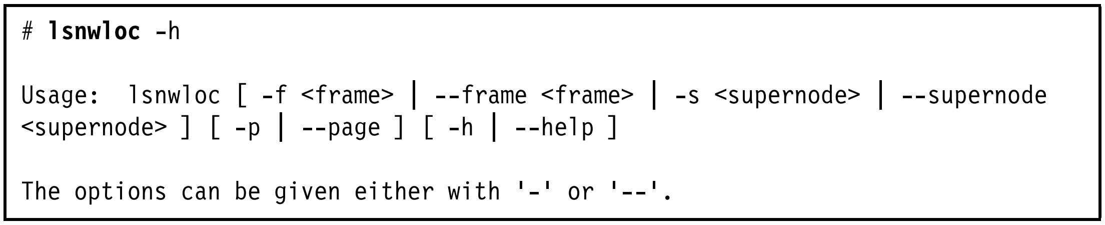
The **lsnwlinkinfo** command lists the integrated switch network links information. The input and output values are the same as the values from the **lsnwexpnbrs** command. The **lsnwlinkinfo** command also displays the status of the link, as shown in Example 3-45. The value of the status attribute is: [UP_OPERATIONAL]

Example 3-45 lsnwlinkinfo command output

```
# lsnwlinkinfo
FR006-CG04-SN051-DR1-HB0-LL0 UP_OPERATIONAL ExpNbr: FR006-CG04-SN051-DR1-HB3-LL0
ActualNbr: FR006-CG04-SN051-DR1-HB3-LL0
FR006-CG04-SN051-DR1-HB0-LL1 UP_OPERATIONAL ExpNbr: FR006-CG04-SN051-DR1-HB5-LL0
ActualNbr: FR006-CG04-SN051-DR1-HB5-LL0
FR006-CG04-SN051-DR1-HB0-LL2 UP_OPERATIONAL ExpNbr: FR006-CG04-SN051-DR1-HB1-LL2
ActualNbr: FR006-CG04-SN051-DR1-HB1-LL2
[...]
```

lsnwloc

For this command, we list the help description that is shown in Figure 3-19. A typical output is shown in Example 3-46.



```
# lsnwloc -h

Usage:  lsnwloc [ -f <frame> | --frame <frame> | -s <supernode> | --supernode
<supernode> ] [ -p | --page ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-19 lsnwloc command flag description

The **lsnwloc** command list integrated switch network locations. The following input and output values are used:

- Command: **lsnwloc**
- Flags:
  - -f, -s: Filters output by FRAME or SUPERNODE types.
- Outputs:

  [no-flags]
  `<LINK_loc> [ STANDBY | RUNTIME ]`
- Designations:
  - <LINK_loc>: FRAME-CAGE-SUPERNODE-DRAWER
  - <FRAME>: Frame number (FRXXX)
  - <CAGE>: Cage number (CGXX)
  - <SUPERNODE>: Supernode number (SNXXX)
  - <DRAWER>: Drawer number (DR[0-3])

Example 3-46 lsnwloc command output

```
FR006-CG13-SN005-DR2 STANDBY
FR006-CG05-SN051-DR2 STANDBY
FR006-CG09-SN004-DR2 STANDBY
FR006-CG07-SN004-DR0 STANDBY
FR006-CG08-SN004-DR1 STANDBY
FR006-CG06-SN051-DR3 STANDBY
FR006-CG14-SN005-DR3 STANDBY
FR006-CG03-SN051-DR0 STANDBY
FR006-CG10-SN004-DR3 STANDBY
FR006-CG04-SN051-DR1 RUNTIME
FR006-CG12-SN005-DR1 STANDBY
FR006-CG11-SN005-DR0 STANDBY
```

lsnwmiswire

For this command, we list the help description shown in Figure 3-20. A typical output is shown in Figure 3-21.

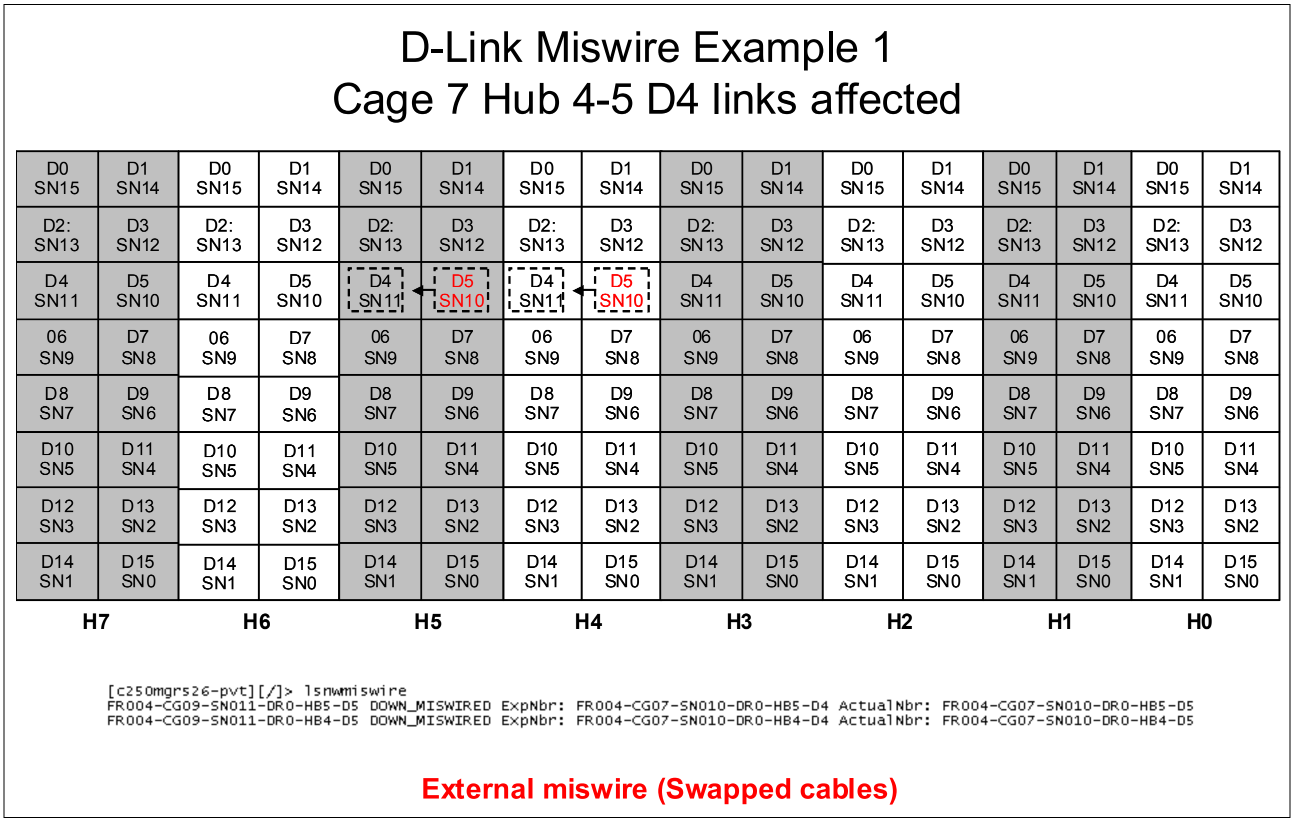
```
# lsnwmiswire -h

Usage:  lsnwmiswire [ { -f <frame> | --frame <frame> } [ -c <cage> | --cage
<cage> ] | { -s <supernode> | --supernode <supernode> } [ -d <drawer> |
--drawer <drawer> ] ] [ -p | --page ] [ -a ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-20   lsnwmiswire command flag description

The **lsnwmiswire** command list miswired links. The following input and output values are used:

- ***Command***: **lsnwmiswire**
- **Flag**: -f, -c, -s, -d: Filters output by FRAME, CAGE, SUPERNODE or DRAWER types
- Outputs:

  [no-flags]
  <syntax1>: <syntax2>: set1=[option1 | option2],set2=<A.A.A.A>,
  set3=<B.B.B.B>,set4=[option1 | option2 | option3]



D-Link Miswire Example 1
Cage 7 Hub 4-5 D4 links affected

D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		H6		H5		H4		H3		H2		H1		H0	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG09-SN011-DR0-HB5-D5 DOWN_MISWIRED ExpNbr: FR004-CG07-SN010-DR0-HB5-D4 ActualNbr: FR004-CG07-SN010-DR0-HB5-D5
FR004-CG09-SN011-DR0-HB4-D5 DOWN_MISWIRED ExpNbr: FR004-CG07-SN010-DR0-HB4-D4 ActualNbr: FR004-CG07-SN010-DR0-HB4-D5
```

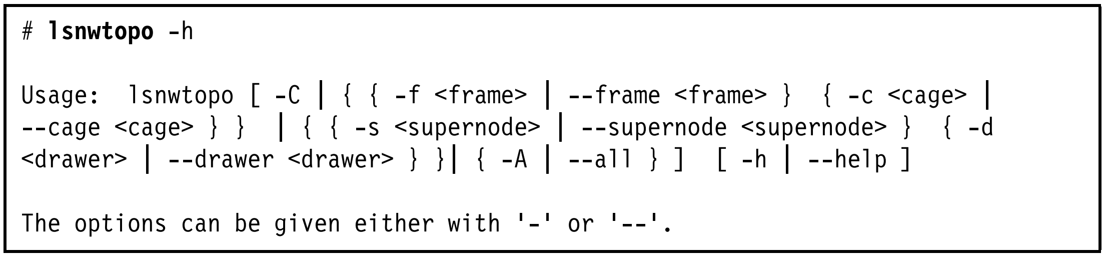
External miswire (Swapped cables)

Figure 3-21   lsnwmiswire command output

More miswired examples figures are show in “The lsnwmiswire command” on page 328.

lsnwtopo

For this command, we list the help description shown in Figure 3-22. A typical output is shown in Example 3-47.



```
# lsnwtopo -h

Usage:  lsnwtopo [ -C | { { -f <frame> | --frame <frame> }  { -c <cage> |
--cage <cage> } }  | { { -s <supernode> | --supernode <supernode> }  { -d
<drawer> | --drawer <drawer> } }| { -A | --all } ]  [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-22   lsnwtopo command flag description

The **lsnwtopo** command displays cluster topology information that is used by the network management software. The following input and output values are used:

- Command: **lsnwtopo**
- Outputs:

  [no-flags]
  ISR network topology that is specified by cluster configuration data is <topology_scheme>
- Attributes:
  - <topology_scheme>: Topology code that represents a predefined topology for 775 Clusters. For more information, see "Network topology" on page 35. This scheme represents X links of type D links. As a result, this attribute is represented as XD.

Example 3-47   lsnwtopo command output

```
# lsnwtopo
ISR network topology specified by cluster configuration data is 128D
```

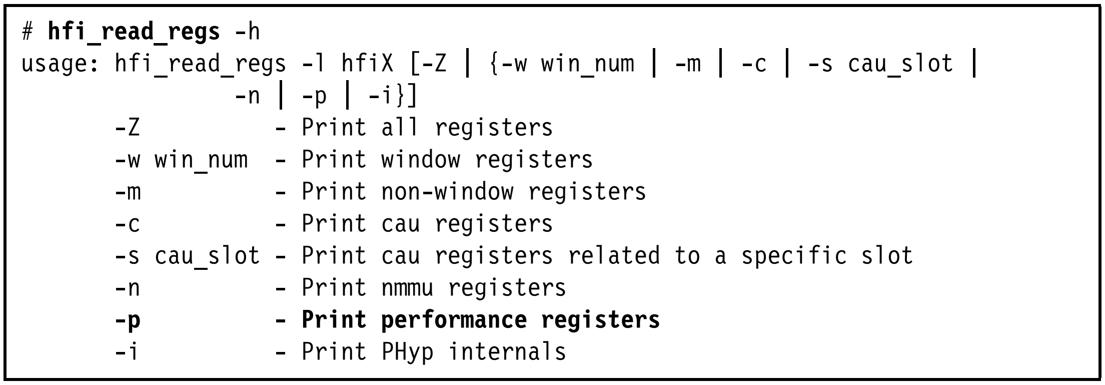
3.1.7 HFI

In this section, we discuss how to monitor the HFI. For more information, see Table 3-1 on page 158.

The hfi_read_regs syntax is used in the command.

hfi_read_regs

For this command, we list the help description that is shown in Figure 3-23 on page 208. Typical outputs are shown in Example 3-48 on page 208, and Example 3-49 on page 210.



```
# hfi_read_regs -h
usage: hfi_read_regs -l hfiX [-Z | {-w win_num | -m | -c | -s cau_slot |
                -n | -p | -i}]
       -Z         - Print all registers
       -w win_num  - Print window registers
       -m         - Print non-window registers
       -c         - Print cau registers
       -s cau_slot - Print cau registers related to a specific slot
       -n         - Print nmmu registers
       -p         - Print performance registers
       -i         - Print PHyp internals
```

Figure 3-23   hfi_read_regs command flag description

The **hfi_read_regs** command reads HFI logical devices registers. The following input and output values are used:

- Command: **hfi_read_regs -l <hfiX>**
- Flags:
  - -p: prints performance registers.
- Outputs:

  [only -l, required]
  For more information, see Example 3-48.
  [-p]
  For more information, see Example 3-49 on page 210.

Example 3-48   hfi_read_regs output example for “hfi0” link

```
hfi_read_regs -l hfi0

Nonwindow Registers
!!!================

   Nonwindow General
      number_of_windows(0:8) . . . . . 0x100 . . . . . .  [256]
      isr_id                           0x0000000000000000 [0]
      rcxt_cache_window_flush_req(0:9) 0x0 . . . . . . .  [0]
         RCWFR.win flush busy(0:0)     0x0                Inactive
         RCWFR.reserved(1:7) . . . . . 0x0 . . . . . . .  [0]
         RCWFR.window id(8:15)         0x0                [0]

   Nonwindow Page Migration
      page_migration_regs[0] . . . . . 0x0000000000000000 [0]
         PMR0.valid(0:0)               0x0                Invalid
         PMR0.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR0.new real addr(18:51)     0x0                [0]
         PMR0.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR0.read target(57:57)       0x0                Old
         PMR0.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[1]           0x0000000000000000 [0]
         PMR1.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR1.reserved1(1:17)          0x0                [0]
         PMR1.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR1.reserved2(52:56)         0x0                [0]
         PMR1.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR1.page size(58:63)         0x0                Reserved
      page_migration_regs[2] . . . . . 0x0000000000000000 [0]
         PMR2.valid(0:0)               0x0                Invalid
         PMR2.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR2.new real addr(18:51)     0x0                [0]
         PMR2.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR2.read target(57:57)       0x0                Old
         PMR2.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[3]           0x0000000000000000 [0]
         PMR3.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR3.reserved1(1:17)          0x0                [0]
         PMR3.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR3.reserved2(52:56)         0x0                [0]
         PMR3.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR3.page size(58:63)         0x0                Reserved
      page_migration_regs[4] . . . . . 0x0000000000000000 [0]
         PMR4.valid(0:0)               0x0                Invalid
         PMR4.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR4.new real addr(18:51)     0x0                [0]
         PMR4.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR4.read target(57:57)       0x0                Old
         PMR4.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[5]           0x0000000000000000 [0]
         PMR5.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR5.reserved1(1:17)          0x0                [0]
         PMR5.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR5.reserved2(52:56)         0x0                [0]
         PMR5.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR5.page size(58:63)         0x0                Reserved
      page_migration_regs[6] . . . . . 0x0000000000000000 [0]
         PMR6.valid(0:0)               0x0                Invalid
         PMR6.reserved1(1:17). . . . . 0x0 . . . . . . .  [0]
         PMR6.new real addr(18:51)     0x0                [0]
         PMR6.reserved2(52:56) . . . . 0x0 . . . . . . .  [0]
         PMR6.read target(57:57)       0x0                Old
         PMR6.page size(58:63) . . . . 0x0 . . . . . . .  Reserved
      page_migration_regs[7]           0x0000000000000000 [0]
         PMR7.valid(0:0) . . . . . . . 0x0 . . . . . . .  Invalid
         PMR7.reserved1(1:17)          0x0                [0]
         PMR7.new real addr(18:51) . . 0x0 . . . . . . .  [0]
         PMR7.reserved2(52:56)         0x0                [0]
         PMR7.read target(57:57) . . . 0x0 . . . . . . .  Old
         PMR7.page size(58:63)         0x0                Reserved

   Nonwindow Page Migration Reservation
      page_migration_reservation[0]. . 0x0000000000000000 [0]
         PMRs0.offset(48:56)           0x0                [0]
         PMRs0.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[1]    0x0000000000000000 [0]
         PMRs1.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs1.reservatn(63:63)        0x0                False
      page_migration_reservation[2]. . 0x0000000000000000 [0]
         PMRs2.offset(48:56)           0x0                [0]
         PMRs2.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[3]    0x0000000000000000 [0]
         PMRs3.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs3.reservatn(63:63)        0x0                False
      page_migration_reservation[4]. . 0x0000000000000000 [0]
         PMRs4.offset(48:56)           0x0                [0]
         PMRs4.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[5]    0x0000000000000000 [0]
         PMRs5.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs5.reservatn(63:63)        0x0                False
      page_migration_reservation[6]. . 0x0000000000000000 [0]
         PMRs6.offset(48:56)           0x0                [0]
         PMRs6.reservatn(63:63). . . . 0x0 . . . . . . .  False
      page_migration_reservation[7]    0x0000000000000000 [0]
         PMRs7.offset(48:56) . . . . . 0x0 . . . . . . .  [0]
         PMRs7.reservatn(63:63)        0x0                False
```

Example 3-49   hfi_read_regs output example for “hfi0” link with performance counters

```
[...]
Performance Counters
:::=================

   Performance Counters: ISR
      cycles blocked sending . . . . . 0x0000000000000000 [0]
      flits sent                       0x0000000000000000 [0]
      flits dropped. . . . . . . . . . 0x0000000000000000 [0]
      link retries                     0x0000000000000000 [0]

   Performance Counters: HFI
      agg pkts sent. . . . . . . . . . 0x0000000008ff67c3 [150955971]
      agg pkts dropped sendng          0x0000000000000000 [0]
      agg pkts received. . . . . . . . 0x0000000008f168e5 [150038757]
      agg pkts dropped rcving          0x0000000000000036 [54]
      agg imm send pkt count . . . . . 0x0000000000000526 [1318]
      agg send recv pkt count          0x0000000000000e00 [3584]
      agg fullRDMA sent count. . . . . 0x00000000084d5339 [139285305]
      agg halfRDMA sent count          0x0000000000000000 [0]
      agg smallRDMA sent count . . . . 0x0000000000000000 [0]
      agg ip pkt sent count            0x0000000000a4f0a3 [10809507]
      agg cau pkt sent count . . . . . 0x0000000000000000 [0]
      agg gups pkt sent count          0x0000000000000000 [0]
      addr xlat wait count . . . . . . 0x0000000239f7adff [9562467839]
```

3.1.8 Reliable Scalable Cluster Technology

The Reliable Scalable Cluster Technology (RSCT) RMC subsystem is part of the RSCT software, and is required for implemented sensing capabilities in the supported operating systems. Example 3-50 on page 211 and Example 3-51 on page 211 demonstrate how to check the status and some specific subsystem details. For more information about the RSCT or RMC subsystem, see to Table 3-1 on page 158.

Example 3-50 Displaying only the status of RMC daemon

```
I# lssrc -s ctrmc
Subsystem         Group            PID          Status
 ctrmc            rsct             12648636     active
```

Example 3-51 Displaying detailed information about the RMC subsystem

```
# lssrc -l -s ctrmc
Subsystem         Group            PID          Status
 ctrmc            rsct             12648636     active
Trace flags set:
_SEM
      Errors = 0          Info = 0           API = 0          Perf = 0
      Cipher = 0
_SEC
      Errors = 0           API = 0          Info = 0          Perf = 0
_SEH
      Errors = 0          Info = 0           API = 0        SvcTkn = 0
      CtxTkn = 0           IDM = 0           ACL = 0          Cred = 0
        Auth = 0
_SEU
      Errors = 0          Info = 0           API = 0        Buffer = 0
      SvcTkn = 0        CtxTkn = 0
_SEL
      Errors = 0          Info = 0           API = 0        Buffer = 0
        Perf = 0
_SEI
       Error = 0           API = 0       Mapping = 0     Milestone = 0
        Diag = 0
_SRA
         API = 0        Errors = 0      Wherever = 0
_SKD
      Errors = 0          Info = 0         Debug = 0
_SEA
      Errors = 0          Info = 0           API = 0        Buffer = 0
      SVCTKN = 0        CTXTKN = 0
_PRM
        Info = 0
_MCD
        init = 1        config = 1         insts = 0        rmctrl = 1
         cci = 1           mcp = 1          obsv = 0          evgn = 1
         reg = 1           pci = 1          msgs = 0         query = 0
         gsi = 1          eval = 0           rdi = 0         sched = 0
         shm = 0           sec = 1         routg = 1       cmdproc = 0
        sami = 0        rstree = 0           rcv = 0           sri = 0
        dcfg = 1       iolists = 0      nodestat = 1        ipstat = 1
     cmdflow = 1

Configuration Data Base version from local copy of CDB:

Daemon started on Tuesday 10/04/11 at 12:21:10
Daemon has been running 17 days, 0 hours, 35 minutes and 56 seconds
Client message policy is Enabled
Client message timeout: 10
Client start session timeout: 300
Client first command threshold: 150
Client first command timeout: 10
Client first command timeout applies only to unauthenticated users
Daemon communications are enabled

Logical Connection Information for Local Clients
    LCID          FD           PID     Start Time
       0          19        20512912    Friday 10/14/11 09:30:37
       1          22        20512912    Friday 10/14/11 09:30:37

Logical Connection Information for Remote Clients
    LCID          FD           PID     Start Time

Resource Manager Information
         Name           ClassKey      ID         FD       SHMID    Mem Charge
IBM.HostRM                   1         0         -1          -1             0
IBM.StorageRM                1         1         -1          -1             0
IBM.GblResRM                 1         2         -1          -1             0
IBM.ERRM                     1         3         -1          -1             0
IBM.AuditRM                  1         4         -1          -1             0
IBM.RecoveryRM               1         5         -1          -1             0
IBM.ConfigRM                 1         6         24          -1             0
IBM.DRM                      1         7         23          -1             0
IBM.FSRM                     1         8         -1          -1             0
IBM.LPRM                     1         9         -1          -1             0
IBM.MgmtDomainRM             1        10         -1          -1             0
IBM.MicroSensorRM            1        11         -1          -1             0
IBM.SensorRM                 1        12         25          -1             0
IBM.ServiceRM                1        13         18          -1             0
IBM.TestRM                   1        14         -1          -1             0
IBM.WLMRM                    1        15         -1          -1             0

Highest file descriptor in use is 25

Internal Daemon Counters
 (0000)    GS init attempts =          0  GS join attempts =          0
           GS resp callback =      12249  writev() stream  =     110164
 (0010)    msgs wrtn stream =     110155  maxd wrtn stream =          3
           CCI conn rejects =          0  RMC conn rejects =          0
 (0020)    Retry req msg    =          0  Retry rsp msg    =          0
           Intervl usr util =          1  Total usr util   =        939
 (0030)    Intervl sys util =          0  Total sys util   =       1406
           Intervl time     =      12020  Total time       =  147089591
 (0040)    lccb's created   =       6124  lccb's freed     =       6122
           ctb's created    =      12327  ctb's freed      =      12325
 (0050)    smb's created    =          0  smb's freed      =          0
           SAMI recs creatd =         16  SAMI recs freed  =         14
 (0060)    mnpd's created   =          0  mnpd's freed     =          0
           ptb's created    =          0  ptb's freed      =          0
 (0070)    Events regstrd   =         16  Events unregstrd =         14
           event cb created =         16  event cb freed   =         14
 (0080)    LERL created     =          0  LERL freed       =          0
           Events generated =          0  Redirects        =          0
 (0090)    PRM msgs to all  =          0  PRM msgs to peer =          0
           PRM resp msgs    =          0  PRM msgs rcvd    =          0
 (00a0)    PRM_NODATA       =         0 PRM_HBRECV       =          0
           PRM_BADMSG errs  =         0 PRM authent errs =          0
 (00b0)    PRM config errs  =         0 data sink msgs   =          0
           Sched q elements =        16 Free q elements  =         14
 (00c0)    xcb allocated    =    110231 xcb free         =     110229
           xcb free cmdlist =     12325 xcb free clntcmd =          0
 (00d0)    xcb free clntrsp =     36775 xcb free frmpeer =          0
           xcb free topeer  =         0 xcb free peerrsp =          0
 (00e0)    xcb free rmrsp   =         0 xcb free rmcmd   =      61129
           xcb free unknown =         0 reserved         =          0
 (00f0)    Generic q elems  =        16 Free gq elems    =         16
           Rsearch nodes    =         0 Free RS nodes    =          0
 (0100)    Rsrc Hndl CBs    =         0 Free RHCBs       =          0
           Peer Msg Anchors =         0 Free PMAs        =          0
 (0110)    Event Rsrc CBs   =         0 free ERCBs       =          0
           NG RH references =         0 free NRRs        =          0
 (0120)    Node Name CBs    =         0 free NNCBs       =          0
           Common SM bufs   =         0 free CSMBs       =          0
 (0130)    rhcl's created   =         0 rhcl's freed     =          0
           CFA's created    =         0 CFA's freed      =          0
 (0140)    CFR's created    =         0 CFR's freed      =          0
           ACL's created    =         3 ACL's freed      =          0
 (0150)    Sec rec methods  =        20 Sec authent      =       6124
           Missed sec rsps  =         0 Wake sec thread  =       6145
 (0160)    Wake main thread =      6143 Enq sec request  =       6145
           Deq sec request  =      6145 Enq sec response =       6145
 (0170)    Deq sec response =      6145 Timer starts     =       6124
           1st msg timeouts =         0 Message timeouts =          0
 (0180)    Start timeouts   =         0 Command timeouts =          0

Daemon Resource Utilization Last Interval
User:                 0.010 seconds    0.008%
System:               0.000 seconds    0.000%
User+System:          0.010 seconds    0.008%

Daemon Resource Utilization Total
User:                 9.390 seconds    0.001%
System:              14.060 seconds    0.001%
User+System:         23.450 seconds    0.002%

Data segment size:  2211K
```

3.1.9 Compilers environment

For the compilation and runtime environments (PE Runtime Edition, ESSL, and Parallel ESSL) monitoring tools are not needed because these environments often are a resource for application code development and runtime optimizations. For more information about PE Runtime Edition, ESSL, and Parallel ESSL, 2.6, “Running workloads by using IBM LoadLeveler” on page 144.

3.1.10 Diskless resources

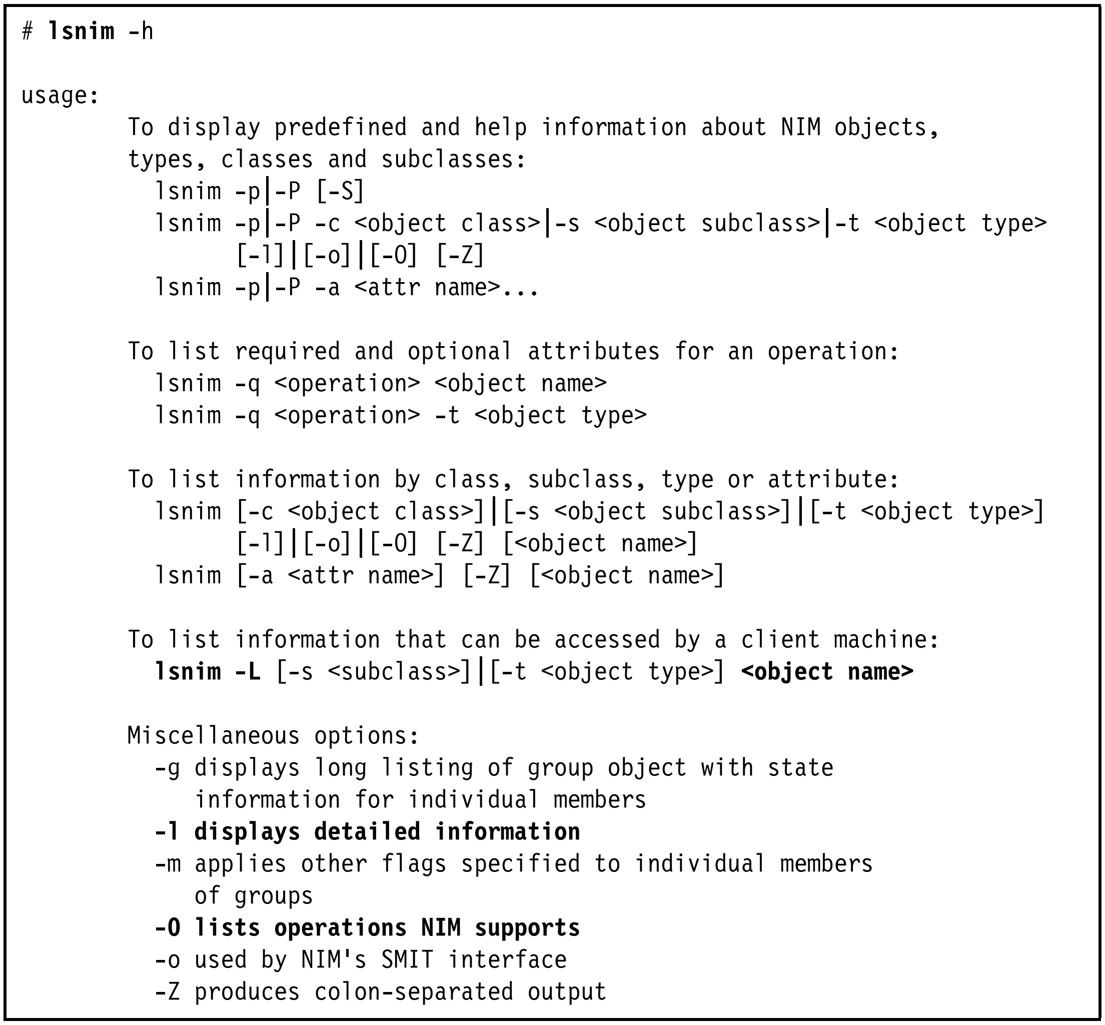
In this section, we describe how to monitor resources that are needed for node installation and reconfiguration. For more information, see Table 3-1 on page 158.

NIM

For remote installation in AIX, we use the Network Installation Manager (NIM). The following command is commonly used to monitor NIM resources:

lsnim

For this command, we list the help description that is shown in Figure 3-24. Typical output is shown in Example 3-52 on page 215 and Example 3-53 on page 215. For more information about this command, see to Table 3-1 on page 158 in the “Diskless resources (AIX) - (NIM)” section.



```
# lsnim -h

usage:
        To display predefined and help information about NIM objects,
        types, classes and subclasses:
          lsnim -p|-P [-S]
          lsnim -p|-P -c <object class>|-s <object subclass>|-t <object type>
                [-l]|[-o]|[-O] [-Z]
          lsnim -p|-P -a <attr name>...

        To list required and optional attributes for an operation:
          lsnim -q <operation> <object name>
          lsnim -q <operation> -t <object type>

        To list information by class, subclass, type or attribute:
          lsnim [-c <object class>]|[-s <object subclass>]|[-t <object type>]
                [-l]|[-o]|[-O] [-Z] [<object name>]
          lsnim [-a <attr name>] [-Z] [<object name>]

        To list information that can be accessed by a client machine:
          lsnim -L [-s <subclass>]|[-t <object type>] <object name>

        Miscellaneous options:
          -g displays long listing of group object with state
             information for individual members
          -l displays detailed information
          -m applies other flags specified to individual members
             of groups
          -O lists operations NIM supports
          -o used by NIM's SMIT interface
          -Z produces colon-separated output
```

Figure 3-24 lsnim command flag description

Example 3-52 lsnim output example with no flags

```
# lsnim
master                             machines        master
boot                               resources       boot
nim_script                         resources       nim_script
itso                               networks        ent
GOLD_71BSN_resolv_conf             resources       resolv_conf
GOLD_71BSN_bosinst_data            resources       bosinst_data
GOLD_71BSN_lpp_source              resources       lpp_source
GOLD_71BSN                         resources       spot
xCATaixSN71                        resources       installp_bundle
c250f10c12ap01                     machines        standalone
xcataixscript                      resources       script
GOLD_71Bdskls_dump                 resources       dump
GOLD_71Bdskls_resolv_conf          resources       resolv_conf
GOLD_71Bdskls_lpp_source           resources       lpp_source
GOLD_71Bdskls                      resources       spot
GOLD_71Bdskls_shared_root          resources       shared_root
GOLD_71Bdskls_paging               resources       paging
xCATaixCN71                        resources       installp_bundle
IBMhpc_base                        resources       installp_bundle
IBMhpc_all                         resources       installp_bundle
GOLD_71Bdskls_util                 resources       spot
GOLD_71Bdskls_util_shared_root     resources       shared_root
```

Example 3-53 lsnim output example for resolv_conf, bosinst_data, lpp_source, and spot

```
# lsnim -l GOLD_71BSN_resolv_conf
GOLD_71BSN_resolv_conf:
   class       = resources
   type        = resolv_conf
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /install/nim/resolv_conf/GOLD_71BSN_resolv_conf/resolv.conf
   alloc_count = 0
   server      = master
# lsnim -l GOLD_71BSN_bosinst_data
GOLD_71BSN_bosinst_data:
   class       = resources
   type        = bosinst_data
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /install/nim/bosinst_data/GOLD_71BSN_bosinst_data
   alloc_count = 0
   server      = master
# lsnim -l GOLD_71BSN_lpp_source
GOLD_71BSN_lpp_source:
   class       = resources
   type        = lpp_source
   arch        = power
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /install/nim/lpp_source/GOLD_71BSN_lpp_source
   simages     = yes
   alloc_count = 0
   server      = master
# lsnim -l GOLD_71BSN
GOLD_71BSN:
   class         = resources
   type          = spot
   plat_defined  = chrp
   arch          = power
   bos_license   = yes
   Rstate        = ready for use
   prev_state    = verification is being performed
   location      = /install/nim/spot/GOLD_71BSN/usr
   version       = 7
   release       = 1
   mod           = 0
   oslevel_r     = 7100-00
   alloc_count   = 0
   server        = master
   if_supported  = chrp.64 ent
   Rstate_result = success
```

iSCSI debug and dumps

On an IBM Power Systems 775 cluster, iSCSI devices (physical- or logical-based) are used only to debug a system problem or to initiate a dump procedure.

On AIX, the iSCSI support is built into the operating system and must be configured to take advantage of the operating system. For more information about how to configure this support, see Table 3-1 on page 158.

For the Linux based cluster, you must configure the devices and ensure that the iSCSI daemon is running, as shown in Example 3-54.

Example 3-54 List the status of the iscsi daemon (Linux only)

```
# service iscsi status
iscsi is stopped
```

NFS

For the NFS subsystem, we check the status of their daemons and list the actual exported directories as shown in Example 3-55, Example 3-56 on page 217, and Example 3-57 on page 217.

Example 3-55 Lists nfs group processes status (AIX only)

```
# lssrc -g nfs
Subsystem         Group            PID          Status
 biod             nfs              7471108      active
 nfsd             nfs              11010194     active
 rpc.mountd       nfs              7274508      active
 rpc.lockd        nfs              3276910      active
 rpc.statd        nfs              8192224      active
 nfsrgyd          nfs                           inoperative
 gssd             nfs                           inoperative
```

Example 3-56 Lists the exported directories on the system that are running the command

```
# showmount -e
export list for c250mgrs40-itso:
/install/postscripts (everyone)
/mntdb2              (everyone)
```

Example 3-57 Lists nfs processes status (Linux only)

```
# service nfs status
rpc.svcgssd is stopped
rpc.mountd (pid 2700) is running...
nfsd (pid 2697 2696 2695 2694 2693 2692 2691 2690) is running...
rpc.rquotad (pid 2684) is running...
```

TFTP

For the TFTP subsystem, we check the status by issuing the **lssrc** command, as shown in Example 3-58.

Example 3-58 Display the status of the tftpd daemon (AIX only)

```
# lssrc -s tftpd
Subsystem         Group            PID          Status
 tftpd            tcpip            4849974      active
```

3.2 TEAL tool

Introduced with the IBM Power System 775 clusters, the TEAL tool provides automatic alerts for specific events that are taking place on the cluster. TEAL provides a central point of monitoring on the EMS. For more information about TEAL, see 1.9.4, “Toolkit for Event Analysis and Logging” on page 75 and Table 3-4.

Table 3-4 TEAL documentation

Section	Type	Link/Description
Overview	HTML (SourceForge): http://pyteal.sourceforge.net	http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Overview
Latest News		http://sourceforge.net/apps/wordpress/pyteal/
Command Reference		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Command_Reference
Install		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Install
Configuration		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Configuration
Release Notes		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Release_Notes
Development		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Development
Environment Variables		http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Environment_Variables
Setting up TEAL	HTML (SourceForge)	http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setting_up_TEAL_on_xCAT_Management_Node
Management Guide	PDF	High-performance clustering that uses the 9125-F2C
Service Guide		

3.2.1 Configuration for LoadLeveler, GPFS, Service Focal Point, PNSD, ISNM

In addition to monitoring the local system for both historic and real-time analysis, TEAL records (from connectors) and analyses (from filters) events from other components, as shown in Table 3-5 on page 219. Actions are triggered through plug-ins as a result of the recording and analysis process alerts that are produced.

Plug-ins

Incorporated into the TEAL code, plug-ins represent added features. One type of plug-ins is the filters that select the appropriate contents to generate alerts. As a result, alerts are generated and then forwarded to a listener. The listeners also act as plug-ins to the base framework of TEAL. The following plug-ins are supported:

- RMC sensor
- File-based (file, stderr, or stdout)
- SMTP to send email
- Call another program

Connectors

Table 3-5 AIX connectors for TEAL

Component connector	Fileset package	Description
LoadLeveler	teal.ll	LoadLeveler events (including daemon down, job vacate, and job rejection)
GPFS	teal.gpfs	GPFS events
Service Focal Point (HMC)	teal.sfp	Hardware and some software events that are sent to HMC
Protocol Network Services daemon	teal.pnsd	
ISNM (CNM and hardware server)	teal.isnm	CNM and Hardware Server events

3.2.2 Management

The TEAL commands that are shown in Table 3-6 are available to interactively manage the tool, and the available log files that are described in Table 3-7, which show the commands that are best-suited for day-to-day monitoring actions.

Table 3-6 Command reference description

Command	Command description
teal	Runs TEAL tool (python script) for real time and historical analysis.
tlchalert	Closes an active alert.
tlgpfschnode	Change GPFS monitoring node or start/stop monitoring of GPFS (AIX only).
tlgpfspurge	Removes entities under a cluster (AIX only).
tlgpfsstatus	Shows health status of GPFS (AIX only).
tllsalert	Lists alerts that are reported for the cluster.
tllsckpt	Lists checkpoints.
tllsevent	Lists events that occurred on the cluster.
tlrmalert	Removes alerts from the alert log.
tlrmevent	Removes events from the event log.
tltab	Database table maintenance.

Table 3-7 Log location reference

Log file at /var/log/teal	Log file description
teal_conn.log	ISNM, GPFS connectors
teal_sfp.log	SFP connector
teal_ll.log	LoadLeveler connector
teal_pnsd.log	PNSD connector
teal.log	TEAL command (historical and real-time analysis)

Log file at /var/log/teal	Log file description
tlchalert.log	Command log files
tllsalert.log	
tllschkpt.log	
tllsevent.log	
tlrmalert.log	
tlrmevent.log	

The following commands from Table 3-6 on page 219 are used to check all the information that is logged by TEAL:

- **tlgpfsstatus**
- **tllsalert**
- **tllsckpt**
- **tllsevent**

tlgpfsstatus

For this command, we list the help description that is shown in Figure 3-25.

```
# tlgpfsstatus -h
Usage: ./tlgpfsstatus[{options}]
  [options] can be any of
    -g                        - print global status of all entities
    -c                        - print status of all clusters
    -n                        - print status of all nodes
    -f                        - print status of all file systems
    -d                        - print status of all disks
    -s                        - print status of all storage pools
    -t                        - print status of all filesets
    -r                        - print status of all recovery groups
    -a                        - print status of all declustered arrays
    -p                        - print status of all pdisks
    -v                        - print status of all vdisks
    -e "colName=colValue"  - expression to filter query result
    -l                        - print detailed status of a entity
    -h                        - print this message
```

Figure 3-25 tlgpfsstatus command flag description

The **tlgpfsstatus** command is used to query status from the database. The command accesses the database and queries a series of tables to ascertain the status of different levels according to the options that are given by user. The command must be executed on the EMS side.

tllsalert

For this command, we list the help description that is shown in Figure 3-26 on page 221. A typical output is shown in Example 3-59 on page 222.

```
# tllsalert -h
Usage: tllsalert [options]

Options:
  -h, --help            show this help message and exit
  -q QUERY, --query=QUERY
                        Query parameters used to limit the range of alerts
                        listed. See list of valid values below
  -f OUTPUT_FORMAT, --format=OUTPUT_FORMAT
                        Output format of alert: json,csv,text [default =
                        brief]
  -w, --with-assoc      Print the associated events and alerts for the
                        matching alert
  -a, --all             Print all open and closed alerts
  -c, --closed          Print only closed alerts
  -d, --with-dups       Print the duplicate alerts also

Valid query values and their operations and formats:

rec_id        - =           - A single id or a comma-separated list of ids
(equals-only)

alert_id      - =           - A single id or a comma-separated list of ids
(equals-only)

creation_time - =,<.>,>=,<= - A time value in the format YYYY-MM-DD-HH:MM:SS

severity      - =           - The severity level, listed in order of severity:
                                F=fatal, E=error, W=warning, I=info
(equals-only)

urgency       - =           - The urgency of the alert, listed in order of
urgency:
                                I=immediate, S=schedule, N=normal, D=defer,
O=optional
                                (equals-only)

event_loc     - =           - A location in the format <location
type>:<location>.
                               The location is optional; otherwise all events
                               with the same location type will be included

event_scope   - =           - A scoping value for the specified reporting
location type

src_name      - =           - A single value or a comma-separated list of
values
```

Figure 3-26 tllsalert command flag description

For more information about the **tllsalert** command, see this website:

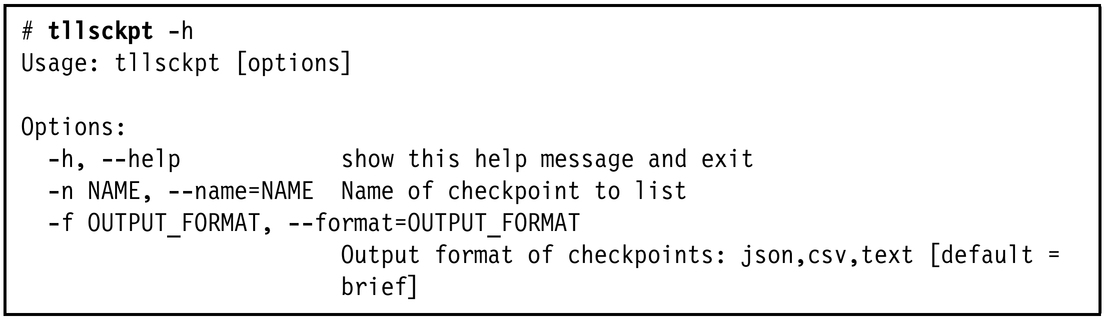
http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Command_-_tllsalert

Example 3-59 tllsalert command output with some events recorded (still in open state)

```
# tllsalert -a
    1: TL000001 2011-10-04 14:08:22.410845 A:c250mgrs40-itso##teal.py##11141200
    2: TL000001 2011-10-04 14:09:15.184180 A:c250mgrs40-itso##teal.py##11862164
    3: TL000001 2011-10-04 14:09:40.337409 A:c250mgrs40-itso##teal.py##11862180
    4: TL000001 2011-10-04 14:10:01.508925 A:c250mgrs40-itso##teal.py##22478866
    5: TL000001 2011-10-04 14:11:10.391145 A:c250mgrs40-itso##teal.py##8913014
    6: BDFF0070 2011-10-18 22:27:07.049295 H:FR010-CG14-SN008-DR3-HB1-HF1-RM3
    7: BD400020 2011-10-22 00:48:54.915296 H:FR010-CG14-SN008-DR3-HB1-HF0-RM2
    8: BDFF0070 2011-10-24 16:16:17.478265 H:FR010-CG14-SN008-DR3-HB3-HF1-RM0
    9: BD400020 2011-10-26 10:52:26.340951 H:FR010-CG14-SN008-DR3-HB2-HF0-RM0
   10: BDFF0070 2011-10-27 09:28:02.211962 H:FR010-CG14-SN008-DR3-HB0-HF1-RM3
```

tllsckpt

For this command, we list the help description shown in Figure 3-27. A typical output is shown in Example 3-60.



```
# tllsckpt -h
Usage: tllsckpt [options]

Options:
  -h, --help            show this help message and exit
  -n NAME, --name=NAME  Name of checkpoint to list
  -f OUTPUT_FORMAT, --format=OUTPUT_FORMAT
                        Output format of checkpoints: json,csv,text [default =
                        brief]
```

Figure 3-27 tllsckpt command flag description

The **tllsckpt** prints the analyzer event records ids checkpoints and shows the last event that is processed by a specific analyzer and the current high watermark. For more information about the **tllsckpt** command, see this website:

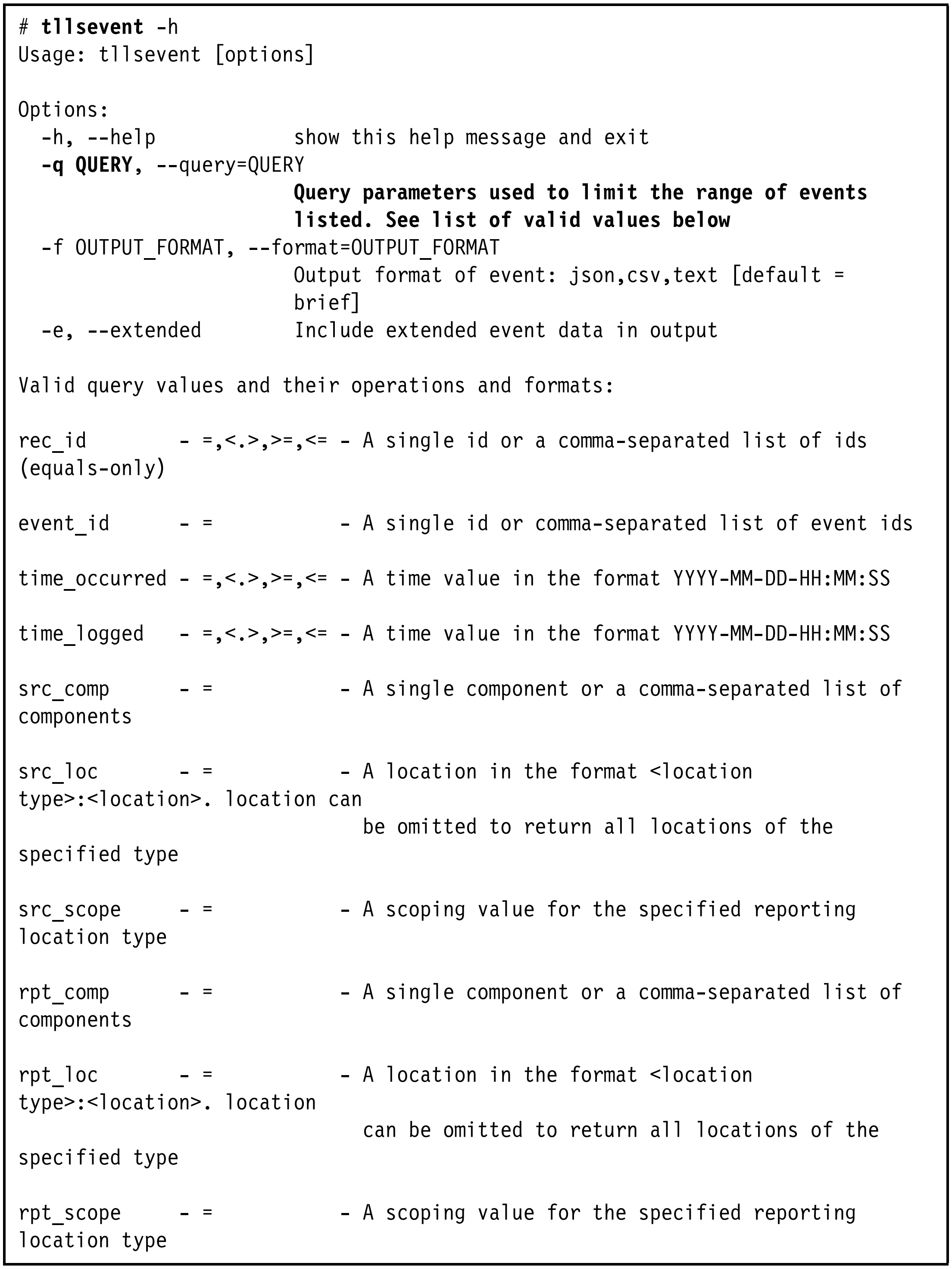
http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Command_-_tllsckpt

Example 3-60 tllsckpt command output

```
# tllsckpt
monitor_event_queue  R   678
SFPEventAnalyzer     R  None
LLEventAnalyzer      R  None
PNSDEventAnalyzer    R  None
CnmEventAnalyzer     R   678
MAX_event_rec_id         678
```

tllsevent

For this command, we list the help description that is shown in Figure 3-28 on page 223. A typical output is shown in Example 3-61 on page 224.



```
# tllsevent -h
Usage: tllsevent [options]

Options:
  -h, --help            show this help message and exit
  -q QUERY, --query=QUERY
                        Query parameters used to limit the range of events
                        listed. See list of valid values below
  -f OUTPUT_FORMAT, --format=OUTPUT_FORMAT
                        Output format of event: json,csv,text [default =
                        brief]
  -e, --extended        Include extended event data in output

Valid query values and their operations and formats:

rec_id        - =,<.>,>=,<= - A single id or a comma-separated list of ids
(equals-only)

event_id      - =           - A single id or comma-separated list of event ids

time_occurred - =,<.>,>=,<= - A time value in the format YYYY-MM-DD-HH:MM:SS

time_logged   - =,<.>,>=,<= - A time value in the format YYYY-MM-DD-HH:MM:SS

src_comp      - =           - A single component or a comma-separated list of
components

src_loc       - =           - A location in the format <location
type>:<location>. location can
                              be omitted to return all locations of the
specified type

src_scope     - =           - A scoping value for the specified reporting
location type

rpt_comp      - =           - A single component or a comma-separated list of
components

rpt_loc       - =           - A location in the format <location
type>:<location>. location
                              can be omitted to return all locations of the
specified type

rpt_scope     - =           - A scoping value for the specified reporting
location type
```

Figure 3-28 tllsevent command flag description

For more information about the **tllsevent** command, see this website:

http://sourceforge.net/apps/mediawiki/pyteal/index.php?title=Command_-_tllsevent

Example 3-61 tllsevent command output with a query for all events after "2011-10-27 at 9:10:00"

```
/tllsevent -q time_occurred\>"2011-10-27-09:10:00"
  671: BD400020 2011-10-27 09:23:01.757546 H:FR010-CG14-SN008-DR3-HB0-HF1-RM0
  672: BD400020 2011-10-27 09:23:09.759567 H:FR010-CG14-SN008-DR3-HB0-HF1-RM1
  673: BD400020 2011-10-27 09:23:09.907939 H:FR010-CG14-SN008-DR3-HB0-HF1-RM3
  674: BD400020 2011-10-27 09:23:09.956038 H:FR010-CG14-SN008-DR3-HB0-HF1-RM0
  675: BD400020 2011-10-27 09:23:10.039571 H:FR010-CG14-SN008-DR3-HB0-HF1-RM0
  676: BD400020 2011-10-27 09:23:10.091358 H:FR010-CG14-SN008-DR3-HB0-HF1-RM1
  677: BD400020 2011-10-27 09:23:10.138539 H:FR010-CG14-SN008-DR3-HB0-HF1-RM3
  678: BD400020 2011-10-27 09:23:10.258573 H:FR010-CG14-SN008-DR3-HB0-HF1-RM0
```

3.3 Quick health check for full HPC cluster system

In this section, an organized set of procedures that checks the status of all of the components of a 775 HPC Cluster is described. The procedures help determine whether a problem is detected, and which steps might determine the problem.

> **For more information:** For more information about high-performance clustering by using the 9125-F2C, see this website:
>
> `https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_planning_installation_guide.rev1.2.pdf?version=1`

3.3.1 Component analysis location

Table 3-8 shows the local 775 cluster component distribution over the different types of cluster nodes.

Table 3-8 Component distribution over node type

Node type	Component	Description
EMS	xCAT	Check all xCAT information, such as: xCAT deamon (xcatd), data in the xCAT table, the xCAT database configuration, the status of ODBC setup, the status of hardware connection, and the running status of nodes.
	DB2	Check all DB2 information, such as: the running status of DB2 server, the xcatdb status to connect to database from DB2 server, the health snapshot for database on xcatdb, the path pointed of DB2 instance directory (default /var/lib/db2), the size of DB2 database, and the DB2 license.
	TEAL	Check all TEAL information, such as: the status of TEAL daemon, the log information for TEAL, the events and alerts of ISNM, SFP, LoadLeveler, PNSD, and GPFS
	ISNM	Check all ISNM information, such as: the running status of CNM, the connection status of LNMC on FSP, the status of miswired links, the network topology, and the ISR links information.
	NIM	Check all NIM objects information, such as: the status of diskfull and diskless image, the status of NIM network, the status of NIM machine for diskfull node, the status of NIM dump, and the information of NIM bundle.

Node type	Component	Description
Service node	GPFS (if installed)	Check GPFS information, such as: the status of the GPFS daemon on the nodes, GPFS cluster configuration information, the list that the nodes have a given GPFS file system that is mounted, NSD information for the GPFS cluster, and policy information.
	HFI	Check HFI information, such as: the status of HFI adapters, the status of HFI interfaces, HFI IP address, HFI MAC address, HFI statistics, data from HFI logical device, HFI registers values, and dump info of HFI device.
	LoadLeveler	Check LoadLeveler information, such as: the running status of LoadLeveler, machine status, job status, and job resource information for accounting and class information.
	NIM (diskless nodes)	Check NIM objects information, especially diskless nodes, such as: the status of diskless image, share root and paging, the status of NIM HFI network, the status of NIM machine for diskless node, the status of NIM dump, and the information of NIM bundle.
I/O GPFS node	GPFS and GPFS GNR	Check GPFS information, such as: the status of the GPFS daemon on the nodes, GPFS cluster configuration information, the list of nodes that feature a given GPFS file system mounted, NSD information for the GPFS cluster, policy information, VDisk topology that is found on all attached DE, and SAS Enclosure Services (SES) Device and SAS Adapter information of DE.
	HFI	Check HFI information, such as: the status of HFI adapters, the status of HFI interfaces, HFI IP address, HFI MAC address, HFI statistics, data from HFI logical device, HFI registers values, and dump information of HFI device.
Utility node	GPFS	Check GPFS information, such as: the status of the GPFS daemon on the nodes, GPFS cluster configuration information, the list of nodes that feature a given GPFS file system that is mounted, NSD information for the GPFS cluster, and policy information.
	HFI	Check HFI information, such as: the status of HFI adapters, the status of HFI interfaces, HFI IP address, HFI MAC address, HFI statistics, data from HFI logical device, HFI registers values, and dump information of HFI device.
	LoadLeveler	Check LoadLeveler information, such as: the running status of LoadLeveler, machine status, job status, and job resource information for accounting and class information.
Compute node	GPFS	Check GPFS information, such as: the status of the GPFS daemon on the nodes, GPFS cluster configuration information, the list of nodes that feature a given GPFS file system that is mounted, NSD information for the GPFS cluster, and policy information.
	HFI	Check HFI information, such as: the status of HFI adapters, the status of HFI interfaces, HFI IP address, HFI MAC address, HFI statistics, data from HFI logical device, HFI registers values and dump information of HFI device.
	LoadLeveler	Check LoadLeveler information, such as: the running status of LoadLeveler, machine status, job status, and job resource information for accounting and class information.

3.3.2 Top-to-bottom checks direction for software to hardware

Complete the following steps to check your environment by using a top-to-bottom approach:

1. From the Service node:
   a. [LoadLeveler] **`llstatus -S`**
   b. (Optional) [LoadLeveler] **`llq`**
   c. (Optional) [LoadLeveler] **`llstatus -L machine`**
   d. [AIX] **`df -g`**
   e. [AIX] **`lsvg rootvg`**
2. From the EMS:
   a. [xCAT] **`nodestat <lpar+fsp> -p`**
   b. [xCAT] **`rpower <lpar+fsp> stat`**: Check power state differences with last command.
   c. [xCAT] **`lshwconn fsp`**
   d. [ISNM] **`lsnwloc`**
   e. [TEAL] **`lssrc -s teal`**
   f. [TEAL] **`tllsckpt -f text`**
   g. [RSCT] **`lssrc -a | grep rsct`**
   h. [NIM] **`lssrc -g nim`**
   i. [SSH] **`lssrc -g ssh`**
   j. [NFS] **`lssrc -g nfs`**
   k. [TCPIP] **`lssrc -g tcpip`**: Check used daemons such as “dhcpsd”, “xntpd”, “inetd”, “tftpd”, “named”, and “snmpd”.
   l. [AIX] df -g
   m. [AIX] lsvg rootvg
3. From the I/O GPFS node:
   a. [GPFS] **`mmgetstate -L -s -a`**
   b. [GPFS] **`mmlsfs`**
   c. [GPFS] **`mmlsnsd`**
   d. [GPFS] **`mmlsvdisk`**

3.3.3 Bottom-to-top direction for hardware to software

Complete the following steps to check your environment by using a bottom-to-top approach:

1. From the EMS (hardware):
   a. [ISNM] **`lsnwloc`**
   b. [xCAT] **`lshwconn fsp`**
   c. [xCAT] **`nodestat <lpar+fsp> -p`**
   d. [xCAT] **`rpower <lpar+fsp> stat`** (Check power state differences with last command.)
2. From the I/O GPFS node:
   a. [GPFS] **`mmgetstate -L -s -a`**
   b. [GPFS] **`mmlsfs`**
   c. [GPFS] **`mmlsnsd`**
   d. [GPFS] **`mmlsvdisk`**

3. From the EMS (software):
   a. [AIX] **lsvg rootvg**
   b. [AIX] **df -g**
   c. [TEAL] **lssrc -s teal**
   d. [TEAL] **tllsckpt -f text**
   e. [RSCT] **lssrc -a | grep rsct**
   f. [NIM] **lssrc -g nim**
   g. [SSH] **lssrc -g ssh**
   h. [NFS] **lssrc -g nfs**
   i. [TCPIP] **lssrc -g tcpip** (Check used daemons such as “dhcpsd”, “xntpd”, “inetd”, “tftpd”, “named”, and “snmpd”.)
4. From the Service Node:
   a. [AIX] l**svg rootvg**
   b. [AIX] **df -g**
   c. [LoadLeveler] **llstatus -S**
   d. (Optional) [LoadLeveler] **llq**
   e. (Optional) [LoadLeveler] **llstatus -L machine**

3.4 EMS Availability+

This section describes the Availability+ functionality that is implemented with the 775 cluster systems when two EMS physical machines are used.

For more information: For more information about high-performance clustering that uses the 9125-F2C, see the Management and Service Guide at this website:

https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_planning_installation_guide.rev1.2.pdf?version=1

Failover procedure: For more information about failover procedure, see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Setup_HA_Mgmt_Node_With_Shared_Disks#Failover

3.4.1 Simplified failover procedure

This section describes the simplified failover procedure.

Disabling Primary EMS

Complete the following steps to disable the primary EMS:

1. Remove the following connections from CEC and FRAME:

   a. **`rmhwconn cec`**
   b. **`rmhwconn frame`**

2. Stop the following xCAT daemon and TEAL services (EMS and service nodes):

   a. Linux: **`service xcatd stop`**
   b. Linux: **`service teal stop`**
   c. Linux: **`service teal_ll stop`**
   d. AIX: **`stopsrc -s xcatd`**
   e. AIX: **`stopsrc -s teal`**
   f. AIX: **`stopsrc -s teal_ll`**

3. Unexport the following xCAT NFS directories:

   `exportfs -ua`

4. Stop DB2 database:

   a. **`su - xcatdb`**
   b. **`db2 connect reset`**
   c. **`db2 force applications all`**
   d. **`db2 terminate`**
   e. **`db2stop`**
   f. **`exit`**

5. Unmount all shared disk file systems:

   a. **`umount <directory_name>`**
   b. (Optional) **`fuser -uxc <directory_name>`**

6. Varyoff xCAT shared volume group (AIX only):

   `varyoffvg xcatvg`

7. Unconfigure service IP (also includes persistent configurations):

   a. AIX: **`chdev -l en0 -a delalias4=<IP>,<NetMask>`**
   b. Linux: **`ifconfig eth0:0 0.0.0.0 0.0.0.0`**

Enabling Primary EMS

Complete the following steps to enable the primary EMS:

1. Stop xCAT daemon and TEAL services:

   a. Linux: **`service xcatd stop`**
   b. Linux: **`service teal stop`**
   c. Linux: **`service teal_ll stop`**
   d. AIX: **`stopsrc -s xcatd`**
   e. AIX: **`stopsrc -s teal`**
   f. AIX: **`stopsrc -s teal_ll`**

2. Stop DB2 database:
   a. **`su - xcatdb`**
   b. **`db2 connect reset`**
   c. **`db2 force applications all`**
   d. **`db2 terminate`**
   e. **`db2stop`**
   f. **`exit`**
3. Configure service IP (includes persistent configurations):
   a. AIX: **`chdev -l en0 -a alias4=<IP>,<NetMask>`**
   b. Linux: **`ifconfig eth0:0 <IP> netmask <NetMask>`**
4. Varyon volume group (AIX only):

   `varyonvg xcatvg`
5. Mount file systems:
   a. On AIX:
      i. **`mount /etc/xcat`**
      ii. **`mount /install`**
      iii. **`mount /.xcat`**
      iv. **`mount /databaseloc`**
   b. On Linux:
      i. **`mount /dev/sdc1 /etc/xcat`**
      ii. **`mount /dev/sdc2 /install`**
      iii. **`mount /dev/sdc3 ~/.xcat`**
      iv. **`mount /dev/sdc4 /databaseloc`**
6. Update DB2 configuration:
   a. AIX: **`/opt/IBM/db2/V9.7/adm/db2set -g DB2SYSTEM=<new_node_name>`**
   b. Linux: **`/opt/ibm/db2/V9.7/adm/db2set -g DB2SYSTEM=<new_node_name>`**
   c. Non DB2 WSE versions: `/databaseloc/db2/sqllib/db2nodes.cfg` to use the new node name
7. Start DB2:
   a. **`su - xcatdb`**
   b. **`db2start`**
   c. **`exit`**
8. Start xCAT daemon:
   a. Linux: **`service xcatd start`**
   b. AIX: **`startsrc -s xcatd`**
9. Set up connection for CEC and Frame (DFM only):
   a. **`mkhwconn cec -t`**
   b. **`mkhwconn frame -t`**

10. Set up network services and conserver.

   a. Network services:

      i. DNS: Run **makedns**. Verify that DNS services are working for node resolution.
      ii. (Optional) DNS: Add "nameserver=<service_ip>" to /etc/resolv.conf

         For more information about setting up name resolution in an xCAT Cluster, see the following website:

         https://sourceforge.net/apps/mediawiki/xcat/index.php?title=Cluster_Name_Resolution

      iii. DHCP (Linux only): run **makedhcp**. Verify DHCP operational for hardware management.

      iv. Verify that bootp is operational for booting the nodes.

   b. Conserver:

```
makeconservercf
```

11. Start the HPC software daemons:

   a. **/opt/isnm/hdwr_svr/bin/hdwr_svr** (hardware server)
   b. Linux: **service teal start** (TEAL)
   c. Linux: **service teal_ll start** (TEAL)
   d. AIX: **startsrc -s teal** (TEAL)
   e. AIX: **startsrc -s teal_ll** (TEAL)
   f. /opt/isnm/cnm/bin/startCNMD (CNM)
   g. GPFS

Setting up the operating system deployment environment

This optional step is required only when you want to use the new primary management node to perform operating system (OS) deployment tasks.

Complete the following steps to use the primary management node to perform OS deployment tasks:

1. Create operating system images:

   For Linux: The operating system images definitions are already in the xCAT database, and the operating system image files are already in /install directory.

   For AIX: If the HANIM is used for keeping the NIM resources synchronized, manual steps are not needed to create the NIM resources on the standby management node. Otherwise, the operating system image files are in the /install directory, but you must create the NIM resources manually.

   Complete the following steps to re-create the NIM resources manually:

   a. If the NIM master is not initialized, run command "**nim_master_setup -a mk_resource=no -a device=<source 'directory>**" to initialize the NIM master, in which the <source 'directory> is the directory that contains the AIX installation image files.

   b. Run the following command to list all the AIX operating system images:

```
lsdef -t osimage -l
```

   c. For each osimage:

      i. Create the lpp_source resource:

         - **/usr/sbin/nim -Fo define -t lpp_source -a server=master -a location=/install**
         - **/nim/lpp_source/<osimagename>_lpp_source <osimagename>_lpp_source**

ii. Create the spot resource:

- **/usr/lpp/bos.sysmgt/nim/methods/m_mkspot -o -a server=master -a**
- location=/install/nim/spot/ -a source=no <osimage>

iii. Check if the osimage includes any of the following resources:

- "installp_bundle", "script", "root", "tmp", "home"
- "shared_home", "dump" and "paging"

If the resources exist, use the following commands:

- **/usr/sbin/nim -Fo define -t <type> -a server=master -a location=<location>**
- <osimagename>_<type>"

d. To create all the necessary NIM resources, use the resource location returned by "**lsdef -t osimage -l**" command.

If the osimage includes shared_root resource is defined, the shared_root resource directory must be removed before the shared_root resource is created, as shown in the following example:

```
rm -rf /install/nim/shared_root/71Bshared_root/
```

```
/usr/sbin/nim -Fo define -t shared_root -a server=master -a
location=/install/nim/shared_root/71Bshared_root -a spot=71Bcosi
71Bshared
```

> **NIM master:** If the NIM master was running on the standby management node before failover, the NIM master hostname must be changed. Make this change by using `smit nim` to perform the NIM master hostname change.

If you are experiencing Secure Shell (SSH) issues when you are attempting to secure the compute nodes or any other nodes, the hostname in the SSH keys under the `$HOME/.ssh` directory must be updated.

e. Run **nimnodeset** or **mkdsklsnode**

Before run nimnodeset or mkdsklsnode, make sure the entries in file `/etc/exports` match the exported NIM resources directories, otherwise, you receive exportfs errors and nimnodeset/mkdsklsnode is not completed successfully.

Performing management operations

After these steps are completed, the standby management node is ready to manage the cluster, and you run any xCAT commands to manage the cluster. For example, if the diskless nodes need to be rebooted, you can run the following command to initialize the network reboot:

rpower <noderange> reset or **rneboot <noderange>**

Complete the following steps to restart the NFS service and re-export the NFS exports:

1. On AIX, run the following commands:

   a. **exportfs -ua**
   b. **stopsrc -g nfs**
   c. **startsrc -g nfs**
   d. **exportfs -a**

2. On Linux, run the following commands:

   a. **exportfs -ua**
   b. **service nfs stop**
   c. **service nfs start**
   d. **exportfs -a**

3. (Optional) Run the following commands to unconfigure the serviceIIP on the previous primary management node when the node is recovered from unplanned failover (persistent configurations):

   a. On AIX: **chdev -l en0 -a delalias4=<IP>,<NetMask>**
   b. On Linux: **ifconfig eth0:0 0.0.0.0 0.0.0.0**

3.5 Component configuration listing

This section describes the commands that are needed to list configurations for each specific component. The objective is to help users focus on key configuration locations when troubleshooting IBM Power Systems 775 HPC clusters. Table 3-9 summarizes the available configuration listing commands.

Commands overview

Table 3-9 Available commands for component configuration listing

Component	Command (AIX/Linux)	Description
LoadLeveler	**llconfig**	Manages (lists also) the LoadLeveler database configuration.
	llctl	Controls LoadLeveler daemons.
	llctl ckconfig	Checks the configuration file or database configuration for errors.
	cat /etc/LoadL.cfg	User and Database master configuration file.
GPFS	**gpfs.snap**	Gathers GPFS configuration that is needed for support (does not send to support).
	mmlscluster	Displays the current configuration data for a GPFS cluster.
	mmlsconfig	Displays the configuration data for a GPFS cluster.
	mmlsfileset (check)	Displays status and attributes of GPFS filesets.
	mmlslicense	Displays information about the GPFS node licensing designation.
	mmlssnapshot	Displays GPFS snapshot information for the specified file system.

Component	Command (AIX/Linux)	Description
xCAT	**lsdef**	Displays xCAT object definitions that are stored in xCAT database.
	lsxcatd	Queries xcatd daemon for database information.
	nodels	Lists the nodes, and their attributes, from the xCAT database.
	tabgrep	Lists table names in which an entry for the node appears.
	xcatsnap	Gathers xCAT configuration that is needed for support (does not send it to support).
DB2	cat /etc/inittab | grep db2	Lists if DB2 Fault monitor coordinator is configured to start in inittab file.
AIX and Linux Systems	See Table 3-10 on page 238	
ISNM	cat /etc/inittab | grep cnm	Lists if the CNM daemon is configured to start in inittab file.
	cat /etc/inittab | grep hdwr_svr	AIX only: Lists if the Hardware Server daemon is configured to start in inittab file.
	chkconfig --list | grep hdwr_svr	Linux only: Lists the init levels where the Hardware Server daemon is configured to start.
	lsnwconfig	Displays the active network management configuration parameters.
HFI (775 only)	**lslpp -L | grep -i hfi**	Displays the current HFI driver version installed.
	hfi.snap	Gathers HFI configuration that is needed for support (does not send to support). Default path: /var/adm/hfi/snaps
RSCT	**cat /etc/inittab | grep ctrmc**	AIX only: Lists if the RMC daemon is configured to start in inittab file.
	chkconfig --list | grep ctrmc	Linux only: Lists the init levels where the RMC daemon is configured to start.
PE Runtime Edition	rset_query	Displays information about the memory affinity assignments that are performed.
ESSL	NA	IBM Engineering and Scientific Subroutine Library for AIX and Linux on POWER.
Parallel ESSL	NA	Parallel Engineering and Scientific Subroutine Library for AIX.
Diskless resource (NFS)	cat /etc/exports	Displays information about the configured NFS exported directories.
Diskless resource (TFTP)	cat /etc/tftpaccess.ctl	AIX only: Displays information about the configured TFTP directories.

Component	Command (AIX/Linux)	Description
TEAL	cat /etc/teal/*.conf (gpfs.conf) (isnm.conf) (ll.conf) (pnsd.conf) (sfp.conf) (teal.conf)	Configuration files for TEAL subsystem and connectors.
	AIX: - /var/lock/teal.pid - /var/lock/teal_ll.pid Linux: - /var/locks/teal.pid - /var/locks/teal_ll.pid	Lock files for TEAL daemon and LoadLeveler connector, on AIX and Linux.

3.5.1 LoadLeveler

LoadLeveler uses file-based or database-based configuration. To determine which configuration is used, check the contents of the `/etc/LoadL.cfg` file.

For a file-based configuration, the value in the keyword LoadLConfig specifies the location of the global configuration file. We view the values in this file and the files that are specified in LOCAL_CONFIG and ADMIN_FILE keywords to see the configuration values.

For a database-based configuration, the value in the keyword LoadLDB specifies the database that contains the configuration tables. We use the **`llconfig`** command or the configuration editor to view the configuration values. When a database-based configuration is used, `/etc/LoadL.cfg` file on compute nodes contains the keyword LoadLConfigHosts, which indicates the nodes that include database access and are used to serve the configuration to this compute node. For more information, see Table 3-1 on page 158.

3.5.2 GPFS

We list the GPFS cluster configurations by using the GPFS commands that are shown in Table 3-9 on page 232. By using these commands, it is possible to collect data for support (gpfs.snap) and list the backups of GPFS configuration (**`mmlssnapshot`**), and license information (**`mmlslicense`**). For more information, see Table 3-1 on page 158.

3.5.3 xCAT

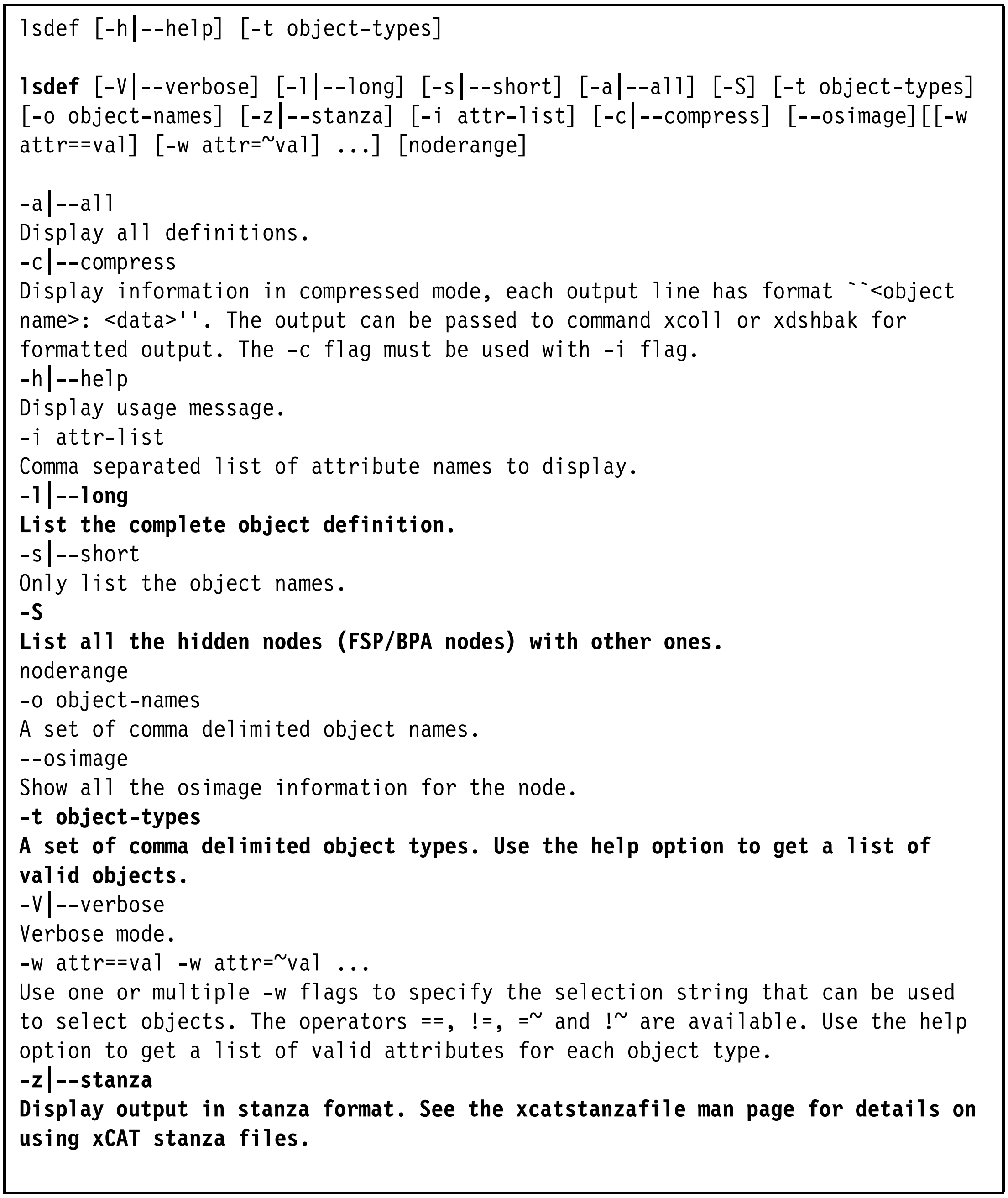
The xCAT internal database for an IBM Power Systems 775 cluster uses IBM DB2, which features commands that list and edit the configuration of the database. The more common commands are listed in Table 3-9 on page 232 (xCAT row). For more information, see Table 3-1 on page 158.

Examples and details are provided for the following commands:

- **`lsdef`**
- **`lsxcatd`**

lsdef

For this command, we check the man page description that is shown in Figure 3-29 on page 235. Typical output for this command is shown in Example 3-62 on page 236, and Example 3-63 on page 237.



```
lsdef [-h|--help] [-t object-types]

lsdef [-V|--verbose] [-l|--long] [-s|--short] [-a|--all] [-S] [-t object-types]
[-o object-names] [-z|--stanza] [-i attr-list] [-c|--compress] [--osimage][[-w
attr==val] [-w attr=~val] ...] [noderange]

-a|--all
Display all definitions.
-c|--compress
Display information in compressed mode, each output line has format ``<object
name>: <data>''. The output can be passed to command xcoll or xdshbak for
formatted output. The -c flag must be used with -i flag.
-h|--help
Display usage message.
-i attr-list
Comma separated list of attribute names to display.
-l|--long
List the complete object definition.
-s|--short
Only list the object names.
-S
List all the hidden nodes (FSP/BPA nodes) with other ones.
noderange
-o object-names
A set of comma delimited object names.
--osimage
Show all the osimage information for the node.
-t object-types
A set of comma delimited object types. Use the help option to get a list of
valid objects.
-V|--verbose
Verbose mode.
-w attr==val -w attr=~val ...
Use one or multiple -w flags to specify the selection string that can be used
to select objects. The operators ==, !=, =~ and !~ are available. Use the help
option to get a list of valid attributes for each object type.
-z|--stanza
Display output in stanza format. See the xcatstanzafile man page for details on
using xCAT stanza files.
```

Figure 3-29 lsdef man page description

The following input and output values are used:

- Command: **`lsdef`**
- Flags:
  - -v: Displays xCAT version information
  - -d: Displays xCAT database configuration
  - -a: Same as -v and -d together
- Outputs:

  [-v]
  Version <X.Y.Z> (svn <build>, built <date>)
  [-d]
  cfgloc=[DB2]:<xcat_user>|<xcat_password>
  dbengine=[DB2]
  dbinstance=<database_instance>
  dbname=<database_name>
  dbloc=<database_path>
- Designations: cfgloc: xCAT configuration type, user authorized to change it, and its password.
- Attributes:
  - <X.Y.Z>: Version of xCAT
  - <built>: SVN build number
  - <date>: Build date
  - <xcat_user>: Configuration user
  - <xcat_password>: Configuration user password
  - <database_instance>: Database instance name
  - <database_name>: Database name
  - <database_path>: Database path

Example 3-62 lsdef output example from "site" table

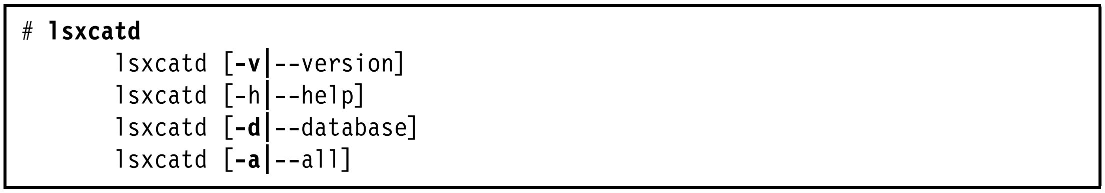
```
# lsdef -t site -l
Object name: clustersite
    SNsyncfiledir=/var/xcat/syncfiles
    blademaxp=64
    cleanupxcatpost=no
    consoleondemand=yes
    databaseloc=/db2database
    db2installloc=/mntdb2
    dhcpinterfaces=en2
    dnshandler=ddns
    domain=ppd.pok.ibm.com
    enableASMI=no
    fsptimeout=0
    installdir=/install
    master=192.168.0.103
    maxssh=8
    nameservers=192.168.0.103
    ntpservers=192.168.0.103
    ppcmaxp=64
    ppcretry=3
    ppctimeout=0
    sharedtftp=1
    sshbetweennodes=ALLGROUPS
    teal_ll_ckpt=0
    tftpdir=/tftpboot
    timezone=EST5EDT
    topology=8D
    useNmapfromMN=no
    useSSHonAIX=yes
    vsftp=y
    xcatconfdir=/etc/xcat
    xcatdport=3001
    xcatiport=3002
```

Example 3-63 lsdef command output example without flags

```
# lsdef
c250f10c12ap01  (node)
c250f10c12ap02-hf0  (node)
c250f10c12ap05-hf0  (node)
c250f10c12ap09-hf0  (node)
c250f10c12ap13-hf0  (node)
c250f10c12ap17  (node)
c250f10c12ap18-hf0  (node)
c250f10c12ap21-hf0  (node)
c250f10c12ap25-hf0  (node)
c250f10c12ap29-hf0  (node)
cec12  (node)
frame10  (node)
llservice  (node)
```

lsxcatd

For this command, we list the help description that is shown in Figure 3-30. A typical output is shown in Example 3-64 on page 238.



```
# lsxcatd
       lsxcatd [-v|--version]
       lsxcatd [-h|--help]
       lsxcatd [-d|--database]
       lsxcatd [-a|--all]
```

Figure 3-30 lsxcatd command flag description

The following input and output values are used:

- Command: **`lsxcatd -a`**
- Flags:
  - -v: Displays xCAT version information
  - -d: Displays xCAT database configuration
  - -a: Same as -v and -d together
- Outputs:

  [-v]
  Version <X.Y.Z> (svn <build>, built <date>)
  [-d]
  cfgloc=[DB2]:<xcat_user>|<xcat_password>
  dbengine=[DB2]
  dbinstance=<database_instance>

dbname=<database_name>
dbloc=<database_path>

- Designations: cfgloc: xCAT configuration type, user authorized to change it, and its password
- Attributes:
  - <X.Y.Z>: Version of xCAT
  - <built>: SVN build number
  - <date>: Build date
  - <xcat_user>: Configuration user
  - <xcat_password>: Configuration user password
  - <database_instance>: Database instance name
  - <database_name>: Database name
  - <database_path>: Database path

Example 3-64 Listing the xCAT database configuration

```
# lsxcatd -a
Version 2.6.8 (svn r10523, built Wed Sep 14 09:06:44 EDT 2011)
cfgloc=DB2:xcatdb|xcatdb
dbengine=DB2
dbinstance=xcatdb
dbname=xcatdb
dbloc=/db2database/db2
```

3.5.4 DB2

There is no specific configuration for the IBM Power Systems 775 cluster that requires monitoring attention. However, if such monitoring is required, there is more documentation for DB2 settings as shown in Table 3-1 on page 158.

3.5.5 AIX and Linux systems

Table 3-10 lists the configuration details for specific areas that are related to the IBM Power System 775 clusters, such as devices, pci cards, scsi cards, or logical device driver configurations. For more information, see Table 3-1 on page 158.

Table 3-10 AIX and Linux system configuration commands equivalencies

AIX	Linux	Command Description
lsdev	-	Lists device status information.
-	lspci	Lists information about PCI device locations.
-	lsscsi	Lists information about detected SCSI devices.
lscfg	lscfg	Lists hardware configuration.

Important: Although some commands feature the same syntax and purpose, their internal arguments and flags might differ for AIX and Linux.

3.5.6 Integrated Switch Network Manager

In this section, we check which files are monitored for troubleshooting or debugging tasks. Table 3-11 on page 239 shows an overview of the logging files for the CNM and the hardware server. For more information, see Table 3-1 on page 158.

Table 3-11 Log files for CNM and hardware server - ISNM components

Directory (AIX)	ISNM Component	Description
/var/opt/isnm/cnm/fnmlock.lck	CNM	Lock file for CNM daemon.

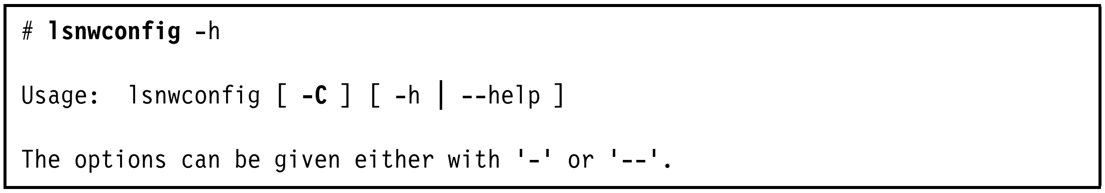
/var/opt/isnm/cnm/data/CNM_DEBUG_START_FILE		CNM debug file.
		Options:
		MULTICAST_ENABLEMENT=[0 I 1 I 2]
		DISABLE_TEAL_CONNECTION=[0 I 1]

/var/opt/isnm/cnm/log		CNM log and snap directory.
/var/opt/isnm/cnm/log/eventSummary.log		Logs CNM events for HFI Network.
/var/opt/isnm/hdwr_svr/data/HmcNetConfig	Hardware Server	Hardware connections output file. Required for `lshwconn`.
/var/opt/isnm/hdwr_svr/log/hdwr_svr.log		Hardware Server log file.

In this section, we describe the **`lsnwconfig`** command.

lsnwconfig

For this command, we list the help description that is shown in Figure 3-31. A typical output is shown in Example 3-65 on page 240.



```
# lsnwconfig -h

Usage:  lsnwconfig [ -C ] [ -h | --help ]

The options can be given either with '-' or '--'.
```

Figure 3-31 lsnwconfig command flag description

The following input and output values are used:

- Command: **`lsnwconfig`**
- **Flags:** -C → compact style
- Outputs:
  - ISNM Configuration parameter values from Cluster Database
  - Performance Data Interval: 300 seconds
  - Performance Data collection Save Period: 168 hours
  - No.of Previous Performance Summary Data: 1
  - RMC Monitoring Support: ON (1)
  - CNM Expired Records Timer Check: 3600 seconds
  - CNM Summary Data Timer: 43200 seconds
  - CNM Recovery Consolidation timer: 300 seconds
  - [-C]

- ISNM Configuration Parameter values that are used by CNM
- CNM Expired Records Timer Check: 3600 seconds
- RMC Monitoring Support: ON (1)
- No.of Previous Performance Summary Data: 1
- Performance Data Interval: 300 seconds
- Performance Data collection Save Period: 168 hours
- CNM Recovery Consolidation timer: 300 seconds
- CNM Summary Data Timer: 43200 seconds

Example 3-65 lsnwconfig output example

```
# lsnwconfig -C
ISNM Configuration Parameter values used by CNM
CNM Expired Records Timer Check: 3600 seconds
RMC Monitoring Support: ON (1)
No.of Previous Performance Summary Data: 1
Performance Data Interval: 300 seconds
Performance Data collection Save Period: 168 hours
CNM Recovery Consolidation timer: 300 seconds
CNM Summary Data Timer: 43200 seconds
```

3.5.7 HFI

The are no configuration files for HFI, only the installation files are listed, as shown in Table 3-9 on page 232.

3.5.8 Reliable Scalable Cluster Technology

For the AIX and Linux operating systems, the only subsystem startup configuration that must be checked is the RMC daemon. Example 3-66 for AIX and Example 3-67 for Linux shows how to check whether the RMC subsystem is configured to start at system startup. For more information, see Table 3-1 on page 158.

Example 3-66 Displaying AIX RMC startup configuration

```
# cat /etc/inittab |grep ctrmc
ctrmc:2:once:/usr/bin/startsrc -s ctrmc > /dev/console 2>&1
```

Example 3-67 Displaying Linux RMC startup configuration

```
# chkconfig --list ctrmc
ctrmc           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Linux configuration: For the Linux example, the init levels that must be turned on are the level 3 (full multiuser mode) and level 5 (same as level 3, but with X11 graphics).

3.5.9 Compilers environment

This section describes running workloads by using IBM LoadLeveler process and includes information that is related to PE Runtime Edition, ESSL, and Parallel ESSL. For more information, see Table 3-1 on page 158.

3.5.10 Diskless resources for NIM, iSCSI, NFS, and TFTP

Use Table 3-9 on page 232 for the configuration listing details for this section. For more information, see Table 3-1 on page 158.

3.6 Component monitoring examples

This section describes monitoring examples.

3.6.1 xCAT for power management, hardware discovery, and connectivity

If the expected state for a specific group is all nodes online and in a running state, Example 3-68 shows the following issues:

- The hostname is missing in `/etc/hosts`.
- Three LPARs are inactive (the operating system is not running).
- One LPAR is running but is not pinged.

Example 3-68 Three nodestat output problems

```
# nodestat lpar -p
c250f10c12ap18-hf0: Please make sure c250f10c12ap18-hf0 exists in /etc/hosts
c250f10c12ap01: sshd
c250f10c12ap02-hf0: sshd
c250f10c12ap05-hf0: sshd
c250f10c12ap09-hf0: sshd
c250f10c12ap13-hf0: sshd
c250f10c12ap17: noping(Running)
c250f10c12ap21-hf0: noping(Not Activated)
c250f10c12ap25-hf0: noping(Not Activated)
c250f10c12ap29-hf0: noping(Not Activated)
```

3.6.2 Integrated Switch Network Manager

Example 3-69 shows the **`lsnwloc`** output.

Example 3-69 lsnwloc command output

```
# lsnwloc
FR010-CG14-SN008-DR3 RUNTIME_CNM_EXCLUDED
```

4

Troubleshooting problems

In this chapter, we provide some examples of common problems and the methods that are used for troubleshooting. The problems might be discovered as a result of running a command or performing a monitoring procedure as described in Chapter 2, “Application integration” on page 99. Potential problems are also provided as a practical scenario in this chapter. Command outputs from an actual system are used in the illustrations that are presented here.

This chapter includes the following topics:

- xCAT
- ISNM
- HFI

4.1 xCAT

This section provides examples of common problems that might be encountered in an environment that uses xCAT. Information about how to resolve those issues on an IBM Power System 775 High Performance Computing (HPC) cluster also is presented. References to tools, websites, and documentation also are included.

For more information about general troubleshooting methods, see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Debugging_xCAT_Problems

4.1.1 xcatdebug

The xcatdebug script is used to trace the xcatd daemon, as shown in Example 4-1. The script runs xcatd with the Perl Debug-Trace and outputs the data from the parameters in the xcatd daemon subroutines that are specified in the input files in the `/opt/xcat/share/xcat/tools/tracelevel*` files. (For more information, see the xcatdebug man pages.)

The Perl Debug-Trace rpm is available in the latest xcat packages. More trace levels are created based on the nature of the problem that is being debugged.

Example 4-1 run xcatd that uses the `/opt/xcat/share/xcat/tools/tracelevel0` file

```
stop xcatd
xcatdebug 0
```

4.1.2 Resolving xCAT configuration issues

xCAT commands do not run if xCAT or DB2 is down. Complete the following steps to resolve problems that are found:

1. Verify that xcatd is running and that DB2 is accessible, as shown in Example 4-2.

Example 4-2 Verify xcatd and DB

```
# lssrc -s xcatd
Subsystem         Group            PID          Status
 xcatd                             18677838     active

# lsxcatd -d
cfgloc=DB2:xcatdb|xcatdb
dbengine=DB2
dbinstance=xcatdb
dbname=xcatdb
dbloc=/db2database/db2

# tabdump site
#key,value,comments,disable
"blademaxp","64",,
"fsptimeout","0",,
"installdir","/install",,
"master","192.168.0.103",,
"nameservers","192.168.0.103",,
"maxssh","8",,
"ppcmaxp","64",,
"ppcretry","3",,
"ppctimeout","0",,
"sharedtftp","1",,
"SNsyncfiledir","/var/xcat/syncfiles",,
"tftpdir","/tftpboot",,
"xcatdport","3001",,
"xcatiport","3002",,
"xcatconfdir","/etc/xcat",,
"timezone","EST5EDT",,
"useNmapfromMN","no",,
"enableASMI","no",,
"db2installloc","/mntdb2",,
"databaseloc","/db2database",,
"sshbetweennodes","ALLGROUPS",,
"dnshandler","ddns",,
"vsftp","y",,
"cleanupxcatpost","no",,
"useSSHonAIX","yes",,
"consoleondemand","yes",,
"domain","ppd.pok.ibm.com",,
"ntpservers","192.168.0.103",,
"teal_ll_ckpt","0","teal_ll checkpoint - DO NOT DELETE OR MODIFY",
"dhcpinterfaces","en2",,
"topology","8D",,
```

If there are security SSH or xCAT keys or certificates problems, proceed with the following steps:

2. Verify whether **xdsh <node> date** runs without prompting for password.
3. Run the **updatenode -k** command to update the keys or certificates on the nodes as shown in Example 4-3.

*Example 4-3 To update the security keys for the node "*c250f06c10ap01*"*

```
# updatenode c250f06c10ap01 -V -k
Enter the password for the userid: root on the node where the ssh keys
will be updated:

Running command on c250mgrs38-pvt: /bin/hostname 2>&1

Running command on c250mgrs38-pvt: ifconfig -a| grep "inet" 2>&1

Running internal xCAT command: makeknownhosts ...

Running command on c250mgrs38-pvt: cat //.ssh/known_hosts >
//.ssh/known_hosts.backup 2>&1

Running command on c250mgrs38-pvt: /bin/sed -e "/^c250f06c10ap01[,| ]/d;
/^10.6.10.1[,| ]/d;" //.ssh/known_hosts > /tmp/7013302 2>&1

Running command on c250mgrs38-pvt: cat /tmp/7013302 > //.ssh/known_hosts 2>&1

Running command on c250mgrs38-pvt: rm -f /tmp/7013302 2>&1
```

```
  c250mgrs38-pvt: run makeknownhosts to clean known_hosts file for nodes:
c250f06c10ap01
Running internal xCAT command: xdsh ...

Running command on c250mgrs38-pvt: /bin/hostname 2>&1

Running command on c250mgrs38-pvt: ifconfig -a| grep "inet" 2>&1

  c250mgrs38-pvt: Internal call command: xdsh -K. nodes = c250f06c10ap01,
arguments = -K, env = DSH_FROM_USERID=root DSH_TO_USERID=root XCATROOT=/opt/xcat
DSH_REMOTE_PASSWORD=ppslab XCATCFG=DB2:xcatdb|xcatdb|db2root
  c250mgrs38-pvt: return messages of last command: Running command on
c250mgrs38-pvt: /opt/xcat/sbin/remoteshell.expect -k 2>&1 Copying SSH Keys Running
command on c250mgrs38-pvt:  cp //.ssh/id_rsa.pub
/install/postscripts/_ssh/authorized_keys 2>&1  cp //.ssh/id_rsa.pub
/install/postscripts/_ssh/authorized_keys succeeded. Running command on
c250mgrs38-pvt:  cp //.ssh/id_rsa.pub //.ssh/tmp/authorized_keys 2>&1  cp
//.ssh/id_rsa.pub //.ssh/tmp/authorized_keys succeeded. Running command on
c250mgrs38-pvt:  cp //.ssh/copy.sh /install/postscripts/_ssh/copy.sh 2>&1
/usr/bin/ssh setup is complete. return code = 0
c250f06c10ap01: Setup ssh keys has completed.
  c250mgrs38-pvt: Internal call command: XCATBYPASS=Y /opt/xcat/bin/xdsh
c250f06c10ap01 -s -v -e /install/postscripts/xcataixpost -m 10.0.0.138 -c 5
'aixremoteshell,servicenode' 2>&1
c250f06c10ap01: Redeliver certificates has completed.
Running internal xCAT command: makeknownhosts ...
Running command on c250mgrs38-pvt: cat //.ssh/known_hosts >
//.ssh/known_hosts.backup 2>&1

Running command on c250mgrs38-pvt: /bin/sed -e "/^c250f06c10ap01[,| ]/d;
/^10.6.10.1[,| ]/d;" //.ssh/known_hosts > /tmp/7013302 2>&1
Running command on c250mgrs38-pvt: cat /tmp/7013302 > //.ssh/known_hosts 2>&1

Running command on c250mgrs38-pvt: rm -f /tmp/7013302 2>&1
```

4.1.3 Node does not respond to queries or `rpower` command

If a node does not respond to queries from the xCAT commands (for example, **`rpower`**), complete the following steps to check the service network status and the hardware connection between XCAT and the hardware:

1. Check the syslog at `/var/log/messages` on the management node.

   By default, the syslog is written into the `/var/log/messages` file. If the syslog PostScript runs correctly for the node, the syslog message of the node is redirected to the management node.

2. Verify the hardware connections between xCAT, the FSPs (CECs), and the BPAs (frames).

The `/var/opt/isnm/hdwr_svr/data/HmcNetConfig` file must contain two hardware connection types for each BPA and each FSP. xCAT uses a connection type (**`-tooltype`**) of lpar and CNM uses a connection type of CNM.

The **`lshwconn`** command is used to list and show the status of the hardware connections that are defined in `/var/opt/isnm/hdwr_svr/data/HmcNetConfig`. Example 4-4 on page 247 and

Example 4-5 show the expected status of FSPs and BPAs, which are included in the xCAT CEC (FSP) and frame (BPA) groups.

Example 4-4 Verify hwconn status of FSP

```
# lshwconn cec
f06cec09: 40.6.9.1: sp=secondary,ipadd=40.6.9.1,alt_ipadd=unavailable,state=LINE UP
f06cec09: 40.6.9.2: sp=primary,ipadd=40.6.9.2,alt_ipadd=unavailable,state=LINE UP
f06cec04: 40.6.4.2: sp=primary,ipadd=40.6.4.2,alt_ipadd=unavailable,state=LINE UP
f06cec04: 40.6.4.1: sp=secondary,ipadd=40.6.4.1,alt_ipadd=unavailable,state=LINE UP
f06cec12: 40.6.12.2: sp=primary,ipadd=40.6.12.2,alt_ipadd=unavailable,state=LINE UP
f06cec12: 40.6.12.1: sp=secondary,ipadd=40.6.12.1,alt_ipadd=unavailable,state=LINE UP
f06cec05: 40.6.5.2: sp=primary,ipadd=40.6.5.2,alt_ipadd=unavailable,state=LINE UP
f06cec05: 40.6.5.1: sp=secondary,ipadd=40.6.5.1,alt_ipadd=unavailable,state=LINE UP
f06cec06: 40.6.6.2: sp=primary,ipadd=40.6.6.2,alt_ipadd=unavailable,state=LINE UP
f06cec06: 40.6.6.1: sp=secondary,ipadd=40.6.6.1,alt_ipadd=unavailable,state=LINE UP
f06cec08: 40.6.8.1: sp=secondary,ipadd=40.6.8.1,alt_ipadd=unavailable,state=LINE UP
f06cec08: 40.6.8.2: sp=primary,ipadd=40.6.8.2,alt_ipadd=unavailable,state=LINE UP
f06cec07: 40.6.7.2: sp=primary,ipadd=40.6.7.2,alt_ipadd=unavailable,state=LINE UP
f06cec07: 40.6.7.1: sp=secondary,ipadd=40.6.7.1,alt_ipadd=unavailable,state=LINE UP
f06cec10: 40.6.10.1: sp=secondary,ipadd=40.6.10.1,alt_ipadd=unavailable,state=LINE UP
f06cec10: 40.6.10.2: sp=primary,ipadd=40.6.10.2,alt_ipadd=unavailable,state=LINE UP
f06cec02: 40.6.2.1: sp=secondary,ipadd=40.6.2.1,alt_ipadd=unavailable,state=LINE UP
f06cec02: 40.6.2.2: sp=primary,ipadd=40.6.2.2,alt_ipadd=unavailable,state=LINE UP
f06cec11: 40.6.11.2: sp=primary,ipadd=40.6.11.2,alt_ipadd=unavailable,state=LINE UP
f06cec11: 40.6.11.1: sp=secondary,ipadd=40.6.11.1,alt_ipadd=unavailable,state=LINE UP
f06cec01: 40.6.1.1: sp=secondary,ipadd=40.6.1.1,alt_ipadd=unavailable,state=LINE UP
f06cec01: 40.6.1.2: sp=primary,ipadd=40.6.1.2,alt_ipadd=unavailable,state=LINE UP
f06cec03: 40.6.3.2: sp=primary,ipadd=40.6.3.2,alt_ipadd=unavailable,state=LINE UP
f06cec03: 40.6.3.1: sp=secondary,ipadd=40.6.3.1,alt_ipadd=unavailable,state=LINE UP
```

Example 4-5 Verify the hwconn status of BPA

```
# lshwconn frame
frame06: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
frame06: 40.6.0.2: side=b,ipadd=40.6.0.2,alt_ipadd=unavailable,state=LINE UP
```

For more information about hardware connections, and the steps that are needed to correct missing BPA hardware connections for xCAT and CNM, see 4.2.3, “Adding hardware connections” on page 253.

4.1.4 Node fails to install

If issues are encountered during the installation of a node, complete the following steps:

1. Check the validation of the following attributes in the tables:

   a. Table Site: Master, domain, nameserver, dhcpinterfaces
   b. Table networks: Has the installation network defined
   c. Table noderes: netboot, tftpserver, nfsserver, installnic, primarynic
   d. Table nodetype: os, arch, profile

2. Verify that multiple DHCP are not in one network/vlan.

   Unpredictable problems occur when more than one DHCP server is on the network. The EMS must be the only DHCP server on the network.

3. Confirm that the Linux packages listed in the `otherpkgs` file are not installed on the node:

   a. Ensure that the following path and name of the `otherpkgs` configuration file are correct: `/opt/xcat/share/xcat/netboot(install)/<platform>/profile.<os>.<arch>.otherpkgs.pkglist`
   b. Ensure that the packages are copied to following correct path: `/install/post/otherpkgs/<os>/<arch>/xcat`
4. Check that the AIX packages in `installp_bundle/otherpkgs` are not installed on the node:
   a. Ensure the `installp_bundle` or `otherpkgs` are configured to the NIM image.
   b. Write into table NIM image.
   c. Set as the parameter for '**nimnodeset**' command.
   d. Ensure the `installp` and rpm packages are updated to the NIM image.

4.1.5 Unable to open a remote console

The **rcons** command is used to open a remote console to a node to monitor the installation progress. If **rcons** fails to open the remote console, the issue might be that conserver is not configured.

Complete the following steps to monitor the network installation of a node:

1. Configure conserver by using the following command:

```
# makeconservercf
```

2. Open a remote console to the node by using the following command:

```
# rcons <nodename>
```

3. Initiate a netboot of the node by using the following command:

```
# rnetboot <nodename>
```

Important: You must run **makeconservercf** after the cluster nodes are defined and before **rcons** are run.

In addition, you use the AIX **lsnim** command to see the state of the NIM installation for a particular node by running the following command on the NIM master:

```
# lsnim -l <nodename>
```

4.1.6 Time out errors during network boot of nodes

To initiate a network boot of all nodes in the group “compute”, you use the following command:

```
# rnetboot compute
```

If you receive timeout errors from the **rnetboot** command, you might need to increase the default timeout from 60 seconds to a larger value by setting the ppctimeout in the site table, such as 180 seconds, as shown in the following example:

```
# chdef -t site -o clustersite ppctimeout=180
```

4.2 ISNM

This section describes some problem scenarios that might be encountered in the areas of Integrated Switch Network Manager (ISNM) and Central Network Manage (CNM). This section also provides a practical understanding of the two types of hardware connections between the various components: the topology and the communication paths. Several CNM commands are shown in examples to identify and correct common CNM issues.

4.2.1 Checking the status and recycling the hardware server and the CNM

There are instances in which it is necessary to manually stop, start, or recycle the hdwr_svr and the CNM, as shown in Example 4-6. For example, when a change or update is implemented on the cluster.

Example 4-6 Confirming status of hdwr_svr and cnm

```
# ps -ef | grep isnm
root 6750532       1   0   Oct 12      - 34:54 /opt/isnm/hdwr_svr/bin/hdwr_svr
-Dhcdaemon -Dfsp run_hmc_only
root 7668424       1   0   Oct 14      - 93:05 /opt/isnm/cnm/bin/cnmd
```

To recycle the hdwr_svr, use the following kill the hdwr_svr process that is identified in Example 4-6:

```
# kill 6750532
```

Restart the following hdwr_svr:

```
# /opt/isnm/hdwr_svr/bin/hdwr_svr -Dhcdaemon -Dfsp run_hmc_only &
```

To recycle CNM, issue the following command:

```
# chnwm -d
```

As shown in the following example, CNMD is deactivated successfully:

```
# chnwm -a
```

If CNM is deactivated, it must be activated so that the threads come up in timely fashion and the daemon is running. Confirm that hdwr_svr and CNM are running as shown in the following example:

```
# ps -ef | grep isnm
root 6619386       1   0 15:15:41      -  0:03 /opt/isnm/cnm/bin/cnmd
root 3539474       1   0 15:06:24      -  4:25 /opt/isnm/hdwr_svr/bin/hdwr_svr
-Dhcdaemon -Dfsp run_hmc_only
```

> **Important:** The hdwr_svr and CNM processes are stopped, started, and recycled independently.

4.2.2 Communication issues between CNM and DB2

Monitoring the /var/opt/isnm/cnm/log/eventSummary.log file is a quick and effective way to identify potential problems with CNM communications or general configuration issues. This feature is demonstrated throughout this chapter. The following high-level steps are used to debug the CNM startup sequence:

1. Identify that there is an issue via the /var/opt/isnm/cnm/log/eventSummary.log file.
2. Isolate the cause of the problem and make the necessary changes.
3. Recycle CNM by issuing **chnwm -d** and then **chnwm -a**.
4. Check the /var/opt/isnm/cnm/log/eventSummary.log to see whether CNM starts without errors, as shown in Example 4-7.

Example 4-7 Checking /var/opt/isnm/cnm/log/eventSummary.log for problems

```
# tail /var/opt/isnm/cnm/log/eventSummary.log
13:19:23.283143  cnm_glob.cpp:1031: CNM Daemon is starting now
13:19:24.485371  cnm_glob.cpp:261: Unable to Connect to XCATDB
13:19:24.490633  cnm_glob.cpp:277: Unable to get configuration parameters from
isnm_config table
13:19:24.492224  cnm_glob.cpp:313: Unable to get cluster information from XCATDB
```

This section illustrates some scenarios of errors reported when CNM is started. One reason for these messages might be a missing parent entry in the PPC table. If the parent entry is missing, we check the PPC table for missing parents, as shown in Example 4-8.

Example 4-8 Using the nodels command to check for missing parents in the PPC table

```
# nodels all ppc.parent -S
40.6.0.1: frame06
40.6.0.2: frame06
40.6.1.1: f06cec01
40.6.1.2: f06cec01
40.6.10.1: f06cec10
40.6.10.2: f06cec10
40.6.11.1: f06cec11
40.6.11.2: f06cec11
40.6.12.1: f06cec12
40.6.12.2: f06cec12
40.6.2.1: f06cec02
40.6.2.2: f06cec02
40.6.3.1: f06cec03
40.6.3.2: f06cec03
40.6.4.1: f06cec04
40.6.4.2: f06cec04
40.6.5.1: f06cec05
40.6.5.2: f06cec05
40.6.6.1: f06cec06
40.6.6.2: f06cec06
40.6.7.1: f06cec07
40.6.7.2: f06cec07
40.6.8.1: f06cec08
40.6.8.2: f06cec08
40.6.9.1: f06cec09
40.6.9.2: f06cec09
```

```
c250hmc10:
f06cec01: frame06
f06cec02: frame06
f06cec03: frame06
f06cec04: frame06
f06cec05: frame06
f06cec06: frame06
f06cec07: frame06
f06cec08: frame06
f06cec09: frame06
f06cec10: frame06
f06cec11:
f06cec12:
frame06:
```

> **Output of the `nodels` command:** The output of the **`nodels`** command shows a few entries without associated parents. In some cases, this output is expected; for example, hmc (c250hmc10) and frame (frame06). However, the f06cec11 and f06cec12 entries are missing their parent (frame06) and they must be corrected, as shown in Example 4-9. For more information, see the man pages for the **`chdef`** command.

Example 4-9   Using chdef to add missing parents to the PPC table

```
Adding the missing parent (frame06) for f06cec11 and f06cec12 to the ppc table:

# chdef f06cec11,f06cec12 parent=frame06

Recycle CNM and verify that it starts correctly:

# tail  /var/opt/isnm/cnm/log/eventSummary.log
09:00:10.827715  cnm_glob.cpp:1049: CNM Daemon is starting now

A second scenario with the same generic symptom is described in Example 4-7 on
page 250, but caused by a different issue:

# tail  /var/opt/isnm/cnm/log/eventSummary.log
13:19:23.283143  cnm_glob.cpp:1031: CNM Daemon is starting now
13:19:24.485371  cnm_glob.cpp:261: Unable to Connect to XCATDB
13:19:24.490633  cnm_glob.cpp:277: Unable to get configuration parameters from
isnm_config table
13:19:24.492224  cnm_glob.cpp:313: Unable to get cluster information from XCATDB
```

If the Open Database Connectivity (ODBC) setup is incorrect, for instance, if some files in the the `/var/lib/db2/sqllib/lib/` directory are invalid or missing, ODBC and SQL cannot make the required connection and CNM cannot receive information from DB2, as shown in Example 4-10.

For more information about ODBC, see this website:

`http://en.wikipedia.org/wiki/ODBC`

Example 4-10   Using `/usr/local/bin/isql-v xcatdb` to check the ODBC setup

```
# /usr/local/bin/isql -v xcatdb
```

If the ODBC setup is correct and the connection to SQL is successful, the command returns the output that is shown in Example 4-11.

Example 4-11 ODBC correct setup

```
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL> quit
```

The error scenario that is shown in Example 4-11 results in the following output:

```
[01000][unixODBC][Driver Manager]Can't open lib
'/var/lib/db2/sqllib/lib/libdb2.so' : file not found
[ISQL]ERROR: Could not SQLConnect
```

This message indicates a problem with the files in the `/var/lib/db2/sqllib/lib/libdb2.so` directory. Confirm that the correct files are available, then restart CNM and verify that the CNM starts without posting the previous errors.

Another scenario of a CNM to DB2 communication issue with slightly different messages posted in `/var/opt/isnm/cnm/log/eventSummary.log` is shown in Example 4-12.

Example 4-12 CNM to DB2 communication scenario case

```
# tail  /var/opt/isnm/cnm/log/eventSummary.log
13:19:23.283143 cnm_glob.cpp:1049: CNM Daemon is starting now
13:19:23.283143 cnm_glob.cpp:400: Unable to create instance of XCATDB ppc table
13:19:23.283143 cnm_glob.cpp:312: Unable to get cluster information from XCATDB
13:22:08.372292  cnm_glob.cpp:1031: DB tread is starting now
```

When the EMS is rebooted, CNM might start before DB2 is ready, and this results in the error messages that are described in Example 4-12. To resolve this problem, run the commands that are shown in Example 4-13.

Example 4-13 Restarting and synchronizing DB2 when CNM is started before DB2 is ready

```
# su xcatdb
# db2
This takes you to the db2 prompt:

db2 => connect to xcatdb
db2 => db2stop force
db2 => db2start
```

Recycle CNM and verify that it starts without the previous errors.

4.2.3 Adding hardware connections

We consider a problem in which one of the BPAs in a frame is not responding to CNM and xCAT as expected with the **lshwconn** command, as shown in Example 4-14.

Example 4-14 lshwconn shows missing BPA connections (A-side responds, B-side does not)

```
# lshwconn frame

frame06: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
frame06: 40.6.0.2: No connection information found for hardware control point
"40.6.0.2", please create the connection for this hardware control point firstly.

# lshwconn bpa -T lpar

40.6.0.2: 40.6.0.2: No connection information found for hardware control point
"40.6.0.2", please create the connection for this hardware control point firstly.
40.6.0.1: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP

# lshwconn bpa -T fnm

40.6.0.2: 40.6.0.2: No connection information found for hardware control point
"40.6.0.2", please create the connection for this hardware control point firstly.
40.6.0.1: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
```

As described in 4.1.3, "Node does not respond to queries or rpower command" on page 246, each BPA and CEC cage requires two hardware connections that are defined in `/var/opt/isnm/hdwr_svr/data/HmcNetConfig`. One of the connections is for xCAT and includes a connection type (-tooltype) of lpar. The other connection is for CNM and includes an FNM connection type. In this case, the hardware connection definitions for the BPA on the B-side of frame 78AC-100*992003H (IP@ 40.6.0.2) are missing, as shown in Example 4-15.

Example 4-15 Checking BPA connections in /var/opt/isnm/hdwr_svr/data/HmcNetConfig

```
# grep 78AC /var/opt/isnm/hdwr_svr/data/HmcNetConfig

BPC 40.6.0.1  -tooltype=lpar -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=2147483647 -mtms=78AC-100*992003H -slot=A -ignorepwdchg=n
BPC 40.6.0.1  -tooltype=fnm -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=2147483647 -mtms=78AC-100*992003H -slot=A -ignorepwdchg=n
```

We use the **mkhwconn** command to create the hardware connections for the BPA, as shown in Example 4-16.

Example 4-16 Using mkhwconn to create missing BPA hardware connections

```
# mkhwconn bpa -t -T lpar
# mkhwconn bpa -t -T fnm
```

After the hardware connections are created by issuing the **mkhwconn** commands in Example 4-16, confirm the BPA configuration as shown in Example 4-17 on page 254.

Example 4-17 Confirming the hardware connections

```
# lshwconn bpa -T lpar
40.6.0.2: 40.6.0.2: side=b,ipadd=40.6.0.2,alt_ipadd=unavailable,state=LINE UP
40.6.0.1: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
# lshwconn bpa -T fnm
40.6.0.1: 40.6.0.1: side=a,ipadd=40.6.0.1,alt_ipadd=unavailable,state=LINE UP
40.6.0.2: 40.6.0.2: side=b,ipadd=40.6.0.2,alt_ipadd=unavailable,state=LINE UP
# grep 78AC /var/opt/isnm/hdwr_svr/data/HmcNetConfig
BPC 40.6.0.1  -tooltype=lpar -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=2147483647 -mtms=78AC-100*992003H -slot=A -ignorepwdchg=n
BPC 40.6.0.1  -tooltype=fnm -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=2147483647 -mtms=78AC-100*992003H -slot=A -ignorepwdchg=n
BPC 40.6.0.2  -tooltype=lpar -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=1319637503483 -mtms=78AC-100*992003H -slot=B
-ignorepwdchg=n
BPC 40.6.0.2  -tooltype=fnm -netc=y
-authtok=00F16D49F49205ED7B15322F8E88196C506235B4DA840D532A81CEDB7868B5E3B5CBF59B9
419613766C28BB0ED8F262F827E95CB3BAECDD8C5700F75EA058C5DF158AE26419C6E5255
-authtoklastupdtimestamp=1319637533657 -mtms=78AC-100*992003H -slot=B
-ignorepwdchg=n
```

4.2.4 Checking FSP status, resolving configuration or communication issues

The **lsnwloc** command provides a list of the FSPs and their states as seen by CNM (see Example 4-18).

Example 4-18 List of the FSPs and their states

```
# lsnwloc
FR006-CG04-SN051-DR1 RUNTIME
FR006-CG03-SN051-DR0 RUNTIME
FR006-CG13-SN005-DR2 RUNTIME
FR006-CG12-SN005-DR1 RUNTIME
FR006-CG10-SN004-DR3 RUNTIME
FR006-CG11-SN005-DR0 RUNTIME_CNM_EXCLUDED
FR006-CG05-SN051-DR2 RUNTIME
FR006-CG09-SN004-DR2 RUNTIME_CNM_UNREACHABLE
FR006-CG07-SN004-DR0 RUNTIME
FR006-CG08-SN004-DR1 RUNTIME
FR006-CG14-SN005-DR3 RUNTIME
FR006-CG06-SN051-DR3 RUNTIME
```

The following list of valid FSP states are returned from the **lsnwloc** command:

- STANDBY: FSP is powered on, LNMC is running, and the Power 775 CEC is not powered on.
- FUNCTIONAL_TORRENT: The Power 775 in the process of powering on.

- RUNTIME: The Power 775 is powered on, and the operating system is not necessarily booted.
- PENDINGPOWEROFF: The Power 775 has received a power-off command.
- LOW_POWER_IPL: Not implemented yet. Used during installation.
- RUNTIME_CNM_EXCLUDED: CNM is not used and the drawer mis-configured.
- STANDBY_CNM_EXCLUDED: CNM is not used and the drawer mis-configured.
- RUNTIME_CNM_UNREACHABLE: CNM has lost contact with the drawer.
- STANDBY_CNM_UNREACHABLE: CNM has lost contact with the drawer.

FSP status: The following FSP status indicates that a drawer mis-configuration exists (EXCLUDED) or that CNM lost contact with the drawer (UNREACHABLE):

```
RUNTIME_CNM_EXCLUDED or STANDBY_CNM_EXCLUDED
RUNTIME_CNM_UNREACHABLE or STANDBY_CNM_UNREACHABLE
```

An example of an FSP with EXCLUDED status is when the LNMC topology does not match the topology that is defined in the cluster DB and CNM, as described in 4.2.7, “Correcting inconsistent topologies” on page 257.

An FSP status of UNREACHABLE indicates a physical hardware problem on the drawer or FSP or possibly an LNMC configuration issue with the FSP.

4.2.5 Verifying CNM to FSP connections

The **`lsnwcomponents`** command displays the BPAs and FSPs that are reachable via the service network, as shown in Example 4-19.

Example 4-19 Displaying the BPAs and FSPs

```
# lsnwcomponents
BPA Primary ip=40.6.0.1 MTMS=78AC-100*992003H FR006
BPA Backup  ip=40.6.0.2 MTMS=78AC-100*992003H FR006
FSP Primary ip=40.6.4.2 MTMS=9125-F2C*02C6946 FR006-CG06-SN051-DR3
FSP Backup  ip=40.6.9.1 MTMS=9125-F2C*02C6A26
FSP Primary ip=40.6.12.2 MTMS=9125-F2C*02C6A86 FR006-CG14-SN005-DR3
FSP Primary ip=40.6.6.2 MTMS=9125-F2C*02C69B6 FR006-CG08-SN004-DR1
FSP Backup  ip=40.6.8.1 MTMS=9125-F2C*02C6A06
FSP Primary ip=40.6.5.2 MTMS=9125-F2C*02C6986 FR006-CG07-SN004-DR0
FSP Primary ip=40.6.7.2 MTMS=9125-F2C*02C69D6 FR006-CG09-SN004-DR2
FSP Primary ip=40.6.2.2 MTMS=9125-F2C*02C68D6 FR006-CG04-SN051-DR1
FSP Primary ip=40.6.1.2 MTMS=9125-F2C*02C68B6 FR006-CG03-SN051-DR0
FSP Backup  ip=40.6.11.1 MTMS=9125-F2C*02C6A66
FSP Backup  ip=40.6.3.1 MTMS=9125-F2C*02C6906
FSP Backup  ip=40.6.10.1 MTMS=9125-F2C*02C6A46
FSP Backup  ip=40.6.2.1 MTMS=9125-F2C*02C68D6
FSP Backup  ip=40.6.1.1 MTMS=9125-F2C*02C68B6
FSP Primary ip=40.6.11.2 MTMS=9125-F2C*02C6A66 FR006-CG13-SN005-DR2
FSP Primary ip=40.6.3.2 MTMS=9125-F2C*02C6906 FR006-CG05-SN051-DR2
FSP Backup  ip=40.6.4.1 MTMS=9125-F2C*02C6946
FSP Primary ip=40.6.9.2 MTMS=9125-F2C*02C6A26 FR006-CG11-SN005-DR0
FSP Backup  ip=40.6.12.1 MTMS=9125-F2C*02C6A86
FSP Backup  ip=40.6.6.1 MTMS=9125-F2C*02C69B6
FSP Primary ip=40.6.8.2 MTMS=9125-F2C*02C6A06 FR006-CG10-SN004-DR3
FSP Backup  ip=40.6.5.1 MTMS=9125-F2C*02C6986
FSP Backup  ip=40.6.7.1 MTMS=9125-F2C*02C69D6
FSP Primary ip=40.6.10.2 MTMS=9125-F2C*02C6A46 FR006-CG12-SN005-DR1
```

Important: The output of **`lsnwcomponents`** indicates that the cluster contains a single 78AC-100 frame FR006 with three supernodes: SN004 with 4x9125-F2C servers in cages 7-10, SN005 with 4x9125-F2C servers in cages 11-14, and SN051 with 4x9125-F2C servers in cages 3 - 6.

4.2.6 Verify that a multicast tree is present and correct

A missing or incorrect multicast tree causes communication issues on the service network, such as nodes that fail to communicate with or ping each other. Verify the presence and integrity of the multicast tree, as shown in Example 4-20.

Example 4-20 Verifying the multicast tree

```
# cat /var/opt/isnm/cnm/log/mctree.out

*********************************************************
    Multicast tree starts with root
*********************************************************
Root: FR006-CG07-SN004-DR0-HB3
1: FR006-CG07-SN004-DR0-HB3-1-->FR006-CG07-SN004-DR0-HB0-1
2: FR006-CG07-SN004-DR0-HB0-2-->FR006-CG07-SN004-DR0-HB5-1
3: FR006-CG07-SN004-DR0-HB5-2-->FR006-CG07-SN004-DR0-HB2-4
3: FR006-CG07-SN004-DR0-HB5-3-->FR006-CG07-SN004-DR0-HB4-6
2: FR006-CG07-SN004-DR0-HB0-3-->FR006-CG07-SN004-DR0-HB1-3
3: FR006-CG07-SN004-DR0-HB1-2-->FR006-CG07-SN004-DR0-HB6-4
3: FR006-CG07-SN004-DR0-HB1-7-->FR006-CG07-SN004-DR0-HB7-7
2: FR006-CG07-SN004-DR0-HB0-42-->FR006-CG11-SN005-DR0-HB0-43
3: FR006-CG11-SN005-DR0-HB0-1-->FR006-CG11-SN005-DR0-HB3-1
3: FR006-CG11-SN005-DR0-HB0-2-->FR006-CG11-SN005-DR0-HB5-1
3: FR006-CG11-SN005-DR0-HB0-3-->FR006-CG11-SN005-DR0-HB1-3
3: FR006-CG11-SN005-DR0-HB0-4-->FR006-CG11-SN005-DR0-HB4-1
3: FR006-CG11-SN005-DR0-HB0-5-->FR006-CG11-SN005-DR0-HB2-2
3: FR006-CG11-SN005-DR0-HB0-6-->FR006-CG11-SN005-DR0-HB7-3
3: FR006-CG11-SN005-DR0-HB0-7-->FR006-CG11-SN005-DR0-HB6-7
3: FR006-CG11-SN005-DR0-HB0-8-->FR006-CG12-SN005-DR1-HB0-8
```

If the multicast tree is incorrect (for example, if it does not include all the expected cages), remove the file, re-ipl the CECs, and recycle CNM.

4.2.7 Correcting inconsistent topologies

The topology must be consistent across the Cluster DB, CNM and in LNMC (on the FSP of each node). Verify the topology by using the following **lsnwtopo** command (see Example 4-21 on page 257):

```
# lsnwtopo
```

The ISR network topology that is specified by the cluster configuration data is 8D.

```
# lsnwtopo -C
```

The ISR network topology in used by CNM is 8D.

Example 4-21 lsnwtopo -A showing an inconsistent topology in LNMC

```
# lsnwtopo -A
Frame 6 Cage 4 : Topology 8D, Supernode 51, Drawer 1
Frame 6 Cage 3 : Topology 8D, Supernode 51, Drawer 0
Frame 6 Cage 13 : Topology 8D, Supernode 5, Drawer 2
Frame 6 Cage 12 : Topology 8D, Supernode 5, Drawer 1
Frame 6 Cage 10 : Topology 8D, Supernode 4, Drawer 3
Frame 6 Cage 11 : Topology 128D, Supernode 5, Drawer 0 *RUNTIME-EXCLUDED
Frame 6 Cage 5 : Topology 8D, Supernode 51, Drawer 2
Frame 6 Cage 9 : Topology 8D, Supernode 4, Drawer 2
Frame 6 Cage 7 : Topology 8D, Supernode 4, Drawer 0
Frame 6 Cage 8 : Topology 8D, Supernode 4, Drawer 1
Frame 6 Cage 14 : Topology 8D, Supernode 5, Drawer 3
Frame 6 Cage 6 : Topology 8D, Supernode 51, Drawer 3
```

The outputs from these iterations of **lsnwtopo** indicate that the 8D topology is consistent across the Cluster DB, CNM, and LNMC on the FSPs, except for Frame 6 Cage 11, which shows: Frame 6 Cage 11: Topology 128D, Supernode 5, Drawer 0 *RUNTIME-EXCLUDED.

Example 4-22 on page 258 shows how to identify and correct topology inconsistencies.

Example 4-22 Correcting an inconsistent topology in the cluster DB and CNM

```
# lsnwtopo

   ISR network topology specified by cluster configuration data is 128D

# lsnwtopo -C
   ISR network topology in use by CNM: 128D

   In both cases, the expected topology is 8D not 128D. To correct the topology in
   the Cluster DB to 8D and then synchronize it with CNM, perform the following
   steps:

# chnwm -d
# chdef -t site topology=8D
# chnwm -a

Confirm that the topology has been updated to 8D via lsnwtopo and lsnwtopo -C

# lsnwtopo
   ISR network topology specified by cluster configuration data is 8D

# lsnwtopo -C

   ISR network topology in use by CNM: 8D
```

We consider a case in which the LNMC topology on a single cage or FSP does not match the topology of the other FSPs, and the topology that is defined in the Cluster DB and CNM. This mismatch is determined globally via the **lsnwtopo -A** command as described in Example 4-21. An incorrect LNMC topology is also identified via the /var/opt/isnm/cnm/log/eventSummary.log as shown in Example 4-23 on page 258.

Example 4-23 Identified incorrect LNMC topology

```
# tail /var/opt/isnm/cnm/log/eventSummary.log

11:00:26.124209  cnm_glob.cpp:1049: DB tread is starting now
11:00:26.141743  cnm_glob.cpp:1049: CNM Routing thread starting.
11:05:26.286466  comm_task.cpp:1765: MISMATCH: Expected frame 6 cage 11 supernode
5 drawer 0 topology 4 numDlink 128, but received frame 6 cage 11 supernode 5
drawer 0 topology 64 numDlink 8 Master id valid bit 1 from LNMC
11:10:26.094019  comm_task.cpp:1765: MISMATCH: Expected frame 6 cage 11 supernode
5 drawer 0 topology 4 numDlink 128, but received frame 6 cage 11 supernode 5
drawer 0 topology 64 numDlink 8 Master id valid bit 1 from LNMC
```

As shown in Example 4-23, when CNM is recycled, a topology mismatch is reported and continues to be reported every 5 minutes until the topology is corrected in LNMC on the FSP of the identified cage.

LNMC topology: The LNMC topology on frame 6 cage 11 is confirmed by issuing the following **lsnwtopo** command:

```
   # lsnwtopo -f 6 -c 11
   Frame 6 Cage 11 : Topology 128D, Supernode 5, Drawer 0
```

The LNMC topology of the specified cage or FSP must be changed from 128D to 8D. To make the change, the cage must be Standby (FSP powered on with LNMC running, and the Power 775 not powered on). Complete the following steps to update the topology on the cage:

1. Check that frame 6 cage 11 is at Standby by powering off the cage (if needed) and then stopping CNM by using the following command:

   ```
   # chnwm -d
   ```

2. If the topology in the Cluster DB and CNM must be updated, the **`chdef`** command must be issued before CNM is restarted. If the topology is correct in the Cluster DB and CNM, restart CNM and then update the topology to LNMC by issuing the following **`chnwsvrconfig`** command:

   ```
   # chnwm -a
   # chnwsvrconfig -f 6 -c 11
   ```

3. Power on the cage and confirm that all topologies show the wanted values by issuing the following commands: **`lsnwtopo`**, **`lsnwtopo -C`** and **`lsnwtopo -A`** (specifically in this case: **`lsnwtopo -f 6 -c 11`**):

   ```
   # lsnwtopo
   ```

   ISR network topology that is specified by cluster configuration data is 8D:

   ```
   # lsnwtopo -C
   ```

   ISR network topology in use by CNM: 8D:

   ```
   # lsnwtopo -f 6 -c 11
   ```

   Frame 6 Cage 11: Topology 8D, Supernode 5, Drawer 0

Important: CNM must be stopped before the topology updates are made to the cluster DB via the **`chdef`** command and then restarted so that CNM picks up the change. In addition, if the **`chnwsvrconfig`** command is used to update LNMC on an FSP, the cage must be powered off (Standby) before stopping CNM, restarting CNM, and issuing **`chnwsvrconfig`**. The drawer is powered on, IPLed after CNM is restarted, and **`chnwsvrconfig`** is issued.

4.3 HFI

The section describes an overview of some of the commands, tools, and diagnostic tests that are used for identifying and resolving HFI-related problems.

4.3.1 HFI health check

It is a good idea to run an HFI health check to identify potential issues within the HFI infrastructure; for example, after a service window or scheduled power cycle of the cluster.

Some examples of the **`lsnwdownhw`** and **`lsnwlinkinfo`** commands are presented in the section to determine whether more problem troubleshooting and link diagnostic tests are needed relative to the HFI network. A certain level of intuition and knowledge of the specific cluster hardware configuration is needed when the output of the **`lsnwdownhw`** and **`lsnwlinkinfo`** commands is interpreted.

Consider a situation in which a server or an entire frame is powered off for service or maintenance. In this case, **lsnwdownhw** and **lsnwlinkinfo** provides status that is interpreted as multiple network hardware issues when in reality the hardware is powered off such that the links are down.

The following examples illustrate the use of the **lsnwdownhw** command and its common options. With no options, the **lsnwdownhw** command displays the following faulty network hardware:

```
# lsnwdownhw
ISR FR006-CG12-SN0091-DR0-HB1 NOT_NM_READY Service_Location:
U78A9.001.1122233-P1-R3
Link FR006-CG12-SN0091-DR0-HB1-D6 DOWN_FAULTY Service_Location:
U78A9.001.1122233-P1-T13-T5
Link FR006-CG12-SN0091-DR0-HB1-D12 DOWN_FAULTY Service_Location:
U78A9.001.1122233-P1-T16-T5
```

With the **-I** option, the following faulty ISRs are displayed:

```
# lsnwdownhw -I
ISR FR006-CG12-SN0091-DR0-HB1 NOT_NM_READY Service_Location:
U78A9.001.1122233-P1-R3
```

With the **-L** option, the following faulty links are displayed:

```
# lsnwdownhw -L
Link FR006-CG12-SN0091-DR0-HB1-D6 DOWN_FAULTY Service_Location:
U78A9.001.1122233-P1-T13-T5
Link FR006-CG12-SN0091-DR0-HB1-D12 DOWN_FAULTY Service_Location:
U78A9.001.1122233-P1-T16-T5
```

In addition, **lsnwdownhw -H** displays the faulty HFIs and **lsnwdown -a** displays the network hardware which is in any state other than UP_OPERATIONAL. For more information about the **lsnwdownhw** command, see the man pages.

The **lsnwlinkinfo** command is used to display more link information and status. The server is specified as a ‘frame-cage’ combination or as a ‘supernode-drawer’ combination as shown in the following example:

```
# lsnwlinkinfo -f 7 -c 7 -m 7
FR007-CG07-SN004-DR0-HB7-LL5 UP_OPERATIONAL ExpNbr:FR007-CG07-SN004-DR0-HB6-LL5
ActualNbr: FR007-CG07-SN004-DR0-HB6-LL5
FR007-CG07-SN004-DR0-HB7-LR10 DOWN_NBRNOTINSTALLED
ExpNbr:FR00-CG00-SN004-DR2-HB2-LR15 ActualNbr: FR000-CG00-SN000-DR0-HB0-Lxx
FR007-CG07-SN004-DR0-HB7-D14 UP_OPERATIONAL
ExpNbr:FR007-CG04-SN001-DR0-HB7-D11ActualNbr: FR007-CG04-SN001-DR0-HB7-D11
```

The LR link that includes the DOWN_NBRNOTINSTALLED status indicates a problem that must be investigated as a possible hardware or cabling issue, as described in 4.3.2, “HFI tools and link diagnostic tests” on page 260.

4.3.2 HFI tools and link diagnostic tests

This section describes practical examples of identifying and resolving HFI cabling issues. These types of errors often are encountered and corrected during the installation and bring-up phases of the cluster deployment. However, it is not uncommon for fiber cables or other hardware to become defective over time because of a physical mis-cabling or a poorly seated cable as a result of a service action.

Issue the following **lsnwlinkinfo** command to help you identify and resolve a defective or poorly seated fiber cable:

```
# lsnwlinkinfo
FR006-CG06-SN051-DR3-HB3-LR9 DOWN_FAULTY ExpNbr: FR006-CG04-SN051-DR1-HB1-LR11
ActualNbr: FR000-CG00-SN511-DR0-HB0-Lxx
```

Output from lsnwlinkinfo: The output from **lsnwlinkinfo** indicates that the expected L-link connection between FR006-CG06-SN051-DR3-HB3-LR9 and FR006-CG04-SN051-DR1-HB1-LR11 is missing. The generic ActualNbr value of FR000-CG00-SN511-DR0-HB0-Lxx implies that the fiber cable between the specified link ports is missing, improperly seated, or defective.

Complete the following steps to use the **nwlinkdiag** command to identify the cause of the problem:

1. Remove the cable from L-link port FR006-CG06-SN051-DR3-HB3-LR9. If the cable is missing or not properly seated, this condition is likely the cause of the problem.
2. Install a wrap device in L-link port FR006-CG06-SN051-DR3-HB3-LR9.
3. Issue the **nwlinkdiag -s 51 -d 3 -m 3 -lr 9** command.

   The command results show that the FR006-CG06-SN051-DR3-HB3-LR9 LOCATION_CODE: U78A9.001.9920389-P1-T9 link is operational.
4. Remove the wrap device and reinstall the cable at FR006-CG06-SN051-DR3-HB3-LR9.
5. Remove the cable from L-link port FR006-CG04-SN051-DR1-HB1-LR11. If the cable is missing or not properly seated, this condition is likely the cause of the problem.
6. Install a wrap device in L-Link port FR006-CG04-SN051-DR1-HB1-LR11.
7. Issue the **nwlinkdiag -s 51 -d 1 -m 1 -lr 11** command.

   The command results show that the FR006-CG04-SN051-DR1-HB1-D11 LOCATION_CODE: U78A9.001.9920137-P1-T9 link is operational.
8. Remove the wrap device and reinstall the cable at FR006-CG04-SN051-DR1-HB1-D11.
9. Because the wrap test passed on both ports, the problem is most likely a missing, poorly seated, or defective cable.
10. Rerun **lsnwlinkinfo** to confirm whether an issue with the original suspect L-link connection is still indicated. If the suspect link is still identified, replace the L-link cable between FR006-CG06-SN051-DR3-HB3-LR9 and FR006-CG04-SN051-DR1-HB1-LR11. If the suspect link is no longer identified, the problem was a disconnected or poorly seated cable.
11. After the cable issue is resolved, rerun **lsnwmiswire** to ensure that the suspect link is no longer reported as DOWN_FAULTY.

Wrap tests: If one of these wrap tests fails to indicate a hardware issue with the optical L-link port, the **nwlinkdiag** command returns the following message:

```
FR006-CG06-SN051-DR3-HB3-LR9     LOCATION_CODE: U78A9.001.9920389-P1-T9
```

This message indicates that the link is not operational. If a cable is installed in the optical port, remove the cable, install an optical wrap device, and reissue the **nwlinkdiag** command. If a wrap device already is installed, then the problem is behind the optical port.

Use the **lsnwmiswire** command to identifying and resolving a physical mis-cable, as shown in the following example:

```
# lsnwmiswire
FR004-CG09-SN011-DR0-HB5-D5 DOWN_MISWIRED ExpNbr: FR004-CG07-SN010-DR0-HB5-D4
ActualNbr: FR004-CG07-SN010-DR0-HB5-D5
FR004-CG09-SN011-DR0-HB4-D5 DOWN_MISWIRED ExpNbr: FR004-CG07-SN010-DR0-HB4-D4
ActualNbr: FR004-CG07-SN010-DR0-HB4-D5
```

The lsnwmiswire output: The following **lsnwmiswire** output indicates that two D-Links are incorrectly cabled:

- FR004-CG09-SN011-DR0-HB5-D5 is connected to FR004-CG07-SN010-DR0-HB5-D5 instead of FR004-CG07-SN010-DR0-HB5-D4, as expected.
- FR004-CG09-SN011-DR0-HB4-D5 is connected to FR004-CG07-SN010-DR0-HB4-D5 instead of FR004-CG07-SN010-DR0-HB4-D4, as expected.

This instance is typical of two fiber cables that are plugged into the wrong ports at the D-Link hub. The problem is resolved by swapping the following cables at the specified D-link ports:

> The end of the cable plugged into FR004-CG07-SN010-DR0-HB5-D5 is moved to FR004-CG07-SN010-DR0-HB5-D4 and the end of the cable plugged into FR004-CG07-SN010-DR0-HB4-D5 is moved to FR004-CG07-SN010-DR0-HB4-D4.

After the mis-cables are corrected, rerun the **lsnwmiswire** command to check that the links are no longer reported as DOWN_MISWIRED.

4.3.3 SMS ping test fails over HFI

If the SMS ping test over HFI fails, check the following conditions or try the following fixes:

- Is hf0 configured and working properly on the service node?
- Are the HFI device drivers installed on the service node?
- Is `ipforwarding` set to 1 on the service node?
- Does **lsnwloc** show the CEC at RUNTIME?
- Are the FSPs pingable?
- Restart `lnmcd` on the FSP.
- Ensure that CNM and the HFI device driver are up to date on the service node.

4.3.4 netboot over HFI fails

If the SMS ping is successful but **rnetboot** fails, check the following conditions or try the following fixes:

- Did bootp start on the service node?
- Refresh inetd on the service node.
- Ensure that the service node and xcatmaster of the node are set correctly.
- Check **mkdsklsnode** for errors.
- Confirm that the hfi_net nim object on the service node is set to 'hfi'.
- Check that the HFI device drivers are the same as on the compute image.
- Verify the Media Access Control (MAC) address.
- Check **lsnim**, /etc/bootptab, and l**s -l /tftpboot** on the service node.
- Is **/install** set to mount automatically and mounted on service node?
- Is **/install** on a file system other than / on the EMS and service node?
- Are any file systems full on the EMS or service node?
- Do **rpower** and **lshwconn** return the expected values?
- Are the directories in /etc/exports mountable?

4.3.5 Other HFI issues

In rare instances, HFI problems might exist that result in scenarios in which link connections are lost and recovered intermittently, unexplained node or cluster performance issues are occurring, and so on.

When this type of situation is encountered, it is necessary to open a PMR and gather the following data for review by the appropriate IBM teams:

- From the EMS: Run /usr/bin/cnm.snap and provide the snap file of type snap.tar.gz created in /var/opt/isnm/cnm/log/.
- If applicable, from the affected node, run /usr/hfi/bin/hfi.snap and provide the snap file of type hfi.snap.tar created in /var/opt/isnm/cnm/log/.

Depending on the nature of the problem, more data might be required, but this data must be available when IBM support is contacted.

5

Maintenance and serviceability

This chapter describes topics that are related to IBM Power Systems 775 maintenance and serviceability.

This chapter includes the following topics:

- Managing service updates
- Power 775 xCAT startup and shutdown procedures
- Managing cluster nodes
- Power 775 Availability Plus

5.1 Managing service updates

This section provides an overview on updating the firmware and software for the various components within the IBM Power 775 High Performance Computing cluster.

5.1.1 Service packs

Service packs are used to indicate components and levels of components that are verified to operate together. The service pack includes a readme file for installing and updating the IBM HPC clustering with Power 775. The service pack also contains the following useful service information:

- Software offerings
- Recommended code levels
- Recommended installation sequence
- Known restrictions and limitations
- Known problems and workarounds
- Hints and tips
- Service pack support matrix and archive

For the latest information about applying code maintenance, see the appropriate service pack on the IBM HPC clustering with Power 775 servers/service packs portion of the IBM High Performance Computing clusters service packs at this website:

http://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+High+Performance+Computing+Clusters+Service+Packs#IBMHighPerformanceComputingClustersServicePacks-IBMHPCClusteringwithPower775serversServicePacks

If your service pack is not listed, see the most recent service pack and access the archived service packs that lists your service pack.

5.1.2 System firmware

This section describes the basic steps for updating and validating firmware updates to the frame power code and the CECs GFW system firmware.

For more information about updating the Central Electronic Complex (CEC) firmware, and validating CECs power up, see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_System_p7_775_Hardware_Management#Update_the_CEC_firmware.2C_and_Validate_CECs_Can_Power_Up

IBM fix central

Locate and download the supported Power 775 power code and GFW System firmware (CEC) from IBM fix central to a directory on the EMS. The IBM HPC clustering with Power 775 service pack contains the recommended code levels. Links to fix central and to download the firmware are available at this website:

http://www.ibm.com/support/fixcentral

Direct FSP/BPA management

The current **rflash** implementation of direct Flexible Service Processor/Bulk Power Assembly (FSP/BPA) management does not support the **concurrent** value for the **--activate** flag, and supports only the **disruptive** option. The **disruptive** option causes any affected systems that are powered on to power down. This update requires that the systems are powered off before the firmware is updated.

A **-d** (data-directory) option exists in direct FSP/BPA management. The default directory is /tmp. When performing the firmware update, the **rflash** command imports some related data from the Rational Portfolio Manager (RPM) packages into the (data-directory) directory, so the execution of the **rflash** command requires the available disk space in the data-directory. For more information about the available disk space requirements, see the man page for the **rflash** command.

Applying system firmware updates

To apply system firmware updates, use the xCAT **rinv** command to determine the current level and the **rflash** command to update the code level, then validate that the update is successful. Complete the following command sequence to apply the firmware updates:

1. Check the current firmware level by using the **rinv cec firm** command.
2. Flash the new firmware level by using the **rflash cec -p <directory> --activate disruptive** command.
3. Check for the new firmware level by using the **rinv cec firm**command.
4. Check the health of the CEC by using the following command:

   rpower cec state

   rvitals cec lcds

Applying power code updates

To apply system firmware updates, use the xCAT **rinv** command to get the current level and the **rflash** command to update the code level. Complete the following command sequence to apply the firmware updates:

1. Check the current firmware level by using the **rinv frame firm** command.
2. Flash the new firmware level by using the **rflash frame -p <directory> --activate disruptive** command.
3. Check for the new firmware level by using the **rinv frame firm** command.
4. Check the health of the CEC by using the following command:

   rpower frame state

   rvitals frame lcds

Important: **-p** indicates the directory in which the firmware code is available in the EMS.

5.1.3 Managing multiple operating system images

xCAT supports the creation and installation of diskfull or diskless images on the nodes. The operating system (OS) images are built and retained on the management node. Each node runs its own osimage definition that is tailored to the functionality they perform. The nodes also include their own specific security hardening needs. xCAT also ensures that the node uses a minimal osimage definition without any unnecessary file sets and applications. The ability to rapidly reinstall the original image to the node helps protect against malicious software.

Types of nodes for diskfull and diskless

This section describes the types of nodes that are used for diskfull and diskless.

Diskfull node

For AIX systems, this node has local disk storage that is used for the OS (a stand-alone node). Diskfull AIX nodes often are installed by using the NIM *rte* or *mksysb* installation methods.

Diskless node

The OS is not stored on local disk. For AIX systems, this configuration means that the file systems are mounted from a NIM server. An AIX diskless image is essentially a SPOT. This image provides a `/usr` file system for diskless nodes and a root directory the contents of which are used for the initial diskless nodes root directory. The image also provides network boot support.

You use diskless nodes as *stateful* or *stateless*. If you want a stateful node, you must use an NIM root resource. If you want a stateless node, you must use an NIM shared_root resource.

Stateful node

A stateful node is a node that maintains its state after it is shut down and rebooted. The node state is any node-specific information that is configured on the node. For AIX diskless nodes, this state means that each node has its own NIM root resource that is used to store node-specific information. Each node mounts its own root directory and preserves its state in individually mounted root file systems. When the node is shut down and rebooted, any information that is written to a root file system is available

Stateless node

A stateless node is a node that does not maintain its state after it is shut down and rebooted. For AIX diskless nodes, this state means that all of the nodes use the same NIM shared_root resource. Each node mounts the same root directory. Anything that is written to the local root directory is redirected to memory and is lost when the node is shut down. Node-specific information must be re-established when the node is booted.

The advantage of stateless nodes is that there is much less network traffic and fewer resources used. which is especially important in a large cluster environment.

AIX

This section describes basic steps for updating and validating a diskfull node and a diskless node.

For more information about updating software on AIX stand-alone (diskfull) nodes, and updating software for AIX diskless nodes, see this website:

> http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Updating_AIX_Software_on_xCAT_Nodes

Updating diskfull

The xCAT **updatenode** command is used to perform software maintenance operations on AIX/NIM stand-alone machines. This command uses underlying AIX commands to perform the remote customization of AIX diskfull (stand-alone) nodes. The command supports the AIX **installp**, **rpm**, and **emgr** software packaging formats.

As part of this approach, the recommended process is to copy the software packages or updates that you want to install on the nodes into the appropriate directory locations in the NIM lpp_source resource that you use for the nodes.

> **Important:** If you want to use another directory to store your software, you use the **updatenode** command with the **-d <dirname>** to specify an alternate source location. You must ensure that the alternate directory is mountable and that the files are readable. For more information, see the **updatenode** man page.

A simple way to copy software to the lpp_source locations is to use the **nim -o update** command. With this NIM command, you must provide the name of the NIM lpp_source resource and your files are automatically copied to the correct locations.

For example, to add all the packages to the lpp_source resource named “61image_lpp_source” you run the following command:

```
nim -o update -a packages=all -a source=/tmp/images 610image_lpp_source
```

The NIM command finds the correct directories and update the lpp_source resource.

Assume all your software is saved in the temporary location /tmp/images.

When you run the **updatenode** command, the default behavior is to get the name of the osimage definition that is defined for each node and use those names to determine the location of the software and the names of the software to install. The command uses the location of the lpp_resource to determine where to find any defined rpm, installp, or emgr packages.

The default value for installp_flags is **-agQX** and the default value for rpm_flags is **-Uvh -replacepkgs**. No flags are used by default in the call to emgr.

> **Hierarchical xCAT cluster:** When you are working in a hierarchical xCAT cluster, the **updatenode** command automatically distributes the software to the appropriate service nodes.

Examples of lpp, rpm, and efix

You also might specify alternative installp, emgr, and rpm flags for **updatenode** to use when the underlying AIX commands are called. Use the **installp_flags**, **emgr_flags**, and **rpm_flags** attributes to provide this information. You must specify in quotes the exact string that you want used. For example: **installp_flags="-apXY" rpm_flags="-i -nodeps"**.

Complete the following steps to install lpp filesets into diskfull node:

1. Update the AIX node “xcatn11” by installing the bos.cpr fileset by using the “-agQXY” installp flags. Also, display the output of the **installp** command:

   ```
   updatenode xcatn11 -V -S otherpkgs="bos.cpr" installp_flags="-agQXY"
   ```

2. Uninstall the “bos.cpr” fileset that was installed in the previous step:

   ```
   updatenode xcatn11 -V -S otherpkgs="bos.cpr" installp_flags="-u"
   ```

Complete the following steps to install rpm packages into diskfull node:

1. Update the AIX nodes “xcatn11” with the “rsync” rpm by using the rpm flags **"-i --nodeps"**:

   ```
   updatenode xcatn11 -V -S otherpkgs="R:rsync-2.6.2-1.aix5.1.ppc.rpm"
   rpm_flags="-i --nodeps"
   ```

2. Uninstall the rsync rpm that was installed in the previous step:

   ```
   updatenode xcatn11 -V -S otherpkgs="R:rsync-2.6.2-1" rpm_flags="-e"
   ```

Complete the following steps to install efix packages into diskfull node:

1. Install the interim fix package in the `/efixes` directory:

   ```
   updatenode node29 -V -S -d /efixes otherpkgs=IZ38930TL0.120304.epkg.Z
   ```

2. Uninstall the interim fix that was installed in the previous step:

   ```
   updatenode xcatsn11 -V -S -c emgr_flags="-r -L IZ38930TL0"
   ```

Updating diskless nodes

To update an AIX diskless node with new or more software, you must modify the NIM SPOT resource (OS image) that the node uses and then reboot the node with the new SPOT. You cannot directly install software on a running diskless node.

This section describes how AIX diskless nodes are updated by using xCAT and AIX/NIM commands. The section describes the process to switch the node to a different image or update the current image. The following information is not meant to be an exhaustive presentation of all options that are available to xCAT/AIX system administrators.

> **Important:** Because you cannot modify a SPOT while it is used by a node, there are two options to modify a SPOT: stop all of the nodes and then update the existing OS image, or create an updated image to use to boot the nodes.

Copy an existing image

You use the **mknimimage** command to create a copy of an image. For example, if the name of the currently running image is 61dskls and you want to make a copy of the image to update, you run the following command:

```
mknimimage -t diskless -i 61dskls 61dskls_updt
```

The new image is updated and used to boot the node.

Update the image (SPOT)

Several types of updates are performed on a SPOT:

- Add or update software.
- Update the system configuration files.
- Run commands in the SPOT by using the **xcatchroot** command.

You use the xCAT **mknimimage -u** command to install installp filesets, rpm packages, and epkg (the interim fix packages) in a SPOT resource.

Before the **mknimimage** command is run, you must add the new filesets, RPMs, or epkg files to the lpp_source resource that was used to create the SPOT. If we assume that the lpp_source location for 61dskls is `/install/nim/lpp_source/61dskls_lpp_source`**,** the files are in the following directories:

- installppackages: `/install/nim/lpp_source/61dskls_lpp_source/installp/ppc`
- RPM packages: `/install/nim/lpp_source/61dskls_lpp_source/RPMS/ppc`
- epkg files: `/install/nim/lpp_source/61dskls_lpp_source/emgr/ppc`

The easiest way to copy the software to the correct locations is to use the **nim -o update** command. To use this command, you must provide the directory that contains your software and the NIM lpp_source resource name (for example, **"**`61dskls_lpp_source`**"**).

If your new packages are in `/tmp/myimages` directory, you run the following command:

```
nim -o update -a packages=all -a source=/tmp/myimages 61dskls_lpp_source
```

> **Important:** If you do not use this command to update the lpp_source, you must update the `.toc` file by running the **inutoc** command. After the `lpp_source` is updated, you use the **mknimimage** command to install the updates in the SPOT resource for this xCAT osimage.

Examples of lpp, rpm, and efix

You add the information to the **mknimimage** command line. If you provide one or more of the "installp_bundle", "otherpkgs", or "synclists" values on the command line, the **mknimimage** command uses only these values. The xCAT osimage definition is not used or updated when you install software into a SPOT by using the following command:

```
   mknimimage -u my61dskls installp_bundle="mybndlres1,mybndlres2"
   otherpkgs="openssh.base,R:popt-1.7-2.aix5.1.ppc.rpm,IZ38930TL0.120304.epkg.Z"
```

The `installp_bundle` value is a comma-separated list of (previously defined) NIM `installp_bundle` resource names. The `otherpkgs` value is a comma-separated list of installp filesets, rpm package names, or epkg file names. The rpm names must be preceded by "R:" (for example, `R:foo.rpm`).

You specify rpm flags on the **mknimimage** command line by setting the `rpm_flags` attribute to the value you want to use, as shown in Table . If the default flags are not specified, the flags are "**-Uvh -replacepkgs**". You also specify emgr flags on the **mknimimage** command line by setting the "emgr_flags" attribute to the value you want to use. There are no default flags for the **emgr** command.

Install software with the lpp_flag and rpm_flag by using the following command:

```
mknimimage -u my61dskls installp_flags="-agcQX" rpm_flags="-i --nodeps"
```

5.2 Power 775 xCAT startup and shutdown procedures

This section describes the steps that are used to start up and shut down the Power 775 cluster.

5.2.1 Startup procedures

This section describes the steps that are used to start xCAT 2.66 and the Power 775 hardware and software. The verification steps that are used as the system is started also are described.

The commands that are shown in this section are for an AIX environment. All examples assume that the administrator has root authority on the EMS. This section is intended only as a post-installation startup procedure. Initial installation and configuration are not addressed in this publication.

For more information about the Power 775 related software, see these websites:

- https://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+Overview
- https://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+-+Cluster+Guide

Terminology

The following terms are used in this document:

- *xCAT Direct FSP Management*

  Direct FSP Management (DFM) is the name that is used to describe the ability of the xCAT software to communicate directly to the service processor of the Power system without the need for the HMC management.

- *Frame node*

  This node features the hwtype set to frame, which represents a high-end Power system server 24-inch frame.

- *CEC node*

  This node features the attribute hwtype set to cec, which represents a Power system CEC (for example, one physical server).

- *BPA node*

  This node features a hwtype set to BPA and represents one port on one bpa (each BPA has two ports). For the purposes of xCAT, the BPA is the service processor that controls the frame. The relationship between the frame node and the BPA node from the system administrator's perspective is that the administrator must always use the frame node definition for the xCAT hardware control commands. xCAT figures out which BPA nodes and their respective IP addresses to use for hardware service processor connections.

- *FSP node*

  This node features the hwtype set to FSP and represents one port on the FSP. In one CEC with redundant FSPs, there are two FSPs and each FSP has two ports. There are four FSP nodes that are defined by xCAT per server with redundant FSPs. Similar to the relationship between Frame node and BPA node, system administrators always use the CEC node for the hardware control commands. xCAT automatically uses the four FSP node definitions and their attributes for hardware connections.

- *Service node (SN)*

  This node s an LPAR which helps the hierarchical management of xCAT by extending the capabilities of the EMS. The SN have a full disk image and is used to serve the diskless OS images for the nodes that it manages.

- *IO node*

  This node is an LPAR which includes attached disk storage and provides access to the disk for applications. In 775 clusters the IO node is running GPFS and is managing the attached storage as part of the GPFS storage.

- *Compute node*

  This node is used for customer applications. Compute nodes in a 775 cluster have no local disks or Ethernet adapters. They are diskless nodes.

- *Utility node*

  This node is a general term which refers to a non-compute node/LPAR and a non-IO node/LPAR. Examples of LPARs in a utility node are the service node, login node, and local customer nodes for backup of data, or other site-specific functions.

- *Login node*

  This node is an LPAR defined to allow the users to log in and submit the jobs in the cluster. The login node most likely has an Ethernet adapter that connects it to the customer VLAN for access.

Power 775 architecture interrelationships and dependencies

In a Power 775 cluster, interrelationships and dependencies in the hardware and software architecture require that the startup is performed in a specific order. In this publication, we explain these relationships and dependencies and describe the process to properly bring the system up to a running state in which users log in and submit jobs.

Hardware roles

Each hardware set features a designated role in the cluster. This section describes each part of the hardware and its role.

Ethernet network switch

The Ethernet switch hardware is key to any computer complex and provides the networking layer for IP communication.

In a 775 cluster, the switch hardware is used to support the cluster management LAN, which is used by xCAT for OS distribution from the EMS to SN and administration from the EMS to the SN. This hardware also is used to support the cluster service LAN, which connects the EMSs, SNs, HMCs, FSPs, and BPAs to provide access to the service processors within each frame.

To understand the flow of the startup process, we distinguish the different hardware responsibilities in the order in which each set of hardware becomes involved in the start process.

Executive management server

The xCAT executive management server (EMS) is the central point of control for administration of the cluster. The EMS contains the xCAT DB, the DB of the central network manager, and TEAL and its DB.

Hardware Management Consoles

The hardware management consoles (HMCs) are used for Service Focal Point and repair and verification procedures. During the initial installation and configuration, the HMCs are assigned the frames and CECs that they monitor for any hardware failures.

Service node

The service nodes are LPARs within a building block, which consists of a full disk image and serves the diskless OS images for the nodes that it manages. All diskless nodes require that the SN supporting them is running before successfully booting. Some administrative operations in xCAT issued on the EMS are pushed out to the SN to perform the operations in a hierarchical manner, which is needed for system administration performance.

I/O node

The I/O node is the LPAR with attached storage. The node contains the GPFS software that manages the global file system for the cluster. The I/O nodes must be operational before compute nodes mount the global file system.

Startup assumptions

Some areas are outside of the scope of this process. To draw a boundary on what hardware is part of the start process and what is considered a prerequisite, some assumptions are made. For example, it is assumed that the site has power and that everything is in place to begin the start process, including the site cooling is up and operational and all power to the devices (switch, EMS, HMC, frames) is ready to be applied.

The network switch hardware is a gray area in this process as some network switch hardware is part of the HPC cluster and others might be outside the cluster. For this discussion, we make the assumption that all network switches that are customer site-specific and not HPC cluster-specific are up and operational.

There are some manual tasks that are involved in this process which require a manual start of the equipment. There must be people available to perform these tasks and they must be familiar with the power-on controls that are needed for each task they are to perform. Examples of these controls include powering on the Ethernet network switches, shared disk for dual EMS, HMC, frames, and so on. These manual tasks must be performed when it is time to perform the step.

This process also assumes that all initial cabling and configuration (hardware and software) is completed before this process is started. It is also assumed that the entire system is booted and testes were conducted to eliminate any hardware or software problems before performing this procedure.

Dependencies

As the cluster is started, it is critical that hardware or software dependencies are up and operational before the successful completion of a hardware or software item that features the dependency. A high-level view of the dependencies helps to outline the flow of the startup process. This section provides an overview of the following dependencies (details about accomplishing the task or verifying its completion are not provided):

- Ethernet switches

  At the top of the dependencies is the HPC cluster Ethernet switch hardware and any customer Ethernet switch hardware. These items are the first items that must be started.

- EMS and HMCs

  The next level of dependency is the EMS and HMCs. These items are started at the same time after the network switches are started.

- Frames

  After the EMS and HMCs are started, we begin to start the 775 hardware by powering on all of the frames. The frames are dependent on the switches and the EMS to come up properly.

- CECs

  After the frame is powered on, the CECs are powered on. The CECs depend on the switches, EMS, and frames. Applying power to the CECs brings up the HFI network hardware, which is critical to distributing the operating system to diskless nodes, as well as for application communication.

- Service Node

  The SN is started after the CECs are powered on. The SN is dependent on the switches, EMS, frame, and CEC.

- I/O node

  The I/O node is started after the SN is operational. The I/O node is dependent on the switches, EMS, frame, CEC, and SN.

- Compute nodes

  Last in the list is the starting of the compute nodes. The compute nodes are done after the SN and I/O nodes are up and operational. The login and compute node require the SN to be operational for the OS images loading. Compute nodes depend on Ethernet switches, EMS, frame, CEC, SN, and I/O nodes.

After the compute nodes start, the hardware startup process ends and the administrator begins to evaluate the HPC cluster state by checking the various components of the cluster. After the HPC stack is verified, the cluster startup is complete.

Other node types that each customer define to meet their own specific needs might be needed. Some examples are nodes responsible for areas such as login and data backup. These nodes must be brought up last to allow the rest of the cluster to be up and running. Because these nodes are outside the HPC cluster support and software and the nature of their startup is an unknown factor, the nodes are outside the scope of this publication and are not part of any timing of the cluster startup process.

Startup procedure

This section describes the startup procedure. The following sections describe the prerequisites, the process for this step, and the verification for completion. Some assumptions are made about the current site state, which must be met before starting this process. These assumptions include cooling and power, and initial configuration and verification of the cluster that is performed during installation.

Before we begin with the startup procedure, we describe the benefit of the use of xCAT group names. xCAT supports the use of group names that allow the grouping of devices or nodes in a logical fashion to support a type of nodes. We recommend that the following node groups be in place before performing this procedure: frame, cec, bpa, fsp, service, storage, and compute. Other node groups might be used to serve site-specific purposes.

Creating node groups significantly enhances the ability to start a group of nodes at the same time. Without these definitions, an administrator must issue many separate commands instead of a single command.

It is also key to manage any node failures in the startup process and continue when possible. An issue with some part of the cluster that starts might exist that does not affect other parts of the cluster. When this condition occurs, you must continue with the boot process for all areas that are successful and retry or diagnose the section with the failure. Continuing with the boot process allows the rest of the cluster to continue to start, which is more efficient than holding up the start of the entire cluster.

Optional power-on hardware

During the cluster shutdown process, there is an optional task for disconnecting power. If you turned off breakers or disconnected power to the management rack or the 775 frames during the cluster shutdown process, you must continue with this step to connect power.

Perform the following steps only if any of the breakers are shut off or power is disconnected:

1. Turn on breakers or connect power to the management rack.
2. Turn on breakers or connect power to the 775 frames.

Powering on external disks attached to EMS

Power on any external disks that are used for dual-EMS support. This power-on is required before starting the primary EMS.

Power on EMS and HMCs

After the EMS shared disk drives are up, it is time to power on the primary EMS and the HMCs.

The backup EMS is started after the cold start is complete and the cluster is operational. The backup EMS is not needed for the cluster start, and spending time to start it takes away from the limited time for the entire cluster start process.

Starting the primary EMS and the HMCs is a manual step that requires the administrator to push the power button on each of these systems to start the boot process. The EMS and HMCs are started at the same time because they do not have a dependency on each other.

Primary EMS start process

The administrator must execute multiple tasks that work with the primary xCAT EMS. Confirm that all of the local and external attached disks are started and available to the xCAT EMS.

> **Important:** Do not start the backup EMS or perform any steps on a backup EMS now. The backup EMS must be started after the Power 775 cluster start process is complete and working with the primary xCAT EMS.

The administrator must ensure that all of the files systems are mounted properly, including file systems on external shared disks. The expectation is that some directory names might vary depending on the site, as shown in Example 5-1.

Example 5-1 Checking the mounted file systems

```
$ mount ? /etc/xcat
$ mount ? /install
$ mount ?~/.xcat
$ mount ? /databaseloc</pre>
```

The administrator must ensure that the DB2 environment is enabled on the xCAT EMS. This verification includes validating that the DB2 monitoring daemon is running, and that the xCAT DB instance is set up, as shown in Example 5-2 on page 277.

Example 5-2 Starting the DB2 daemon

```
$ /opt/ibm/db2/V9.7/bin/db2fmcd &
```

Example 5-3 shows the DB2 commands to start the xcatdb instance.

Example 5-3 To start the xcatdb instance

```
$ su - xcatdb
$ db2start xcatdb
$ exit
```

The administrator checks that multiple daemons (including xcatd, dhcpd, hdwr_svr, cnmd, teal) are properly started on the xCAT EMS. For the xCAT Linux EMS, execute the Linux "**service**" command with the start attribute to start each of the daemons, as shown in Example 5-4.

Example 5-4 Starting each of the daemons

```
$ ? (start xCAT deamon)
$ ? (start dhcpd daemon)
$ ? (start hdwr_svr daemon)
$ ? (start teal daemon)
$ ? (start cnmd daemon)
```

The administrator uses the **ps -ef | grep xxx** command to validate that the daemons are running after they are started. The administrator also verifies that the daemons are running by using the Linux service command that works with status attribute, as shown in Example 5-5.

Example 5-5 Verifying the daemons are running

```
$ lsxcatd -a (verify xCAT deamon is running and can access the xCAT database)
$ ? (verify dhcpd is running)
$ ?(verify conserver is running)
$ ?(verify hardware server is running)
$ ? (verify cnmd is running)
$ ?(verify teal.py pid is running)
```

HMC verification

By using the xCAT EMS, verify that each of the HMCs is up and running.

If SSH is configured, verify that the HMC is running, and that there is connectivity by checking that SSH is configured on all HMCs, as shown in Example 5-6.

Example 5-6 Verifying the HMC is running and ssh is configured on all HMCs

```
$ rspconfig hmc sshconfig
```

If SSH is not enabled on the HMCs, validate that the HMCs are powered up. You might issue an **ssh** command to each HMCs to see whether SSH is available. You execute **rspconfig <hmc node> sshconfig=enable** to configure the SSH environment.

Power on frames

Powering on the frame is a manual process that requires the administrator to turn on the red Emergency Power Off (EPO) switch in front of the frame. This task applies power only to the bulk power unit of the frame. The frame BPAs take approximately 3 minutes to boot after power is applied. The BPAs stop at a rack standby state.

The administrator executes the hardware discovery command **`lsslp -m -s FRAME`** to keep track of all of the frame BPA IPs. The administrator executes the command **`lshwconn frame`** to ensure that there are hardware connections between the xCAT EMS and the frame BPAs. The administrator executes the **`rpower frame state`** command to ensure that the frame status is set as both BPAs at rackstandby.

Issue the commands that are shown in Example 5-7 to verify that the frame BPAs are properly set in the rack standby state.

Example 5-7 Verifying the frame BPAs are set in the rack

```
$ lsslp -m -s FRAME
$ lshwconn frame
$ rpower frame state
```

Power on the CEC FSPs

To apply power to all the CEC FSPs for each frame, we need to exit rack standby mode by issuing the command **`rpower frame exit_rackstandby`**. It takes approximately 5 minutes for the frame BPAs to be placed in the standby state. The administrator executes the **`rpower frame state`** to make sure the frame status is set as Both BPAs at standby. The administrator checks the frame environmental status by using the **`rvitals frame all`** command.

Important: The BPAs and FSPs can take 10 - 15 minutes to complete the starting process.

Issue the commands shown in Example 5-8 to verify that the frame BPAs are properly set in the standby state.

Example 5-8 Verifying the frame BPAs are in standby state

```
$ rpower frame exit_rackstandby
$ rpower frame state
$ rvitals frame all
```

As part of the exit rack standby, each frame BPA applies power to all the FSPs in its frame, which causes the CEC FSPs to initial program load (IPL). The expectation is that it might take up to 10 minutes for all of the CEC FSPs to enable in the frame after exiting rack standby. The administrator executes the hardware discovery command **`lsslp -m -s CEC`** to track of all the CEC FSP IPs, as shown in Example 5-9. The administrator executes the command **`lshwconn frame`** to ensure that there are hardware connections between the xCAT EMS and the CEC FSPs. The administrator executes the **`rpower cec state`** to ensure that the CECs are placed in a power off state.

Example 5-9 Checking the status of all the CEC FSP IPs

```
$ lsslp -m -s CEC
$ lshwconn cec
$ rpower cec state
```

After the IPL of the CEC FSPs is complete, they are in a power off state. The administrator also needs to validate that CNM includes proper access to the CEC FSPs and frame BPAs from the xCAT EMS. Verify that there are proper hardware server connections by issuing the **lshwconn** command by using the fnm tool type, as shown in Example 5-10. Confirm that every frame BPA and CEC FSP are listed in the command output.

Example 5-10 Verifying the hardware server connections

```
$ lshwconn frame -T fnm
$ lshwconn cec –T fnm
```

Another CNM task is to ensure that the HFI master ISR identifier is properly loaded on the CEC FSPs. This task is accomplished by using the CNM commands **lsnwloc** and **lsnwcomponents** to show the CNM HFI drawer status information. If required, the administrator might need to execute the **chnwsvrconfig** command to load in the HFI master ISR identifier into the CEC FSPs, as shown in Example 5-11.

Example 5-11 Showing the CNM HFI drawer status information

```
$ lsnwloc
$ lsnwcomponents
$ chnwsvrconfig -f 17 -c 3 (Configures one cec in P775 frame 17 with cage id 3)
```

CEC power on to standby

After the frames BPAs and the CECs FSPs are booted, we bring the CEC to on standby state. This state powers on the CECs, but does not autostart the power to the LPARs. This configuration is required because we need to coordinate the powering on of selected xCAT service nodes, and GPFS I/O server nodes before powering on the compute nodes. To power on the CECs to on standby state, issue the command that is shown in Example 5-12.

Example 5-12 Powering on the CECs to standby state

```
$ rpower cec onstandby
```

CEC power on monitoring and verification

There are multiple tasks that are monitored by the administrator. These tasks include checking the CEC LCDs by using the **rvitals** command, and tracking the CEC status by using the **rpower** command. It is a good time for the administrator to validate the CNM ISR network environment with the CNM commands **lsnwloc** and **lsnwcomponents** during the powering on of CECs.

> **CEC power-on:** The CEC power-on is one of the largest time-consuming steps in the start process because each CEC is performing all of the hardware verification of the server during this process. The current timings that we observed take approximately 45 - 70 minutes for the CEC to become available.

The CEC IPL process is monitored by using the xCAT **rvitals** and **rpower** commands, as shown in Example 5-13.

Example 5-13 CEC IPL process monitoring

```
$ rvitals cec lcds
$ rpower cec state
```

When the CECs are powering on, verify that the CNM manage the ISR network with the CNM commands shown in Example 5-14.

Example 5-14 Verifying the CNM can manage the ISR network

```
$ lsnwcomponents (provides CEC drawer configuration information)
$ lsnwloc | grep -v EXCLUDED | wc -l (match the number of CEC drawers in the
cluster)
$ lsnwloc | grep EXCLUDED (issues that cause a CEC drawer to be excluded by CNM)
```

If any CEC drawers are excluded (STANDBY_CNM_EXCLUDED or RUNTIME_CNM_EXCLUDED), see *High performance clustering using the 9125-F2C Management Guide* at this website:

https://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+-+Cluster+Guide

More information about CNM commands, implementation, and debug before powering the CECs on to standby is available in this management guide.

Powering on service nodes

At this stage of the process, power is on the frame and all of the CECs in which we are ready to boot the xCAT SNs. The xCAT SNs are the first nodes to boot within the Power 775s because they supply the OS diskless images for the remaining nodes. The administrator executes the **rpower** command by using the service node group to power on all of the xCAT service nodes, as shown in Example 5-15.

Example 5-15 Powering on all of the xCAT service nodes

```
$ rpower service on
```

The verification process for the service node includes validating the operating system boot, critical daemons and services that started and the proper communication to the xCAT EMS and other xCAT SNs. Example 5-16 shows the xCAT commands that are issued from the EMS to all of the service nodes at the same time.

> **Important:** This process assumes that the service node already is properly configured to not start GPFS and LoadLeveler. GPFS is not available until the GPFS I/O storage nodes are booted after this step and the LoadLeveler requires GPFS.

Example 5-16 xCAT commands

```
$ rpower service state (verify the Service Node state indicates Success)
$ nodestat service (verify network communication to service nodes)
$ xdsh ? (verify that xCAT daemon, xcatd, is running on the service nodes)
$ xdsh ? <service node1 hf0 IP> -c 5</pre> (verify HFI connectivity between the
                                            service nodes)
```

It is important that the xCAT SN includes the proper diskless installation environment to install the GPFS I/O storage nodes. We also must validate that the diskless images are set for the login and compute nodes by using the nodeset command, as shown in Example 5-17 on page 281.

Example 5-17 Run nodeset on all diskless groups to prepare then for booting with XCAT SNs

```
$ nodeset storage osimage=? (setup install/boot for storage nodes)
$ nodeset compute osimage=? (setup install/boot for compute nodes)
$ nodeset login osimage=? (setup install/boot for login nodes)
```

Important: The osimage definition name must be the image name that you are using for that node type.

The disk enclosures received power when the frame in which they are enclosed exited rack_standby mode. This configuration powers up the disk enclosures so that they are operational when the GPFS storage nodes to which they are attached are started.

The frames, CECs, and xCAT SN are powered up and active. We validated that the xCAT SN properly enabled the CNM HFI network, and that diskless installation environment is set up. The administrator boots the GPFS I/O storage nodes, and begins to configure GPFS on each of the storage nodes by using the following **rpower** command that works with the storage node group:

```
$ rpower storage on
```

To verify that the storage node OS booted up properly, the administrator checks if the required services are active. This task includes checking the status of the service nodes by using the following **rpower** command:

$ rpower storage state (verify that the storage node state is successful)

Important: Because of the significant number of disk drives, there is a delay in the total time it takes for a storage node operating system to complete the boot process.

The administrator needs to reference the GPFS documentation to properly validate that the disks are properly configured. For more information, see this website:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks.html

After the GPFS storage nodes operating system completes the boot and the disks are configured, we start GPFS on each storage node. The administrator executes the following GPFS commands from the xCAT EMS:

$ xdsh storage -v mmstartup (startup GPFS environment on storage nodes)

$ xdsh storage -v mmgetstate (verify that GPFS is running and operational)

Optional - Check the status of the disks

The following commands help check the status of the disks:

$ xdsh storage -v mmlsdisk all --not -ok (list all disks on storage nodes which are not “ok”)

$ xdsh storage -v mmlspdisk all --not -ok (list all physical disks which are not “ok” on storage nodes)

Many sites include more utility node types such as login nodes or nodes responsible for backup. After the xCAT service nodes are set up in the diskless environment and the GPFS I/O storage nodes are configured, the administrator powers up any site-specific utility nodes. The administrator executes the **rpower** command working with the following login node group:

$ rpower login on

$ rpower login stat (verify that the login node state is successful)

The administrator might want to execute other commands to the utility login nodes to ensure that the application environment is set up.

All critical parts of the Power 775 cluster infrastructure are operational. All of the frames, CECs, and disk enclosures also are powered up and running.

5.2.2 Shutdown procedures

This section describes shutting down the xCAT HPC system Power 775 hardware and software with verification steps as the system is shut down.

See 5.2.1, “Startup procedures” on page 272 to review the overview and dependencies sections because the sections describe the general hardware roles and interdependencies that affect the start and shut down of the cluster.

The examples in this publication are for an AIX environment. All commands assume root access on the EMS. Everything that is described in this publication is supported only in xCAT 2.6.6 and greater. Furthermore, this procedure is intended only as a post-installation procedure.

For more information about the Power 775 related software, see these websites:

- https://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+Overview
- https://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+-+Cluster+Guide

Terminology

The terminology for this section is shared with the terminology section in 5.2.1, “Startup procedures” on page 272.

Cluster shutdown assumptions

This section describes the assumptions that are made about the state of the cluster before the cluster is shut down.

Shutting down an HPC cluster is a task that requires planning and preparation. Care must be taken to inform users that this cluster shutdown operation is occurring. Removing user access to the cluster and stopping the jobs in the LoadLeveler queue are critical first steps in shutting down the cluster.

Timing the shutdown process: When timing the shutdown process for any shutdown benchmarking, the process to drain running jobs must not be considered part of the IBM HPC cluster shutdown process. This shutdown is excluded because it is dependent on how long a user job runs to completion and is not considered part of the actual shutdown. After all of the jobs are stopped, the official timing of the IBM HPC cluster shutdown process begins. For a complete site shutdown process, the time it takes to drain the jobs might be included, but the time varies depending on where each job is in its execution.

Cluster shutdown process

The following sections describe the steps to shut down the cluster. Each step outlines the necessary process and verifications that must be completed before moving to the next step.

The cluster shutdown process is faster than the startup process. During the startup process, it is necessary to manually start some daemons and the hardware verification during startup is a longer process than shutting down the system.

User access

Care must be taken to ensure that any users of the cluster are logged off and that their access is stopped during this process.

Site utility functions

Any site-specific utility nodes that are used to load, backup, or restore user data or other data that is related to the cluster are disabled or stopped.

Preparing to stop LoadLeveler

To shut down the cluster, all user jobs must be drained or canceled. It is assumed that the administrator has a thorough understanding of job management and scheduling to drain or cancel all jobs. There are two methods to accomplish these tasks: draining the jobs and cancelling the jobs.

Environmental site conditions affect whether to drain or cancel the jobs. If the shutdown is a scheduled shutdown with sufficient time for the jobs to complete, then draining the jobs is the best practice. A shutdown that does not allow for all of the jobs to complete requires that the jobs must be canceled.

Shutdown scheduling and preparation for this task in advance is needed, especially when jobs are drained to allow sufficient time for the jobs to complete.

Method 1: Draining LoadLeveler jobs

Draining the jobs is the preferred method but is only attainable when there is time to allow the jobs to complete before the cluster must be shutdown.

Example 5-18 shows how to drain the jobs on compute and service nodes.

Example 5-18 Draining the jobs on compute nodes and schedule jobs on service nodes

```
for n in `nodels compute`
do
   llctl -h $n drain startd
done

xdsh service -v llctl drain
```

To monitor the status of the jobs, issue the **llstatus** command to one of the service nodes, as shown in Example 5-19.

Example 5-19 Monitoring the status of the jobs

```
xdsh f01sv01-v llstatus
```

> **Important:** Because the draining process allows the jobs to complete, this step continues as long as it takes for the longest running job to complete. Familiarity with the jobs and the amount of time the job run helps determine the length of this task.

Method 2 - Stopping LoadLeveler jobs

Example 5-20 shows how to keep any new jobs from starting in LoadLeveler.

Example 5-20 Draining LoadLeveler jobs on the compute nodes and service nodes

```
for n in `nodels compute`
do
   llctl -h $n drain startd
done
xdsh service -v llrctl drain
```

Wait for running jobs to complete, or alternately, if the jobs are terminated and restarted, flush the jobs on the compute nodes by entering the commands that are shown in Example 5-21.

Example 5-21 Flushing the jobs in the compute node

```
xdsh compute -v llrctl flush
```

> **Important:** In the job command file, the **restart=yes** flag must be specified. If the flag is not specified, it is similar to **llcancel**. The jobs that are running on a node are gone permanently after you flush the node.

To monitor the status of the jobs, issue the **llstatus** command to one of the service nodes as shown in Example 5-22.

Example 5-22 Monitoring the status of the jobs in a service node

```
xdsh f01sv01-v llstatus
```

Stopping LoadLeveler

Shutting down LoadLeveler early in the process reduces any chances of job submission, and eliminates any LoadLeveler dependencies on the cluster.

As described in “Cluster shutdown assumptions” on page 282, it is necessary to drain or cancel all jobs and remove users from the system. It is assumed that all jobs are drained or canceled by using the steps that were described previously in this section. Because there are no jobs active in the system, this step describes shutting down LoadLeveler down.

LoadLeveler must be stopped on all compute nodes and service nodes, as shown in Example 5-23 on page 285.

Example 5-23 Stopping LoadLeveler

```
xdsh compute -v -l loadl llrctl stop
xdsh service -v -l loadl llctl stop
```

Stopping GPFS and unmounting the file system

After LoadLeveler is stopped, GPFS also is stopped. It is important to ensure that all applications that need to access files within GPFS are stopped before performing this step. A single command is run on any storage node to complete this step. For this example, we use a storage node called f01st01 (for frame one storage node one), as shown in Example 5-24.

Example 5-24 Stopping GPFS

```
xdsh f01st01 -v mmshutdown -a
```

> **Important:** This command shuts down GPFS everywhere it is running in the cluster. After the command completes, GPFS is down and no longer available.

Optional: Capturing the HFI lInk status

Before the compute LPARs are shut down, it might be useful to get a state of the HFI link status. This status is useful if there are any HFI errors before shutting down so that the errors are understood when the cluster is restarted. This restart is done by listing the connection state for the BPAs and FSPs and listing the CEC link status.

Verify that CNM successfully contacted all BPAs and FSPs by issuing the command that is shown in Example 5-25.

Example 5-25 Checking the status of all the BPAs and FSPs

```
lsnwcomponents
```

Example 5-26 must match the number of CEC drawers that are in the cluster.

Example 5-26 Matching the number of the CEC drawer

```
lsnwloc | grep -v EXCLUDED | wc -l
```

If the number is incorrect, check for any issues that cause a CEC drawer to be excluded by CNM, as shown in Example 5-27.

Example 5-27 Checking issues with CEC

```
$ xdsh f01st01 -v mmshutdown -a
$ lsnwloc | grep EXCLUDED
```

Shutting down compute nodes

Now that LoadLeveler and GPFS are stopped, the next step is to shut down the compute nodes, as shown in Example 5-28. The compute nodes are shut down first because other nodes within the cluster do not depend on the compute nodes.

Example 5-28 Shutting down the compute nodes

```
xdsh compute -v shutdown -h now
```

The command that is shown in Example 5-29 verifies that the compute nodes are shutdown.

Example 5-29 Verifying the compute nodes are down

```
rpower compute state
```

Other utility nodes

In this step, any utility nodes (login, backup, and so on) are shutdown, as shown Example 5-30.

Example 5-30 Shutting down the utility nodes

```
xdsh login -v shutdown -h now
```

The command that is shown in Example 5-31 verifies that the utility nodes are stopped.

Example 5-31 Utility nodes stopped

```
rpower login state
```

Shutting down the storage nodes

LoadLeveler, GPFS, and the compute nodes are down. This means that all dependencies on the storage nodes are stopped and the storage nodes are shut down, as shown in Example 5-32.

Example 5-32 Shutting down the storage nodes

```
xdsh storage -v shutdown -h now
```

The command that is shown in Example 5-33 verifies that the storage nodes are shut down.

Example 5-33 Verifying the storage nodes are down

```
rpower storage state
```

Shutting down the service nodes

With the compute and storage nodes down, there are no other dependencies on the service nodes and the service nodes are shut down, as shown in Example 5-34.

Example 5-34 Shutting down the service nodes

```
xdsh service -v shutdown -h now
```

The command that is shown in Example 5-35 verifies that the service nodes are shut down.

Example 5-35 Verifies that the service nodes are down

```
rpower service state
```

Powering off the CECs

This section describes the process for shutting down the CECs.

After the compute, utility nodes (if any), storage, and service nodes are shut down, the CECs are powered off, as shown in Example 5-36 on page 287.

Example 5-36 Powering down the CEC

```
rpower cec off
```

The command that is shown in Example 5-37 verifies that the CECs are off.

Example 5-37 Verifying the CEC are off

```
rpower cec state
```

Placing the frames in rack standby mode

After all of the CECs are powered off and the Central Network Manager is off, the frames are placed in rack standby mode, as shown in Example 5-38.

Example 5-38 Frames that are placed into rack standby mode

```
rpower frame rackstandby
```

The command that is shown in Example 5-39 validates that the frames are in rack standby issue.

Example 5-39 Validating the frames are in rack standby

```
rpower frame state
```

Turning off the frames

After the frames enter rack standby, they are ready for power off. Manually turn of the red switch for each frame.

Shutting down EMS and HMCs

After all of the nodes and CECs are down and the frames are in rack standby, the EMS and HMCs are shutdown. Depending on the goal for this shutdown process, this step might be skipped.

If the goal is to shut down only the 775 servers and attached storage, these steps are completed and you stop here in the process.

If the goal is to restart the entire cluster, including the EMS and HMCs, you must continue with the following process to shut down the HMCs and the EMS:

1. Log in to the HMC and issue the following command to shut down the HMCs:

   `$ `**`hmcshutdown -t now`**

2. Shut down the primary and backup EMS servers. Start with the backup EMS by issuing the following command:

   `$ `**`shutdown -h now`**

3. On the primary EMS, shut down the following daemons in this order:

   i. Stop Teal:

      `$ `**`<need stopsrc`**`>`

   ii. Stop xcatd:

      `$ `**`<need stopsrc>`**

   iii. Stop dhcp:

      `$ `**`<need stopsrc>`**

iv. Stop named:

```
$ <need stopsrc>
```

v. Stop the xcat db:

```
$ su - xcatdb
```

```
$ db2stop
```

```
$ exit
```

4. If the stop is unsuccessful, use the following force:

```
$ su - xcatdb
```

```
$ db2stop force
```

```
$ exit
```

5. Unmount the shared file systems.

   Your site directory names might vary from the following sample:

```
$ umount ? /etc/xcat
$ umount ? /install
$ umount ? ~/.xcat
$ umount ? /databaseloc
```

6. Shut down the primary EMS:

```
$ shutdown -h now
```

   After the primary is shut down, the backup is shut down.

7. Log in to the backup EMS and issue the following command:

```
$ shutdown -h now
```

Turning off the external disks that are attached to the EMS

After the primary and backup EMS are shutdown, you turn off the external disk drives.

Optional: Turning off breakers or disconnecting power

Now that all of the cluster-related hardware is turned off and the EMS and HMCs are down, the power for the management rack and the 775 frames are turned off. If you have breaker switches, these switches might be used. If not you do not have breaker switches, the power is disconnected from the management rack and the frames by disconnecting the power cords on the IBM Power System 775 management rack.

> **Handle with care:** Care must be taken when power to the hardware is handled.

All software and hardware for the cluster is now stopped and the process is complete.

5.3 Managing cluster nodes

This section describes the process to manage the types of nodes in Power 755 cluster and some solutions with xCAT.

5.3.1 Node types

Several types of nodes are featured in the Power 755 cluster in the xCAT database, including four hardware nodes and one LPAR node. For the LPAR node, there are four subtype nodes that are used to distinguish in logical function.

For more information, see the Define the Frame Name/MTMS Mapping and Define the LPAR Nodes and Create the Service/Utility LPARs sections of the xCAT Power 775 Hardware Management document at these websites:

- http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Power_775_Hardware_Management#Define_the_Frame_Name.2FMTMS_Mapping
- http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Power_775_Hardware_Management#Define_the_LPAR_Nodes_and_Create_the_Service.2FUtility_LPARs

Hardware nodes

In hardware roles, the following types of nodes are included in the Power 755 cluster:

- Frame node

  A node with hwtype set to frame represents a high-end IBM Power Systems server 24-inch frame, as shown in Example 5-40.

  Example 5-40 Frame node in xCAT database

```
# lsdef frame15
Object name: frame15
    groups=frame,all
    hcp=frame15
    hidden=0
    hwtype=frame
    id=15
    mgt=bpa
    mtm=78AC-100
    nodetype=ppc
    postbootscripts=otherpkgs
    postscripts=syslog,aixremoteshell,syncfiles
    serial=992003N
    sfp=c250hmc21
```

  The following attributes are included in Example 5-40:

  – hcp: The hardware control point for this frame. For DFM, this point is always set to itself.

  – id: The frame number.

  – mgt: The type of the hardware control point (hcp). A bpa setting means xCAT manages the point directly without the HMC.

  – mtm: The machine type and model.

  – parent: The parent for this frame. This setting is set to blank, or contains the building block number.

  - serial: The serial number of the frame.
  - sfp: The HMC that is connected to this frame for collecting hardware-serviceable events.
- BPA node

  A node with a hwtype is set to BPA and represents one port on one BPA (each BPA has two ports). For the purposes of thexCAT, the BPA is the service processor that controls the frame. The relationship between the Frame node and the BPA node from the perspective of a system administrator is that the administrator must always use the Frame node definition for the xCAT hardware control commands. xCAT determines which BPA nodes and their IP addresses to use for hardware service processor connections, as shown in Example 5-41.

  Example 5-41 BPA node in xCAT database

```
# lsdef -S 40.14.0.1
Object name: 40.14.0.1
    groups=bpa,all,f14bpa
    hcp=40.14.0.1
    hidden=1
    hwtype=bpa
    id=14
    ip=40.14.0.1
    mac=001a6454e64a
    mgt=bpa
    mtm=78AC-100
    nodetype=ppc
    parent=frame14
    postbootscripts=otherpkgs
    postscripts=syslog,aixremoteshell,syncfiles
    serial=992003L
    side=A-0
```

  Example 5-41 includes the following attribute meanings:

  - side - <BPA>-<port>**:** The side attribute refers to which BPA, A or B, is determined by the slot value that is returned from **lsslp** command. The side also lists the physical port within each BPA that is determined by the IP address order from the **lsslp** response. This information is used internally during communication with the BPAs.
  - parent: This attribute is always set to the frame node that of which this BPA is part.
  - mac: The MAC address of the BPA, which is acquired from the **lsslp** command.
  - hidden: Set to 1 means that xCAT hides the node by default in **nodels** and **lsdef** output. BPA nodes often are hidden because you must use the frame nodes for management. To see the BPA nodes in the **nodels** or **lsdef** output, use the **-S** flag.
- CEC node

  This node features the attribute hwtype set to cec which represents a Power Systems CEC (for example, one physical server). Refer to Example 5-42 on page 291.

Example 5-42 CEC node in xCAT database

```
# lsdef f15cec01
Object name: f15cec01
    groups=cec,all,f15cec
    hcp=f15cec01
    hidden=0
    hwtype=cec
    id=3
    mgt=fsp
    mtm=9125-F2C
    nodetype=ppc
    parent=frame15
    postbootscripts=otherpkgs
    postscripts=syslog,aixremoteshell,syncfiles
    serial=02C4DB6
    sfp=c250hmc21
    supernode=2,0
```

Example 5-42 includes the following attributes:

- `hcp`: The hardware control point for this CEC. For DFM, the HCP always is set to itself.
- `id`: The cage number of this CEC in a 24-inch frame.
- `mgt`: This attribute is set to fsp.
- `mtm`: Indicates the machine type and model.
- `parent`: The frame node in which this CEC is located.
- `serial`: The serial number of the CEC.
- `sfp`: The HMC that is connected to this CEC for collecting hardware-serviceable events.

`supernode`: The HFI network supernode number of which this CEC is a part. For more information about the value to which this number must be set, see the *Power Systems High performance clustering using the 9125-F2C Planning and Installation Guide*, at this website:

http://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+Recommended+Installation+Sequence+-+Version+1.0#IBMHPCClusteringwithPower775RecommendedInstallationSequence-Version1.0-ISNMInstallation

Important: In addition to setting the CEC supernode numbers, set the HFI switch topology value in the xCAT site table. For more information about the value to which this number must be set, see the *Power Systems High performance clustering using the 9125-F2C Planning and Installation Guide*, at this website:

http://www.ibm.com/developerworks/wikis/display/hpccentral/IBM+HPC+Clustering+with+Power+775+Recommended+Installation+Sequence+-+Version+1.0#IBMHPCClusteringwithPower775RecommendedInstallationSequence-Version1.0-ISNMInstallation

FSP node

FSP node is a node with the hwtype set to fsp and represents one port on the FSP. In one CEC with redundant FSPs, there are two FSPs and each FSP has two ports. There are four FSP nodes that are defined by xCAT per server with redundant FSPs. Similar to the relationship between Frame node and BPA node, the system administrator always uses the CEC node for the hardware control commands. xCAT automatically uses the four FSP node definitions and their attributes for hardware connections, as shown in Example 5-43 on page 292.

Example 5-43 FSP node in xCAT database

```
# lsdef -S 40.14.1.1
Object name: 40.14.1.1
    groups=fsp,all,f14fsp
    hcp=40.14.1.1
    hidden=1
    hwtype=fsp
    id=3
    ip=40.14.1.1
    mac=00215ee990dc
    mgt=fsp
    mtm=9125-F2C
    nodetype=ppc
    parent=f14cec01
    postbootscripts=otherpkgs
    postscripts=syslog,aixremoteshell,syncfiles
    serial=02C5096
    side=A-0
```

Example 5-43 shows the following attribute meanings:

- `side - <FSP>-<port>`: The side attribute refers to the FSP (A or B) that is determined by the slot value that is returned from **`lsslp`** command. It also lists the physical port within each FSP, which is determined by the IP address order from the **`lsslp`** response. This information is used internally when communicating with the FSPs.
- `parent`: Set to the CEC in which this FSP is located.
- `mac`: The mac address of the FSP, which is retrieved from **`lsslp`**.
- `hidden`: Set to 1 means that xCAT hides the node by default in **`nodels`** and **`lsdef`** output. FSP nodes are hidden because you use only the CEC nodes for management. To see the FSP nodes in the **`nodels`** or **`lsdef`** output, use the **`-S`** flag.

DFM: For the Power 755 cluster, we use DFM to communicate directly to the Power Systems server's service processor without the use of the HMC for management.

LPAR types of nodes

There are four types of nodes in a Power 755 cluster:

- Service node

  This is an LPAR node that helps the hierarchical management of xCAT by extending the capabilities of the EMS. The service node features a full disk image and is used to serve the diskless OS images for the nodes that it manages, as shown in Example 5-44.

  Example 5-44 The NIM OS image in a service node

  ```
  # lsnim | grep GOLD_71Bdskls
  GOLD_71Bdskls                      resources       spot
  GOLD_71Bdskls_dump                 resources       dump
  GOLD_71Bdskls_paging               resources       paging
  GOLD_71Bdskls_shared_root          resources       shared_root
  GOLD_71Bdskls_resolv_conf          resources       resolv_conf
  ```

- Compute node

  This node is used for customer applications. Compute nodes in a 775 cluster include no local disks or Ethernet adapters. The nodes are diskless nodes, as shown in Example 5-45.

Example 5-45 The share_root mode in compute node (diskless node)

```
# mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
SN.ibm.com /install/nim/shared_root/GOLD_71Bdskls_shared_root /
stnfs  xxx xx xx:xx hard
SN.ibm.com /install/nim/spot/GOLD_71Bdskls/usr /usr            nfs3   xxx xx
xx:xx ro,hard,intr,acl,nimperf
```

- I/O node

  This LPAR node features attached disk storage and provides access to the disk for applications. In IBM Power 775 clusters, the I/O node runs GPFS and manages the attached storage as part of the GPFS storage, as shown in Example 5-46.

Example 5-46 Split the I/O slots C9-C12 (connected to DE) for the IO node f10c12ap13-`hf0`

```
# lsvm f10c12ap13-hf0
13: 537/U78A9.001.1122233-P1-C9/0x21010219/2/13
13: 536/U78A9.001.1122233-P1-C10/0x21010218/2/13
13: 529/U78A9.001.1122233-P1-C11/0x21010211/2/13
13: 528/U78A9.001.1122233-P1-C12/0x21010210/2/13
```

Example 5-47 shows the topology information of an I/O node.

Example 5-47 The topology info of an IO node

```
# topsummary /gpfslog/pdisktopology.out
P7IH-DE enclosures found: 000DE22
Enclosure 000DE22:
Enclosure 000DE22 STOR P1-C4/P1-C5 sees both portcards: P1-C4 P1-C5
Portcard P1-C4: ses20[0154]/mpt2sas0/24 diskset "37993" ses21[0154]/mpt2sas0/24
diskset "18793"
Portcard P1-C5: ses28[0154]/mpt2sas2,mpt2sas1/24 diskset "37993"
ses29[0154]/mpt2sas2,mpt2sas1/24 diskset "18793"
Enclosure 000DE22 STOR P1-C4/P1-C5 sees 48 disks
Enclosure 000DE22 STOR P1-C12/P1-C13 sees both portcards: P1-C12 P1-C13
Portcard P1-C12: ses22[0154]/mpt2sas0/24 diskset "26285"
ses23[0154]/mpt2sas0/23 diskset "44382"
Portcard P1-C13: ses30[0154]/mpt2sas2,mpt2sas1/24 diskset "26285"
ses31[0154]/mpt2sas2,mpt2sas1/23 diskset "44382"
Enclosure 000DE22 STOR P1-C12/P1-C13 sees 47 disks
Enclosure 000DE22 STOR P1-C20/P1-C21 sees both portcards: P1-C20 P1-C21
Portcard P1-C20: ses36[0154]/mpt2sas2,mpt2sas1/24 diskset "04091"
ses37[0154]/mpt2sas2,mpt2sas1/24 diskset "31579"
Portcard P1-C21: ses44[0154]/mpt2sas3/24 diskset "04091"
ses45[0154]/mpt2sas3/24 diskset "31579"
Enclosure 000DE22 STOR P1-C20/P1-C21 sees 48 disks
Enclosure 000DE22 STOR P1-C28/P1-C29 sees both portcards: P1-C28 P1-C29
Portcard P1-C28: ses38[0154]/mpt2sas2,mpt2sas1/24 diskset "64504"
ses39[0154]/mpt2sas2,mpt2sas1/24 diskset "52307"
```

```
Portcard P1-C29: ses46[0154]/mpt2sas3/24 diskset "64504"
ses47[0154]/mpt2sas3/24 diskset "52307"
Enclosure 000DE22 STOR P1-C28/P1-C29 sees 48 disks
Enclosure 000DE22 STOR P1-C60/P1-C61 sees both portcards: P1-C60 P1-C61
Portcard P1-C60: ses32[0154]/mpt2sas2,mpt2sas1/24 diskset "27327"
ses33[0154]/mpt2sas2,mpt2sas1/24 diskset "43826"
Portcard P1-C61: ses40[0154]/mpt2sas3/24 diskset "27327"
ses41[0154]/mpt2sas3/24 diskset "43826"
Enclosure 000DE22 STOR P1-C60/P1-C61 sees 48 disks
Enclosure 000DE22 STOR P1-C68/P1-C69 sees both portcards: P1-C68 P1-C69
Portcard P1-C68: ses34[0154]/mpt2sas2,mpt2sas1/24 diskset "05822"
ses35[0154]/mpt2sas2,mpt2sas1/24 diskset "59472"
Portcard P1-C69: ses42[0154]/mpt2sas3/24 diskset "05822"
ses43[0154]/mpt2sas3/24 diskset "59472"
Enclosure 000DE22 STOR P1-C68/P1-C69 sees 48 disks
Enclosure 000DE22 STOR P1-C76/P1-C77 sees both portcards: P1-C76 P1-C77
Portcard P1-C76: ses16[0154]/mpt2sas0/24 diskset "37499"
ses17[0154]/mpt2sas0/24 diskset "34848"
Portcard P1-C77: ses24[0154]/mpt2sas2,mpt2sas1/24 diskset "37499"
ses25[0154]/mpt2sas2,mpt2sas1/24 diskset "34848"
Enclosure 000DE22 STOR P1-C76/P1-C77 sees 48 disks
Enclosure 000DE22 STOR P1-C84/P1-C85 sees both portcards: P1-C84 P1-C85
Portcard P1-C84: ses18[0154]/mpt2sas0/24 diskset "33798"
ses19[0154]/mpt2sas0/24 diskset "40494"
Portcard P1-C85: ses26[0154]/mpt2sas2,mpt2sas1/23 diskset "56527"
ses27[0154]/mpt2sas2,mpt2sas1/24 diskset "40494"
Enclosure 000DE22 STOR P1-C84/P1-C85 sees 48 disks
Carrier location P1-C40-D1 appears empty but should have an HDD
Carrier location P1-C86-D3 appears only on the portcard P1-C84 path
Enclosure 000DE22 sees 383 disks

mpt2sas3[1005480000] U78A9.001.1122233-P1-C9-T1 000DE22 STOR 3 P1-C21 (ses44
ses45) STOR 4 P1-C29 (ses46 ses47) STOR 5 P1-C61 (ses40 ses41) STOR 6 P1-C69
(ses42 ses43)
mpt2sas2[1005480000] U78A9.001.1122233-P1-C10-T1 000DE22 STOR 3 P1-C20 (ses36
ses37) STOR 4 P1-C28 (ses38 ses39) STOR 5 P1-C60 (ses32 ses33) STOR 6 P1-C68
(ses34 ses35)
mpt2sas1[1005480000] U78A9.001.1122233-P1-C11-T1 000DE22 STOR 1 P1-C5 (ses28
ses29) STOR 2 P1-C13 (ses30 ses31) STOR 7 P1-C77 (ses24 ses25) STOR 8 P1-C85
(ses26 ses27)
mpt2sas0[1005480000] U78A9.001.1122233-P1-C12-T1 000DE22 STOR 1 P1-C4 (ses20
ses21) STOR 2 P1-C12 (ses22 ses23) STOR 7 P1-C76 (ses16 ses17) STOR 8 P1-C84
(ses18 ses19)
```

- Utility node

  This general term refers to a non-compute node/LPAR and a non-I/O node/LPAR. Examples of LPARs in a utility node are the service node, login node, and local customer nodes for backing up of data or other site-specific functions.

- Login node

  This LPAR node is defined to allow the users to log in and submit the jobs in the cluster. The login node most likely includes an Ethernet adapter that is connecting to the customer VLAN for access. For more information about setting up the login node, see the Granting Users xCAT privileges document in the setup login node (remote client) section at this website:

  http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Granting_Users_xCAT_privileges#Setup_Login_Node_.28remote_client.29

5.3.2 Adding nodes to the cluster

You need to define the types of nodes with xCAT commands in a Power 755 cluster before adding the nodes into the cluster.

For more information about defining cluster nodes, see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Defining_cluster_nodes_on_System_P

There are several ways to create xCAT cluster node definitions. Use the following method that best fits your situation.

Using the mkdef command

You use the xCAT **mkdef** command to create basic xCAT node definitions, as shown in Example 5-48.

Example 5-48 Define a node by xCAT command mkdef

```
# mkdef -t node -o clstrn27 mgt=hmc os=AIX nodetype=ppc,osi hwtype=lpar
groups=lpar,all,compute
```

To get a description of all the valid attributes that might be specified for a node definition, you run the following **lsdef** command:

```
lsdef -t node -h | pg
```

Using an xCAT stanza file

A stanza file contains information that is used to create xCAT data object definitions. A stanza file is used as input to several xCAT commands. The stanza file contains one or more stanzas that provide information for individual object definitions, as shown in Example 5-49.

Example 5-49 xCAT stanza file for nodes

```
# cat lpar_stanza.txt
f10c12ap01:
        objtype=node
        nodetype=ppc,osi
        id=1
        os=AIX
        hcp=cec12
        parent=cec12
        groups=lpar,all,service,llserver
        mgt=fsp
        cons=fsp
        hwtype=lpar
```

After you create a stanza file, you pass the information to the following **mkdef** command:

```
cat lpar_stanza.txt | mkdef -z
```

Adding nodes into node groups in the cluster

After the nodes are defined, you add the nodes into the cluster groups to manage the nodes. A node group is a named collection of cluster nodes that are used as a simple way to target an action to a specific set of nodes. The node group names are used in any xCAT command that targets a node range.

There are two ways to create xCAT static node groups: you set the groups attribute of the node definition or you create a group definition directly, as shown in Example 5-50.

Example 5-50 Create a group definition by the mkdef command

```
# mkdef -t group -o aixnodes members="node01,node02,node03"
# lsdef -t group -o aixnodes
Object name: compute
    grouptype=static
    members=node01,node02,node03
```

You then manage the nodes by groups with xCAT commands.

5.3.3 Removing nodes from a cluster

This section describes the process of removing the nodes from the cluster.

Removing compute nodes

Complete the following steps to remove compute nodes from the service node and xCAT database:

1. Check the status of the nodes:
   a. Shut down the nodes if the status is “Running”:

      ```
      # rpower <noderange> stat
      <noderange>: Running
      # xdsh <noderange> -v shutdown -h now
      ```

   b. Or, force that is powering off the nodes, but shut down gracefully:

      ```
      # rpower <noderange> off
      ```

   c. Recheck the status of the node:

      ```
      # rpower <noderange> stat
      ```

2. Remove the node definitions of NIM machine from the service node:

   ```
   # rmdsklsnode -V -f <noderange>
   ```

   This action might cause a side affect of leaving the alloc_count of spot and share root for the nodes to “1”:

   ```
   #/usr/sbin/lsnim -a alloc_count -Z GOLD_71Bdskls_1132A_HPC
   #name:alloc_count:
   GOLD_71Bdskls_1132A_HPC:1:
   #/usr/sbin/lsnim -a alloc_count -Z GOLD_71Bdskls_1132A_HPC_shared_root
   #name:alloc_count:
   GOLD_71Bdskls_1132A_HPC_shared_root:1:
   ```

Set the alloc_count to “0”:

```
#/usr/lpp/bos.sysmgt/nim/methods/m_chattr -a alloc_count=0 GOLD_71Bdskls_1132A_HPC
#/usr/lpp/bos.sysmgt/nim/methods/m_chattr -a alloc_count=0
GOLD_71Bdskls_1132A_HPC_shared_root
```

Check the alloc_count:

```
# /usr/sbin/lsnim -a alloc_count -Z GOLD_71Bdskls_1132A_HPC
#name:alloc_count:
GOLD_71Bdskls_1132A_HPC:0:
# /usr/sbin/lsnim -a alloc_count -Z GOLD_71Bdskls_1132A_HPC_shared_root
#name:alloc_count:
GOLD_71Bdskls_1132A_HPC_shared_root:0:
```

3. Remove the node definitions from xCAT group:

   ```
   #chdef -t group compute -m members=noderange
   ```

4. Remove the node definitions from xCAT database:

   ```
   #rmdef -t node noderange
   ```

5.4 Power 775 Availability Plus

To minimize the amount of service windows that are needed to repair defective hardware in a cluster and maximize customer satisfaction with regard to these expectations, the Availability Plus (A+) strategy is implemented. This strategy also is called Fail in Place (FIP) and some terms still refer to this older naming convention.

The A+ event is implemented by redundant or spare hardware that is activated in a failure event or covered by redundant hardware that continues to run the system operation. Although the A+ event does not require a service action, such as hardware replacement, some administrative actions must be performed. The failed hardware remains in the system and no service action is initiated by the A+ event. A repair action is initiated if there are enough failures that are reached by the FIP threshold.

For more information about service procedures, see the *POWER Systems High Performance clustering using the 9125-F2C Service Guide* at this website:

https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_service_guide.pdf?version=1

5.4.1 Advantages of Availability Plus

The use of Availability Plus (A+) features the following advantages:

- Higher system availability time:
  - Mean time to actual physical repair is improved
  - No ship time delay while waiting for parts to arrive
  - Reduced repair time compared to regular replacement of parts
  - Risk early life hardware failures are reduced because A+ hardware is used
- The customer has access to the A+ hardware for more system capacity:
  - A+ hardware is already delivered together with the wanted customer configuration
  - Customer uses the extra hardware for more production capacity
  - A+ hardware is used to test patches, perform fixes without the need to schedule special service windows on the actual production hardware

5.4.2 Considerations for A+

The following are considerations for the use of A+:

- Power: More resources are available to the customer that consume power. One option is to power off unused octants until they are needed.
- Space: The extra hardware requires more space. The configuration of the customer might require an additional CEC or extra Frame.
- Cooling: For some customer configurations, moe cooling might be required.
- Performance: The overall system performance degrades over time with an A+ resources failing. For more information, see Figure 5-2 on page 307.

5.4.3 A+ resources in a Power 775 Cluster

The following resources in the CEC/Cluster are used for A+:

- System planar
- Quad Chip Modules (QCMs)
- Hub modules with internal optical fibers

Figure 5-1 shows the following aspects of the A+ strategy:

- At installation time, all the Fail in Place and workload resources are available.
- When resources fail over the time, failed resources are replaced by the resources that are provided by A+.
- Even with multiple failures during the run time of the maintenance contract, the baseline committed workload resource is met without performing any physical repairs.

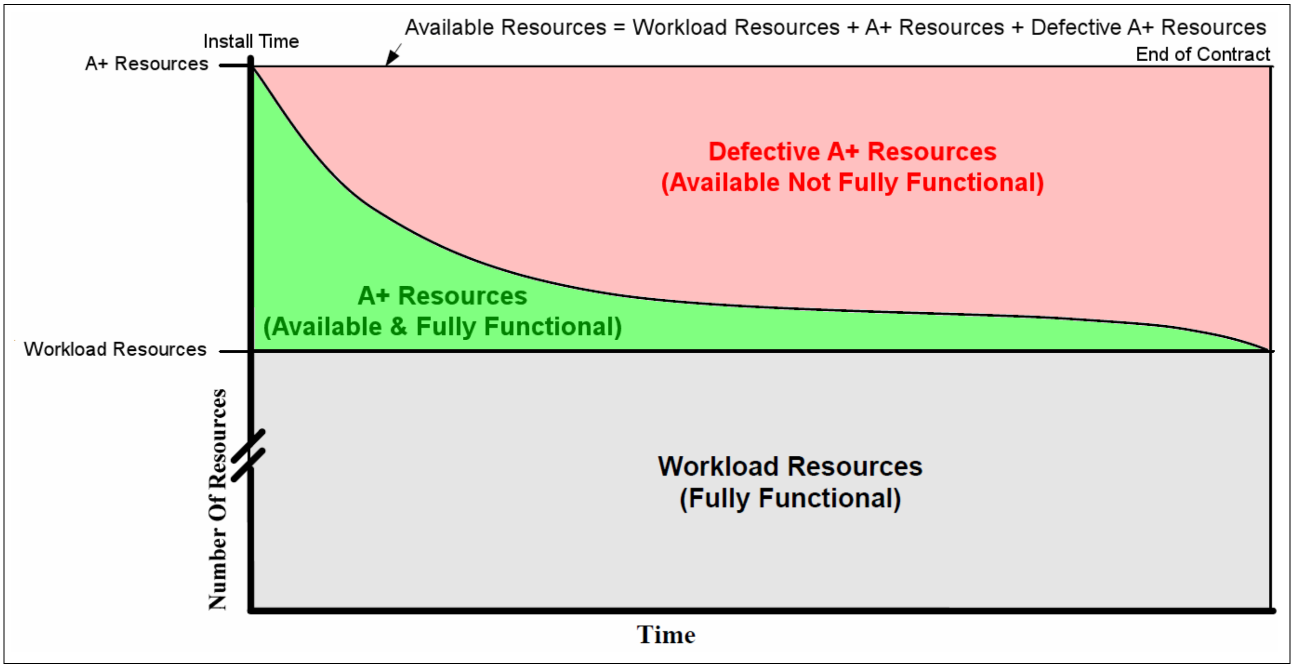


Figure 5-1 Depletion Curve

In the initial planning for the A+ resources of a cluster, it is the goal to keep the required resources fully functional during the lifetime of a cluster. The A+ resources are set in the lifespan to replace faulty components, and allow the customer to run applications with the computing power that is initially ordered.

5.4.4 Identifying an A+ resource

When a hardware serviceable event is reported on a Power 775 System, a field-replaceable unit (FRU) list is included. The A+ procedures must be performed as soon as an A+ resource is reported in the FRU list. For more information about identifying an A+ resource, see Table on page 299.

Many tasks that are performed by the administrator use xCAT commands to reassign the resources. For more information, see this website:

https://sourceforge.net/apps/mediawiki/xcat/index.php?title=Cluster_Recovery

An A+ resource is defined when the following conditions occur:

- The FRU location includes an “R” identifier. For example, U787C.001.1234567-P1-R1.
- The following ranges of locations are defined by using regular expressions:
  - U*-P1-R[1-8] are HFI hub modules. For example: U*-P1-R7
  - U*-P1-R[1-8]-R[1-40] are optical module port locations that are contained on the hub modules: U*-P1-R[1-8]. For example, U*-P1-R3-R5.U*-P1-R[9-40] are processor chip locations.

  For example, U*-P1-R10 are in groups of four per QCM. Also, U*-P1-R[13-16] are all on the same QCM, which is in the second octant.

The FRU location is an HFI port location.U*-P1-T[1-8]-T* and U-P1-T[9-17]-T* are D-link port connector locations on the bulkhead.U*-P1-T9 is the LR-link port connector location on the bulkhead.

> **Important:** The location identifier U787C.001.1234567 often is noted in examples that use the shorthand of U*.

5.4.5 A+ definitions

Although product names changed from FIP to A+, the definitions and functions of the products remain the same, as shown in Table 5-1.

Table 5-1 A+ definitions

Definition	Description	
A+ / Fail in Place Component	All A+ features including Octants and fiber optic interfaces.	
A+ / Fail in Place Event	A failure event that involves an A+ component or FRU element that is left in the failed state in the system.	
A+ / FIP Refresh Threshold	The minimum number of A+ components is available and at that point a hardware replacement is required. The threshold is determined from a table in which the values are set according to the contract policy, expected failure rates, and the amount or time that is remaining on the maintenance contract. There are individual thresholds for different failure types	
A+ / FIP Reset Threshold	The minimum number of A+ components that are needed to restore the system to following the repair of A+ components. The amount of hardware that is replaced is determined from a table in which the values are dependent on the component, and the amount of time that remaining on the service contract. There are individual thresholds for different failure types	
Compute QCM/Octant/Node	A QCM/Octant/Node without I/O adapters assigned to it. It is used solely for running application code, and often runs degraded.	
Non-compute QCM/Octant/Node	This is a QCM/Octant/Node with I/O adapters assigned to it. It is used for disk or I/O access and often must retain full function.	

The administrator must set up xCAT A+ node groups that must work with the A+ environment. One xCAT node group called “Aplus_defective” must be set up for any found A+ defective nodes or octants. A second xCAT node group “Aplus_available” must list the A+ available nodes or octants. You use the xCAT **mkdef** command to create the node groups, and then use the **chdef** command to associate any node (octants) to the proper node group.

The following commands are used to define an A+ group and add a failed resource:

- **`mkdef -t group -o Aplus_defective`**

  Creates an Aplus_defective group that must be empty.

- **`mkdef -t group -o Aplus_available members="node1,node2,node3"`**

  Create an Aplus_available group with node1, node2, and node3.

- **`chdef -t group -o Aplus_defective members=[node]`**

  Adds a failed A+ node to the Aplus_defective resources group.

5.4.6 A+ components and recovery procedures

This section describes the tasks that are performed by the administrator or cluster user to gather problem data or recover from failures.

The following tasks must be performed to recover from a problem:

- Determine the resource that failed.
- Determine whether the resource failed before.
- Determine the effect of the failure on which type of node (if any).
- Perform the appropriate recovery action according to the findings and the spare policy.
- Report the problem to IBM (serviceable events from the HMC must report automatically via the electronic Service Agent).
- Gather data.

  On the EMS Server, it is necessary to gather some data to check or analyze possible FIP events. The xCAT script that is called **`gatherfip`** is in `/opt/xcat/sbin` that collects information about ISNM, SFP, and deconfigured resources. The generated `.gz` file is sent to the IBM service team.

  The console output is shown in Example 5-51.

  Example 5-51 gatherfip command

```
# /opt/xcat/sbin/gatherfip
-----------------------------------------------------------------------------
10/07/2011 10:39 - Start gatherfip
10/07/2011 10:39 - gatherfip Version 1.0
10/07/2011 10:39 - xCAT Executive Management Server hostname is c250mgrs40-itso
10/07/2011 10:39 - Writing hardware service alerts in TEAL to
gatherfip.hardware.service.events
10/07/2011 10:39 - Writing ISNM Alerts in TEAL to gatherfip.ISNM.events
10/07/2011 10:39 - Writing deconfigred resource information to
gatherfip.guarded.resources
10/07/2011 10:39 - Writing ISNM Link Down information to
gatherfip.ISNM.links.down
10/07/2011 10:39 - Created tar file containing these /var/log files:
gatherfip.log gatherfip.hardware.service.events gatherfip.ISNM.events
gatherfip.guarded.resources gatherfip.ISNM.links.down
10/07/2011 10:39 - Created compressed tar file
/var/log/c250mgrs40-itso.gatherfip.20111007_1039.tar.gz
10/07/2011 10:39 - End gatherfip#
# (10:39:24) c250mgrs40-itso [AIX 7.1.0.0 powerpc] /opt/teal/bin
```

  This data is used by IBM support to determine whether a hardware repair is necessary.

  More data is gathered by the xCAT Administrator includes the output of the commands, as shown in Table 5-2 on page 302.

Table 5-2 xCAT commands used for A+

Command	Description	
swapnode	**swapnode -c CURRENTNODE -f TARGETNODE** Swaps the IO and location information in the xCAT database between two nodes.	
rinv	**rinv FRAMENAME firm**, **rinv CECNAME firm** Collects firmware information.	
rpower	**rpower FRAMENAME stat** Shows system status.	
lsdef	**lsdef FRAMENAME -i hcp,id,mtm,serial**, **lsdef CECNAME -i hcp,id,mtm,serial** This command lists the attributes for xCAT table definitions. For FIP and service recovery, it is important to know the VPD information that is listed for the Frame and CECs in the P775 cluster. The important attributes for vpd are hcp (hardware control point), id (frame id # or cage id #), mtm (model type, machine) serial (serial #). The **lsdef** command is also used to validate the P775 LPAR or octant information. The important attributes that are listed with LPARs are cons (console), hcp (hardware control point), id (LPAR/octant id #), mac (Ethernet or HFI MAC address), parent (CEC object), os (operating system), xcatmaster (installation server xCAT SN or xCAT EMS).	
mkhwconn/rmhwconn	Commands are used to create or remove hardware connections for FSP and BPA nodes to HMC nodes or hardware server from the xCAT EMS.	

Example 5-52 shows sample outputs that are used to gather more information.

Example 5-52 multiple command output examples

```
# rinv cec12 deconfig
cec12: Deconfigured resources
cec12: Location_code               RID   Call_Out_Method
Call_Out_Hardware_State   TYPE
cec12: U78A9.001.1122233-P1        800

# rinv cec12 firm
cec12: Release Level  : 01AS730
cec12: Active Level   : 048
cec12: Installed Level: 048
cec12: Accepted Level : 048
cec12: Release Level Primary: 01AS730
cec12: Level Primary  : 048
cec12: Current Power on side Primary: temp

# rpower cec12 stat
cec12: operating

# lsdef cec12  -i hcp,id,mtm,serial
Object name: cec12
```

```
    hcp=cec12
    id=14
    mtm=9125-F2C
    serial=02D7695

# lsdef c250f10c12ap17 -i cons,hcp,id,mac,os,parent,xcatmaster
Object name: c250f10c12ap17
    cons=fsp
    hcp=cec12
    id=17
    mac=e4:1f:13:4f:e2:2c
    os=rhels6
    parent=cec12
    xcatmaster=193.168.0.102

# lsvm cec12
1: 520/U78A9.001.1122233-P1-C14/0x21010208/2/1
1: 514/U78A9.001.1122233-P1-C17/0x21010202/2/1
1: 513/U78A9.001.1122233-P1-C15/0x21010201/2/1
1: 512/U78A9.001.1122233-P1-C16/0x21010200/2/1
13: 537/U78A9.001.1122233-P1-C9/0x21010219/2/13
13: 536/U78A9.001.1122233-P1-C10/0x21010218/2/13
13: 529/U78A9.001.1122233-P1-C11/0x21010211/2/13
13: 528/U78A9.001.1122233-P1-C12/0x21010210/2/13
13: 521/U78A9.001.1122233-P1-C13/0x21010209/2/13
17: 553/U78A9.001.1122233-P1-C5/0x21010229/0/0
17: 552/U78A9.001.1122233-P1-C6/0x21010228/0/0
17: 545/U78A9.001.1122233-P1-C7/0x21010221/0/0
17: 544/U78A9.001.1122233-P1-C8/0x21010220/0/0
29: 569/U78A9.001.1122233-P1-C1/0x21010239/0/0
29: 568/U78A9.001.1122233-P1-C2/0x21010238/0/0
29: 561/U78A9.001.1122233-P1-C3/0x21010231/0/0
29: 560/U78A9.001.1122233-P1-C4/0x21010230/0/0
cec12: PendingPumpMode=1,CurrentPumpMode=1,OctantCount=8:
OctantID=0,PendingOctCfg=5,CurrentOctCfg=5,PendingMemoryInterleaveMode=2,Curren
tMemoryInterleaveMode=2;
OctantID=1,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,Curren
tMemoryInterleaveMode=1;
OctantID=2,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,Curren
tMemoryInterleaveMode=1;
OctantID=3,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,Curren
tMemoryInterleaveMode=2;
OctantID=4,PendingOctCfg=5,CurrentOctCfg=5,PendingMemoryInterleaveMode=2,Curren
tMemoryInterleaveMode=2;
OctantID=5,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,Curren
tMemoryInterleaveMode=1;
OctantID=6,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=1,Curren
tMemoryInterleaveMode=1;
OctantID=7,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,Curren
tMemoryInterleaveMode=2;
```

- Record the failed resource and use Table 5-3 for future reference.

Table 5-3 A+ failed resource record

Resource hostname	Location (Frame/Slot; SuperNode/Drawer)	Spare policy (hot/cold/warm)	Date deployed

It is possible to determine the resource by using the data gathered (including the serviceable event). See to Table 5-4 to cross-reference the FRU locations.

Table 5-4 FRU Location to A+ resource

FRU locations	Resource
U*-P1-R(9-40)	QCM
U*-P1-R(1-8) and no other FRU	Hub module
U*-P1-R(1-8) and U*-P1-R(1-8)-R(1-15) or U*-P1-R(1-8)-R28 and U*P1-T*-T*	D-Link
U*-P1-R(1-8) and U*-P1-R(1-8)-R(16-27) or U*-P1-R(1-8)-R(29-40) and U*-P1-T9	LR-Link
U*-P1-R(1-8)-R(1-15) or U*-P1-R(1-8)-R28 and no other FRU	D-Link optical module
U*-P1-R(1-8)-R(16-27) or U*-P1-R(1-8)-R(29-40) and no other FRU	LR-Link optical Module
Two separate locations with the same unit location (U*) and of the format: U*-P1-R(1-8). Indicates a failure in an interface between two hubs	LL-Link failure

- Check for any previous records that match the same failure. If previous records are found, often the problem is recovered already and no further action is needed.
- If the serviceable event indicates an LR-Link cable assembly and the only FRU information in the FRU list is U*-P1-T9, contact IBM immediately and open a problem management record (PMR).
- Determine the location of the failure:
  - Record the location code (for example, U787A.001.1234567). On the EMS, use xCAT to cross-reference the unit location to frame and drawer.
  - Use xCAT procedure to cross-reference the unit locations and determine the octant for the QCM, as shown in Table 5-5 on page 305.

Table 5-5 QCM to Octant Map

QCM Location	Octant
U*-P1-R(9-12)	0
U*-P1-R(13-16)	1
U*-P1-R(17-20)	2
U*-P1-R(21-24)	3
U*-P1-R(25-28)	4
U*-P1-R(20-32)	5
U*-P1-R(33-36)	6
U*-P1-R(37-40)	7
U*-P1-R1	0
U*-P1-R2	1
U*-P1-R3	2
U*-P1-R4	3
U*-P1-R5	4
U*-P1-R6	5
U*-P1-R7	6
U*-P1-R8	7

- If this failure is a QCM failure, you must determine the type of node (Compute or non-Compute node).
- Perform the necessary recovery procedure. For information about the recovery procedure, see Table 5-6 and this website:

  https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_service_guide.pdf

Table 5-6 A+ recovery procedure

Failed resource	Compute node	Non-compute node
QCM	Availability Plus: Recovering a Compute node	Availability Plus: Recovering a non-Compute node
Hub and QCM with SRC of B114RRRR	Availability Plus: Recovering a hub module	
Hub Modules	Availability Plus: Recovering a hub module	
D-Link	Availability Plus: Recovering a D-Link	
LR-Link	Availability Plus: Recovering an LR-Link	
D-Link optical module	Availability Plus: Recovering a D-Link	
LR-Link optical module	Availability Plus: Recovering an LR-Link	
LL-Link failure	Availability Plus: Recovering a failure that impacts a drawer. The network connectivity for the entire drawer is compromised.	

- Report the problem to IBM and open a PMR.

5.4.7 Hot, warm, and cold policies

The following polices are available for using the Fail in Place nodes in A+:

- *Hot swap policy*: The node is fully in use and provides more productive or non-productive compute processing power.
- *Warm swap policy*: The node is made available for the Power 775 cluster, but are not in use with any production workload.
- *Cold swap policy*: The resource is powered off and must be brought online when it is needed.

The IBM Power Systems 775 includes more compute nodes. The specific amount of resources is determined by IBM during the planning phase, and this hardware is available for the customer without paying any extra charges.

The additional resources are used as added compute nodes, test systems, and so on.

xCAT administrator: The xCAT administrator must determine how to use the resources and how to enable, allocate, and apply these resources according to the three policies. The A+ functionality must be maintained manually to track the Fail in Place resources allocation for the application workloads. This function is not automated.

5.4.8 A+ QCM move example

A QCM move (as shown in Figure 5-2 on page 307) is the first approach to move resources within the CEC drawer.

If a failure in a non-compute QCM that is used for GPFS, the functions and (possibly) the associated PCI slots must be moved to a fully functional compute QCM within the same functional CEC.

The replacement QCM is the next QCM. For example, if QCM0 is failing, the next octant QCM1 takes over the functions of QCM0.

To perform this move, the `swapnode` command (xCAT) is used to drain the original octant and move it to the next octant. The `swapnode` command performs the following tasks:

- All of the location information in the databases between the two nodes is swapped, including the attributes for ppc tables and nodepos tables.
- When the swap occurs in the same CEC, the slot assignments between the two LPARs also are swapped.

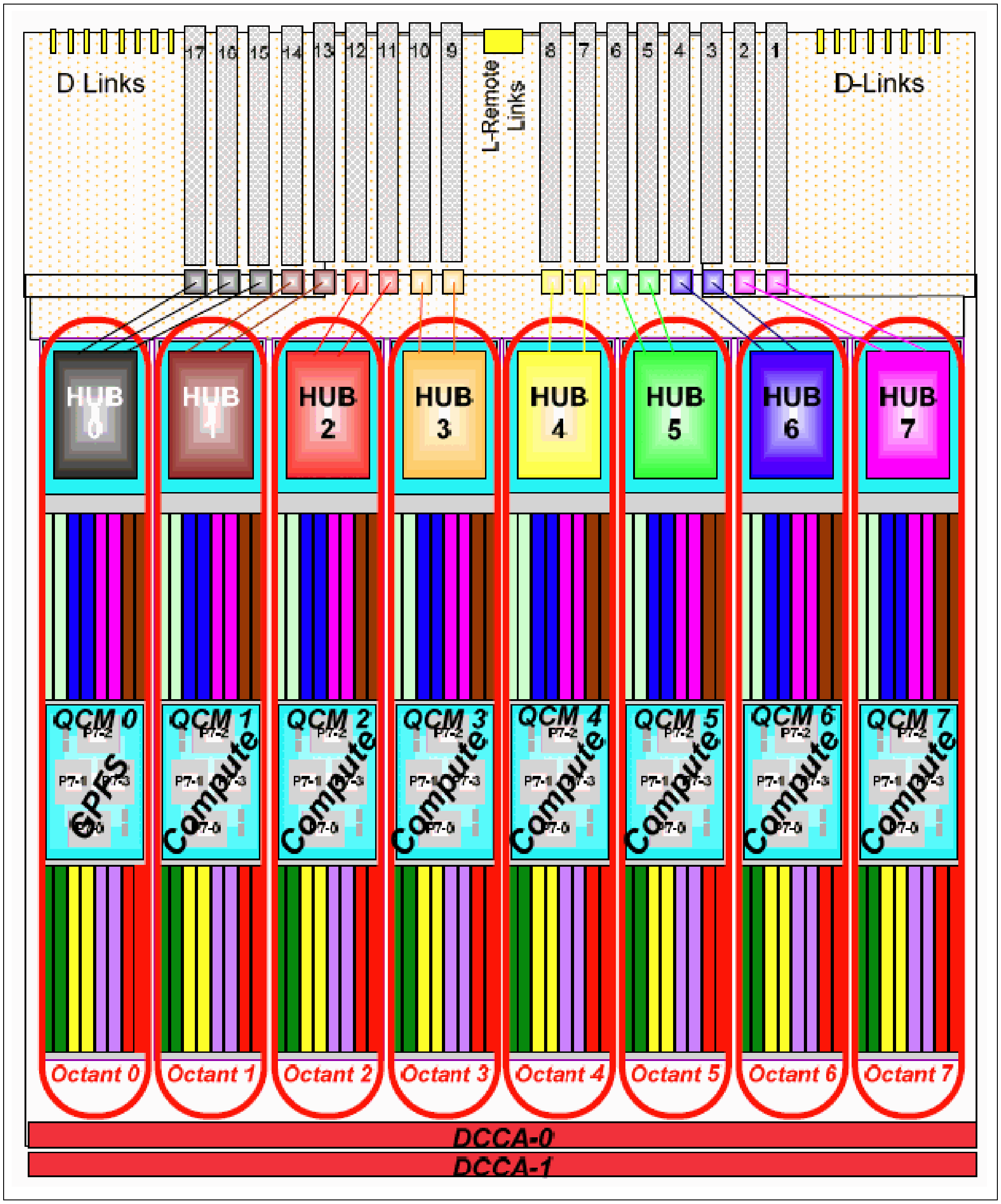


Figure 5-2 Original QCM layout

Figure 5-3 on page 308 shows that a non-compute QCM0 use for GPFS fails. The QCM0 functionality is moved to QCM1, QCM0 is redefined as a compute QCM, and resides in the defective A+ resource region.

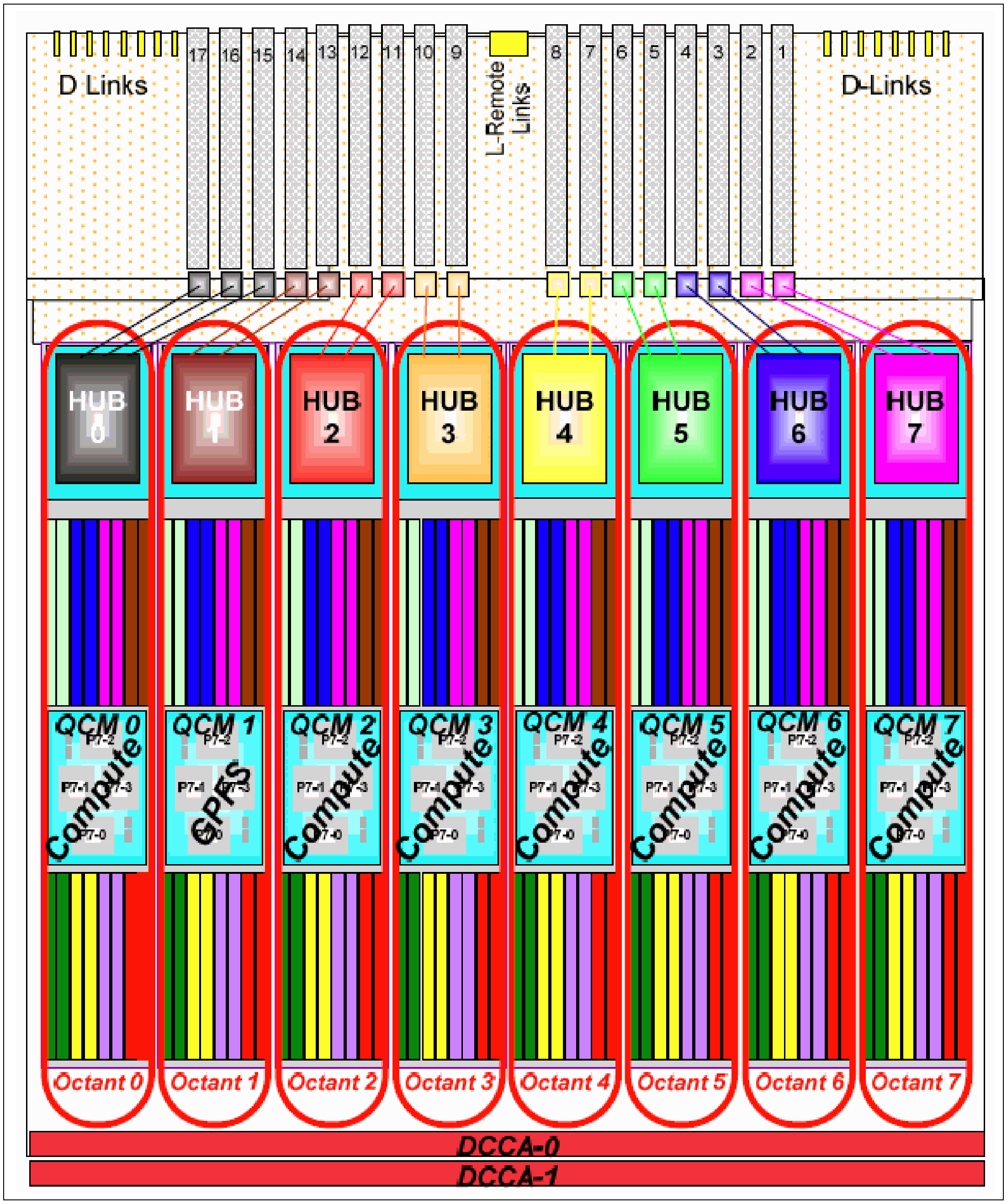


Figure 5-3 QCM0 to QCM1 move

If a failure occurs in the GPFS QCM1, QCM1 is moved to QCM2 and QCM1 is defined as a compute QCM again and resides in the defective A+ resource region, as shown in Figure 5-4 on page 309.

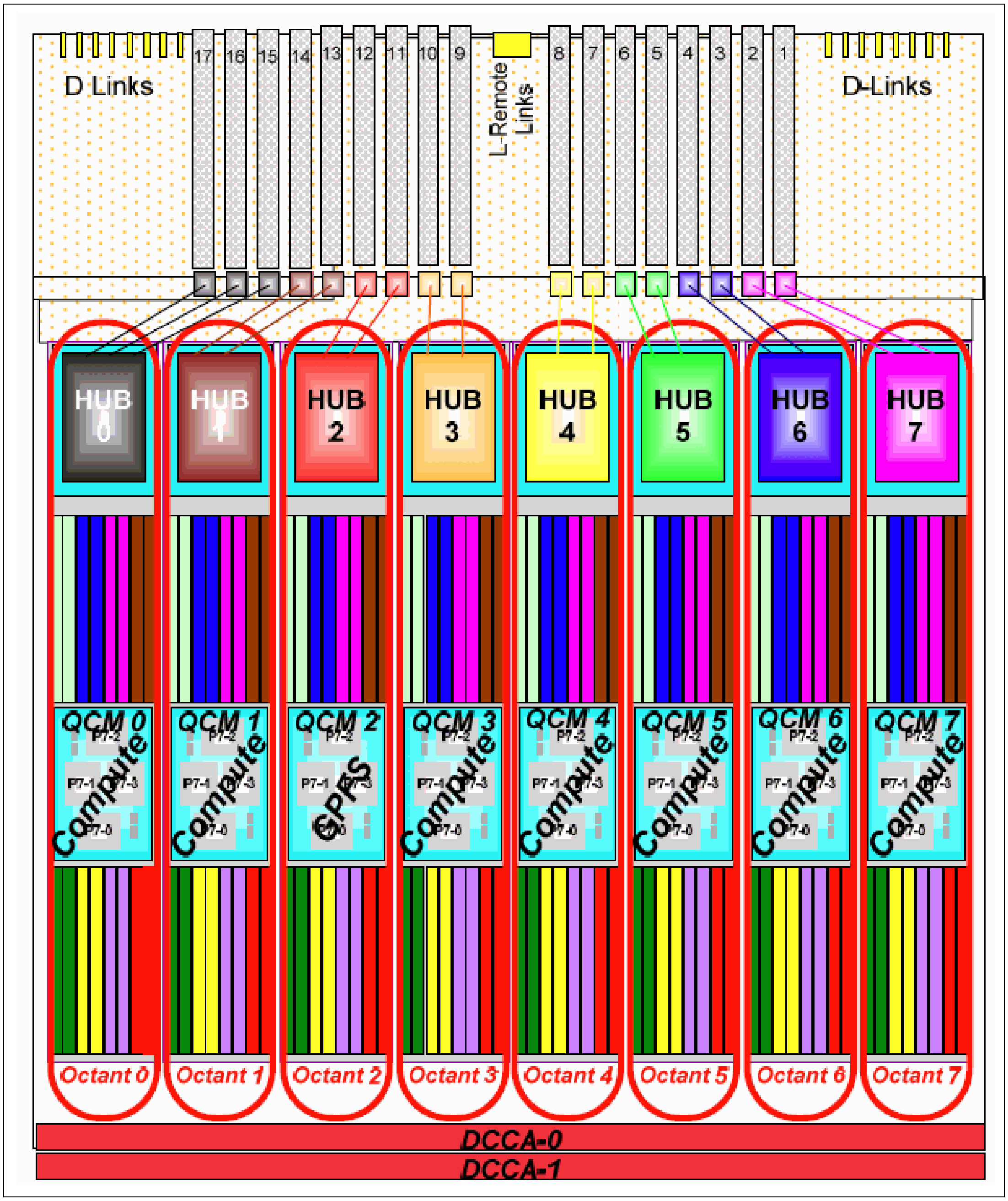


Figure 5-4 QCM1 to QCM2 move

This procedure is repeated until all compute QCMs are functionally replaced.

When all the functional QCMs in one CEC are exhausted, a CEC move is arranged. (The alternate CEC locations are defined by IBM.) A complete CEC is moved to a different CEC, which includes the PCI cards that are used by the other CEC. This move also requires the **swapnode** command and some physical maintenance for the PCI cards.

If no other CEC is available in the rack, the IBM SWAT Team swaps a CEC from one Frame to the Rack that requires the non-compute function.

5.4.9 A+ non-compute node overview

Compared to a compute node, a non-compute node A+ scenario is more sensitive regarding the tasks that must be performed to recover and make the non-compute node available again. You use a target compute node and a spare compute node to accomplish the swap. The target compute node is the new non-compute node after the swap is performed. The spare compute node is the node that fulfills the workload of the former compute node.

Non-compute nodes are defined in Table 5-7.

Table 5-7 non-Compute node configurations

Non-compute node type	Partition and recovery information
Service node	A service node is critical to maintaining operation on the nodes that it services. A disk and an Ethernet adapter are assigned to the node.
GPFS storage node	The GPFS storage node features SAS adapters that are assigned to it. If the GPFS storage node is still operational, ensure that there is an operational backup node or nodes for the set of disks that it owns before proceeding.
Login node	The login node features an Ethernet adapter that is assigned to it. If the login node is still operational, ensure that there is another operational login node or nodes before proceeding.
Other node types	Other non-Compute nodes often include adapters that are assigned to them. If this node provides a critical function to the cluster and it is still operational, you must confirm that a backup node is available to take over its function.

When a compute node is available to swap the resources, determine which node you must use to restore the non-compute node functions. Table 5-8 provides an overview of the tasks that must be performed based on the state of the A+ compute node.

Table 5-8 Compute node actions

Compute node location	Compute node state	Action to be performed on compute node
In drawer with the failed node	Hot spare	Prevent new jobs from starting and drain jobs from the compute node
	Warm spare	Prevent new jobs from starting and boot the partition
	Cold spare	No action required
	Workload resource	Prevent new jobs from and starting drain jobs from the compute node
In backup drawer	Hot spare	Prevent new jobs from starting and drain jobs from the compute node
	Warm spare	Prevent new jobs from starting and boot the partition
	Cold spare	No action required
	Workload resource	Prevent new jobs from starting and drain jobs from the compute node

For more information about performing these tasks, see the *Power Systems - High performance clustering using the 9125-F2C Service Guide* at this website:

https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_service_guide.pdf?version=1

A

Serviceable event analysis

This appendix contains examples of problem determination data that is used to debug reported errors in the Service Focal Point in the Hardware Management Console (HMC). When such error data is provided, the data is uploaded automatically from the HMC to the IBM system or the data is manually collected. In some cases, manual intervention is required to gather more command output for the analysis.

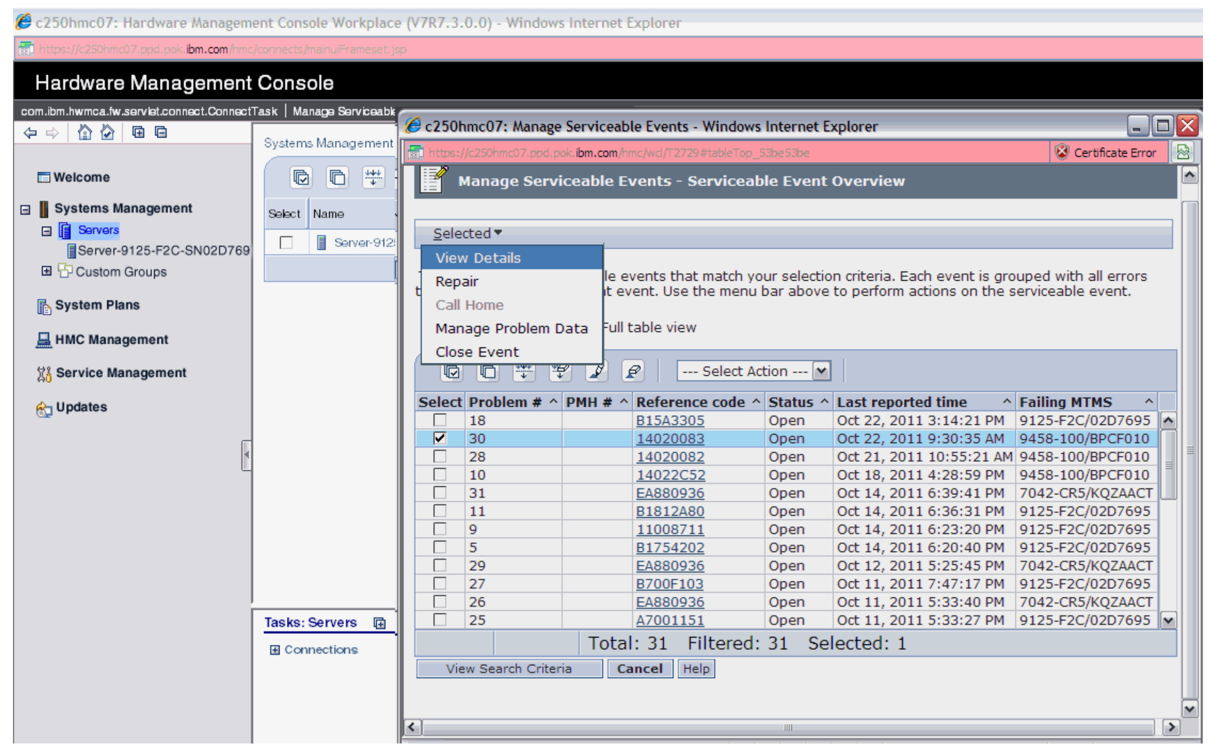
Analyzing a hardware serviceable event that points to an A+ action is described in this appendix.

Analyzing a hardware serviceable event that points to an A+ action

When hardware errors are encountered, often an event on the EMS server is presented, and on the HMC a serviceable event is logged. When the call home function is configured correctly, the HMC sends the gathered logs to an IBM system that collects all the data. The IBM support teams access this data minutes after the event appeared, and automatically an open problem record appears. If the maintenance contract is covering the time that the error occurred, a support person dispatches the call and works with the data.

A serviceable event on the HMC shows details about the involved hardware Field Replaceable Units (FRUs), time, date, error signature, and so on. In the manage serviceable events on the HMC, you view the details of these events. Also, the repair is started if the technical support team recommends such action to the IBM service support representative (SSR).

Figure A-1 shows the access to the serviceable events on the HMC.


c250hmc07: Hardware Management Console Workplace (V7R7.3.0.0) - Windows Internet Explorer
https://c250hmc07.ppd.pok.ibm.com/hmc/connects/mainuiFrameset.jsp
Hardware Management Console
com.ibm.hwmca.fw.servlet.connect.ConnectTask | Manage Serviceable
Welcome
Systems Management
Servers
Server-9125-F2C-SN02D769
Custom Groups
System Plans
HMC Management
Service Management
Updates
Systems Management
Select
Name
Server-912
Tasks: Servers
Connections
c250hmc07: Manage Serviceable Events - Windows Internet Explorer
https://c250hmc07.ppd.pok.ibm.com/hmc/wcl/T2729#tableTop_53be53be
Certificate Error
Manage Serviceable Events - Serviceable Event Overview
Selected
View Details
Repair
Call Home
Manage Problem Data
Close Event
Full table view
--- Select Action ---
View Search Criteria
Cancel
Help

t… …le events that match your selection criteria. Each event is grouped with all errors …t event. Use the menu bar above to perform actions on the serviceable event.

Select	Problem # ^	PMH # ^	Reference code ^	Status ^	Last reported time ^	Failing MTMS ^
□	18		B15A3305	Open	Oct 22, 2011 3:14:21 PM	9125-F2C/02D7695
☑	30		14020083	Open	Oct 22, 2011 9:30:35 AM	9458-100/BPCF010
□	28		14020082	Open	Oct 21, 2011 10:55:21 AM	9458-100/BPCF010
□	10		14022C52	Open	Oct 18, 2011 4:28:59 PM	9458-100/BPCF010
□	31		EA880936	Open	Oct 14, 2011 6:39:41 PM	7042-CR5/KQZAACT
□	11		B1812A80	Open	Oct 14, 2011 6:36:31 PM	9125-F2C/02D7695
□	9		11008711	Open	Oct 14, 2011 6:23:20 PM	9125-F2C/02D7695
□	5		B1754202	Open	Oct 14, 2011 6:20:40 PM	9125-F2C/02D7695
□	29		EA880936	Open	Oct 12, 2011 5:25:45 PM	7042-CR5/KQZAACT
□	27		B700F103	Open	Oct 11, 2011 7:47:17 PM	9125-F2C/02D7695
□	26		EA880936	Open	Oct 11, 2011 5:33:40 PM	7042-CR5/KQZAACT
□	25		A7001151	Open	Oct 11, 2011 5:33:27 PM	9125-F2C/02D7695

Total: 31 Filtered: 31 Selected: 1

Figure A-1 Access to the manage serviceable events on the HMC

The data that is gathered automatically includes a `IQYYLOG.log` that includes all of the saved events. IBM support reviews the entries and identifies the +PEL events that show the recorded reference code and other error details, including the FRU data.

Example A-1 on page 315 shows the detailed data.

Example A-1 Serviceable event detailed data from IQYYLOG

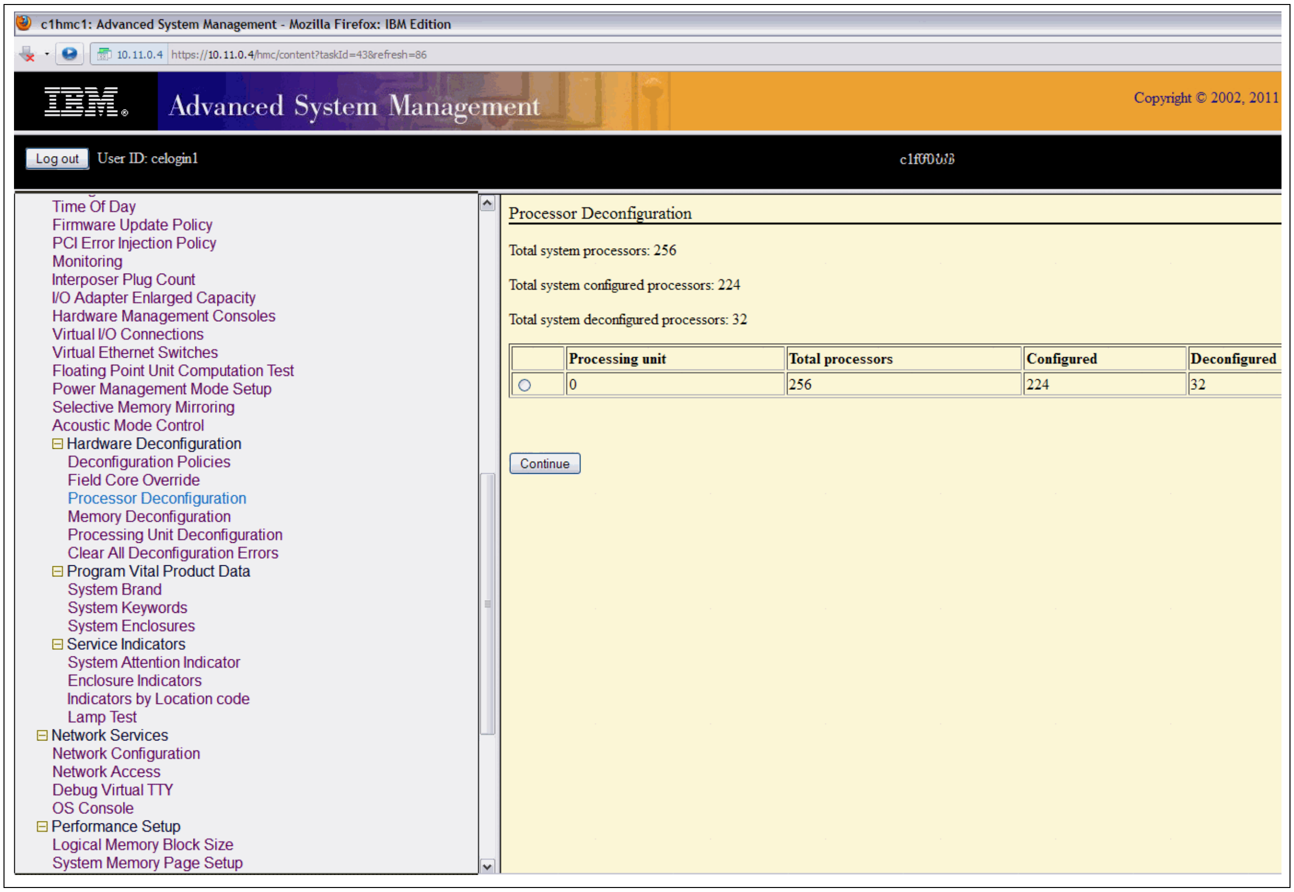
```
|------------------------------------------------------------------------------|
    |                       Platform Event Log - 0x50009190                           |
    |-------------------------------------------------------------------------------|
    |                            Private Header                                     |
    |-------------------------------------------------------------------------------|
    | Section Version          : 1                                                  |
    | Sub-section type         : 0                                                  |
    | Created by               : hfug                                               |
    | Created at               : 10/11/2011 14:47:21                                |
    | Committed at             : 10/11/2011 14:47:27                                |
    | Creator Subsystem        : FipS Error Logger                                  |
    | CSSVER                   :                                                    |
    | Platform Log Id          : 0x50009191                                         |
    | Entry Id                 : 0x50009190                                         |
    | Total Log Size           : 2624                                               |
    |-------------------------------------------------------------------------------|
    |                             User Header                                       |
    |-------------------------------------------------------------------------------|
    | Section Version          : 1                                                  |
    | Sub-section type         : 0                                                  |
    | Log Committed by         : hfug                                               |
    | Subsystem                : Not Applicable                                     |
    | Event Scope              : Single Platform                                    |
    | Event Severity           : Predictive Error                                   |
    | Event Type               : Not Applicable                                     |
    | Return Code              : 0x0000B2DF                                         |
    | Action Flags             : Report to Operating System                         |
    |                          : Service Action Required                            |
    |                          : HMC Call Home                                      |
    |-------------------------------------------------------------------------------|
    |                        Primary System Reference Code                          |
    |-------------------------------------------------------------------------------|
    | Section Version          : 1                                                  |
    | Sub-section type         : 1                                                  |
    | Created by               : hfug                                               |
    | SRC Format               : 0xF0                                               |
    | SRC Version              : 0x02                                               |
    | Virtual Progress SRC     : False                                              |
    | I5/OS Service Event Bit  : False                                              |
    | Hypervisor Dump Initiated: False                                              |
    | Power Control Net Fault  : False                                              |
    | Source Service Processor : A                                                  |
    | System Backplane CCIN    : 0x2E00                                             |
    | Last Progress Code       : 0xC100925C                                         |
    | System IPL State         : 0x02                                               |
    | SP Dump Status           : 0x00                                               |
    | Platform IPL Mode        : 0x00                                               |
    | Platform Dump Status     : 0x00                                               |
    | Partition IPL Type       : 0x00                                               |
    | Partition Dump Status    : 0x00                                               |
    | Deconfigured             : False                                              |
    | Garded                   : True                                               |
    | Error Status Flags       : None declared                                      |
    | Clock State              : None declared                                      |
    | Error SRC Count          : 0xFF                                               |
    |                                                                               |
    | Valid Word Count         : 0x09                                               |
    | Module Id                : 0x3B                                               |
    | Reference Code           : B175B2DF                                           |
    | Hex Words 2 - 5          : 020000F0 2E003B10 C100925C 010000FF               |
    | Hex Words 6 - 9          : 00000000 00000007 81040187 50500000               |
    |                                                                              |
    |                               Callout Section                                |
    |                                                                              |
    | Additional Sections      : Disabled                                          |
    | Callout Count            : 5                                                 |
    |                                                                              |
    |                        Maintenance Procedure Required                        |
    | Priority                 : Mandatory, replace all with this type as a unit  |
    | Procedure Number         : FSPSP55                                           |
    | Description              : An error has been detected on a bus between two FR|
    |                          : Us.  The end-point FRUs have been called out, but |
    |                          : the source of the error could be the bus path betw|
    |                          : een the FRUs.                                     |
    |                                                                              |
    |                             Normal Hardware FRU                              |
    | Priority                 : Medium Priority A, replace these as a group       |
    | Location Code            : U78A9.001.9912345-P1-R10                          |
    | Part Number              : 41U9500                                           |
    | CCIN                     : 2E00                                              |
    | Serial Number            : YH10HA112345                                      |
    |                                                                              |
    |                             Normal Hardware FRU                              |
    | Priority                 : Medium Priority                                   |
    | Location Code            : U78A9.001.9912345-P1-R11                          |
    | Part Number              : 41U9500                                           |
    | CCIN                     : 2E00                                              |
    | Serial Number            : YH10HA112345                                      |
    |                                                                              |
    |                             Normal Hardware FRU                              |
    | Priority                 : Medium Priority                                   |
    | Location Code            : U78A9.001.9912345-P1                              |
    | Part Number              : 41U9500                                           |
    | CCIN                     : 2E00                                              |
    | Serial Number            : YH10HA112345                                      |
    |                                                                              |
    |                        Maintenance Procedure Required                        |
    | Priority                 : Lowest priority replacement                       |
    | Procedure Number         : FSPSP04                                           |
    | Description              : A problem has been detected in the Service Process|
    |                          : or firmware by the Service Processor.             |
    |                                                                              |
    |------------------------------------------------------------------------------|
    |                             Extended User Header                             |
    |------------------------------------------------------------------------------|
    | Section Version          : 1                                                 |
    | Sub-section type         : 0                                                 |
    | Created by               : errl                                              |
    | Reporting Machine Type   : 9125-F2C                                          |
    | Reporting Serial Number  : 0212345                                           |
    | FW Released Ver          : AS730_044                                         |
    | FW SubSys Version        : b0823b_1136.731                                   |
    | Common Ref Time          : 00/00/0000 00:00:00                               |
    | Symptom Id Len           : 76                                                |
    | Symptom Id               : B181B2DF_020000F02E003B10C100925C000000FF000000000|
    |                          : 00000078104018750500000                           |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version         : 2                                                 |
    | Sub-section type        : 4                                                 |
    | Created by              : errl                                              |
    | PID                     : 1635                                              |
    | Process Name            : /opt/fips/bin/cecserver                           |
    | Driver Name             : fips731/b0823b_1136.731                           |
    | FSP Role                : Primary                                           |
    | Redundancy Policy       : Enabled                                           |
    | Sibling State           : Functional                                        |
    | State Manager State     : [SMGR_IPL] - IPL state - transitory               |
    | FSP IPL Type            : [SMGR_PWR_ON_RST] - Power on reset (AC power)     |
    | CEC IPL Type            : [SMGR_HMC_PWR_ON] - HMC init'd power on           |
    |------------------------------------------------------------------------------|
    |                      Machine Type/Model & Serial Number                      |
    |------------------------------------------------------------------------------|
    | Section Version         : 1                                                 |
    | Sub-section type        : 0                                                 |
    | Created by              : errl                                              |
    | Machine Type Model      : 9125-F2C                                          |
    | Serial Number           : 02CA7B6                                           |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version         : 0                                                 |
    | Sub-section type        : 0                                                 |
    | Created by              : hutl                                              |
    | Power State             : on                                                |
    | IPL Steps Completed     : 0x74420027 0xF9F00000 0x0A300904 0x00800000       |
    | SMGR CEC IPL Type       : SMGR_HMC_PWR_ON (0x00000002)                      |
    | SMGR FSP IPL Type       : SMGR_PWR_ON_RST (0x00000801)                      |
    | SMGR Curr State         : SMGR_IPL                                          |
    | SMGR Curr State Timestamp: 10/11/2011 14:42:36                              |
    | SMGR Next State         : SMGR_HOST_STARTED                                 |
    | SMGR Next State Timestamp: 10/11/2011 14:42:36                              |
    | IPLP Diagnostic Level   : IPLP_DIAGNOSTICS_NORMAL                           |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version         : 0                                                 |
    | Sub-section type        : 0                                                 |
    | Created by              : hutl                                              |
    |                                                                              |
    |   00000000     FED8200D  00000020  00036B00  00036330     .. .... ..k...c0   |
    |   00000010     00000EFC  19111B1B  02020202  00030D40     ...............@   |
    |   00000020     331B1B03  02020C02  0C040404  04040404     3...............   |
    |   00000030     04040404  04040404  04040404  04040404     ................   |
    |   00000040     04040404  04040404  0487BCC8  01000000     ................   |
    |   00000050     106BC018  10099428  00000000  00000000     .k.....(........   |
    |   00000060     00000000  0F5E1898  00040404  04040404     .....^..........   |
    |   00000070     04040404  04040404  04040404  04040404     ................   |
    |   00000080     04040404  04040404  0487C078  10878588     ...........x....   |
    |   00000090     0A037801  00000000  9C342B03  00000081     ..x......4+.....   |
    |   000000A0     1087CC90  10878B98  81181818  18181818     ................   |
    |   000000B0     18181818  18181818  18181818  18181818     ................   |
    |   000000C0     18181818  18181818  18F5A020  00000000     ........... ....   |
    |   000000D0     00000000  10884974  00000000  00000000     ......It........   |
    |   000000E0     00000000  00000000  00181818  18181818     ................   |
    |   000000F0     18181818  18181818  18181818  18181818     ................   |
    |   00000100     18181818  18181818  18000000  0F5E1898     .............^..   |
    |   00000110     00000000  00000000  00000120  00000020     ........... ...    |
    |   00000120     10878C18  10878588  10020202  02020202      ................   |
    |   00000130     02020202  02020202  02020202  02020202      ................   |
    |   00000140     02020202  02020202  0202000F  1017CB10      ................   |
    |   00000150     1017E7E0  00000000  00000003  00000040      ...............@   |
    |   00000160     0C030008  002001C1  0E000000  00000000      ..... ..........   |
    |   00000170     00000000  00000000  00000000  00000000      ................   |
    |   00000180     00000000  00000000  00002201  01004793      .........."...G.   |
    |   00000190     00000000  36005000  00000000  00000000      ....6.P.........   |
    |   000001A0     00000000  0F5E1898  00000000  00000000      .....^..........   |
    |   000001B0     00000000  00000000  00000000  00000000      ................   |
    |   000001C0     00000000  00000000  00900518  B5900518      ................   |
    |   000001D0     00000007  000000A1  1088BBF8  1087C7C8      ................   |
    |   000001E0     8102000F  1017CB10  10010101  01010101      ................   |
    |   000001F0     01010101  01010101  01010101  01010101      ................   |
    |   00000200     01010101  01010101  01000000  108784C4      ................   |
    |   00000210     00000000  00000000  00000000  00000000      ................   |
    |   00000220     00000000  01000000  10010101  01010101      ................   |
    |   00000230     01010101  01010101  01010101  01010101      ................   |
    |   00000240     01010101  01010101  01000000  00000000      ................   |
    |   00000250     00000080  00000020  1087C1E0  1087C1E8      ....... ........   |
    |   00000260     00000000  00000000  04033001                ..........0.       |
    |                                                                              |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version          : 0                                                 |
    | Sub-section type         : 0                                                 |
    | Created by               : hutl                                              |
    | HOM UnitId               : HOM_CECCAGE0-HOM_FABRIC3 (0x81030003)             |
    | SCOM Address             : 0x4011460                                         |
    | SCOM Data                : 0x04000000 00000000                               |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version          : 0                                                 |
    | Sub-section type         : 0                                                 |
    | Created by               : hutl                                              |
    | HOM UnitId               : HOM_CECCAGE0-HOM_FABRIC3 (0x81030003)             |
    | SCOM Address             : 0x4011461                                         |
    | SCOM Data                : 0x00003001 00000000                               |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    | Section Version          : 0                                                 |
    | Sub-section type         : 0                                                 |
    | Created by               : hutl                                              |
    | HOM UnitId               : HOM_CECCAGE0-HOM_FABRIC3 (0x81030003)             |
    | SCOM Address             : 0x4011462                                         |
    | SCOM Data                : 0x80400000 00000000                               |
    |------------------------------------------------------------------------------|
    |                              User Defined Data                               |
    |------------------------------------------------------------------------------|
    |------------------------------------------------------------------------------|
    |                          Firmware Error Description                          |
    |------------------------------------------------------------------------------|
    | Section Version          : 2                                                 |
    | Sub-section type         : 1                                                 |
    | Created by               : hutl                                              |
    | File Identifier          : 0x0013                                            |
    | Code Location            : 0x002B                                            |
    | Return Code             : 0x0000B70C                                       |
    | Object Identifier       : 0x00000000                                       |
    |----------------------------------------------------------------------------|
```

The reference code in Example A-1 on page 315 includes the following line:

```
Garded                          : True
```

In this example, the firmware guards and deconfigures the resource so that no other problems occur on that resource. In the Advanced System Management Interface (ASMI) of that specific node, it is possible to view the deconfigured resources. The xCAT command **`rinv NODENAME deconfig`** also is available to view the deconfigured resources.

Figure A-2 shows a view on the ASMI deconfiguration window for processors and memory. Because a Quad Chip Module (QCM) included dedicated memory that is associated with it, the memory is deconfigured as well.


c1hmc1: Advanced System Management - Mozilla Firefox: IBM Edition
10.11.0.4 https://10.11.0.4/hmc/content?taskId=43&refresh=86
IBM
Advanced System Management
Copyright © 2002, 2011
Log out
User ID: celogin1
c1f0003
Time Of Day
Firmware Update Policy
PCI Error Injection Policy
Monitoring
Interposer Plug Count
I/O Adapter Enlarged Capacity
Hardware Management Consoles
Virtual I/O Connections
Virtual Ethernet Switches
Floating Point Unit Computation Test
Power Management Mode Setup
Selective Memory Mirroring
Acoustic Mode Control
⊟ Hardware Deconfiguration
Deconfiguration Policies
Field Core Override
Processor Deconfiguration
Memory Deconfiguration
Processing Unit Deconfiguration
Clear All Deconfiguration Errors
⊟ Program Vital Product Data
System Brand
System Keywords
System Enclosures
⊟ Service Indicators
System Attention Indicator
Enclosure Indicators
Indicators by Location code
Lamp Test
⊟ Network Services
Network Configuration
Network Access
Debug Virtual TTY
OS Console
⊟ Performance Setup
Logical Memory Block Size
System Memory Page Setup
Continue

Processor Deconfiguration

Total system processors: 256

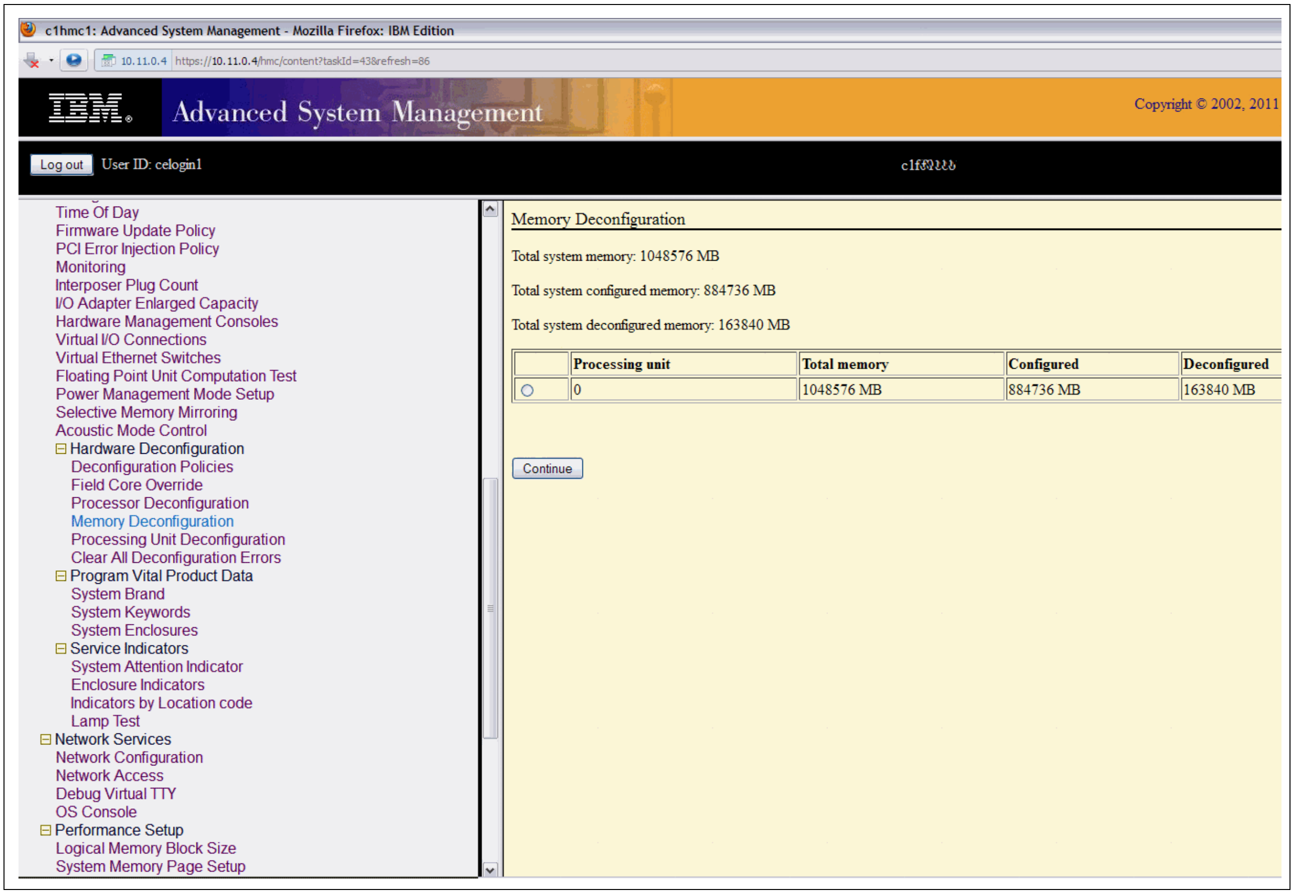
Total system configured processors: 224

Total system deconfigured processors: 32

	Processing unit	Total processors	Configured	Deconfigured
○	0	256	224	32

Figure A-2 ASMI processor deconfiguration window

Figure A-3 on page 320 shows the memory deconfiguration window.


c1hmc1: Advanced System Management - Mozilla Firefox: IBM Edition
10.11.0.4
https://10.11.0.4/hmc/content?taskId=43&refresh=86
IBM.
Advanced System Management
Copyright © 2002, 2011
Log out
User ID: celogin1
Time Of Day
Firmware Update Policy
PCI Error Injection Policy
Monitoring
Interposer Plug Count
I/O Adapter Enlarged Capacity
Hardware Management Consoles
Virtual I/O Connections
Virtual Ethernet Switches
Floating Point Unit Computation Test
Power Management Mode Setup
Selective Memory Mirroring
Acoustic Mode Control
⊟ Hardware Deconfiguration
Deconfiguration Policies
Field Core Override
Processor Deconfiguration
Memory Deconfiguration
Processing Unit Deconfiguration
Clear All Deconfiguration Errors
⊟ Program Vital Product Data
System Brand
System Keywords
System Enclosures
⊟ Service Indicators
System Attention Indicator
Enclosure Indicators
Indicators by Location code
Lamp Test
⊟ Network Services
Network Configuration
Network Access
Debug Virtual TTY
OS Console
⊟ Performance Setup
Logical Memory Block Size
System Memory Page Setup

Memory Deconfiguration

Total system memory: 1048576 MB

Total system configured memory: 884736 MB

Total system deconfigured memory: 163840 MB

	Processing unit	Total memory	Configured	Deconfigured
○	0	1048576 MB	884736 MB	163840 MB

Figure A-3 ASMI memory deconfiguration window

When you verify any deconfigured resources, check that after the reference code, you see the section called: *Maintenance Procedure Required*.

The first procedure listed is the FSPSP55. For more information about this procedure, see IBM Power Systems Hardware Information Center at this website:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp

The fastest way to find the FSPSP55 isolation procedure is to search for it in the Information Center. The procedure guides you to: Power Systems Information → POWER7 Systems → 9125-F2C → Troubleshooting, service and support → Isolation Procedures → Service Processor Isolation Procedures → FSPSP55.

In this example, the requested Word 6 Data shows 000000000 and the procedure points to the FRU callouts in the event. The first FRU includes the following location code:

```
U78A9.001.9912345-P1-R10
```

For more information about the FRU Location to A+ resource, see Table 5-4 on page 304.

Table 5-4 on page 304 shows U*-P1-R(9-40) is a QCM. In our example, R10, and in Table 5-5 on page 305 (QCM to OCTANT MAP), you see that U*-P1-R(9-12) is in octant 0. Figure A-4 on page 321 shows a view of the board layout for the octants.

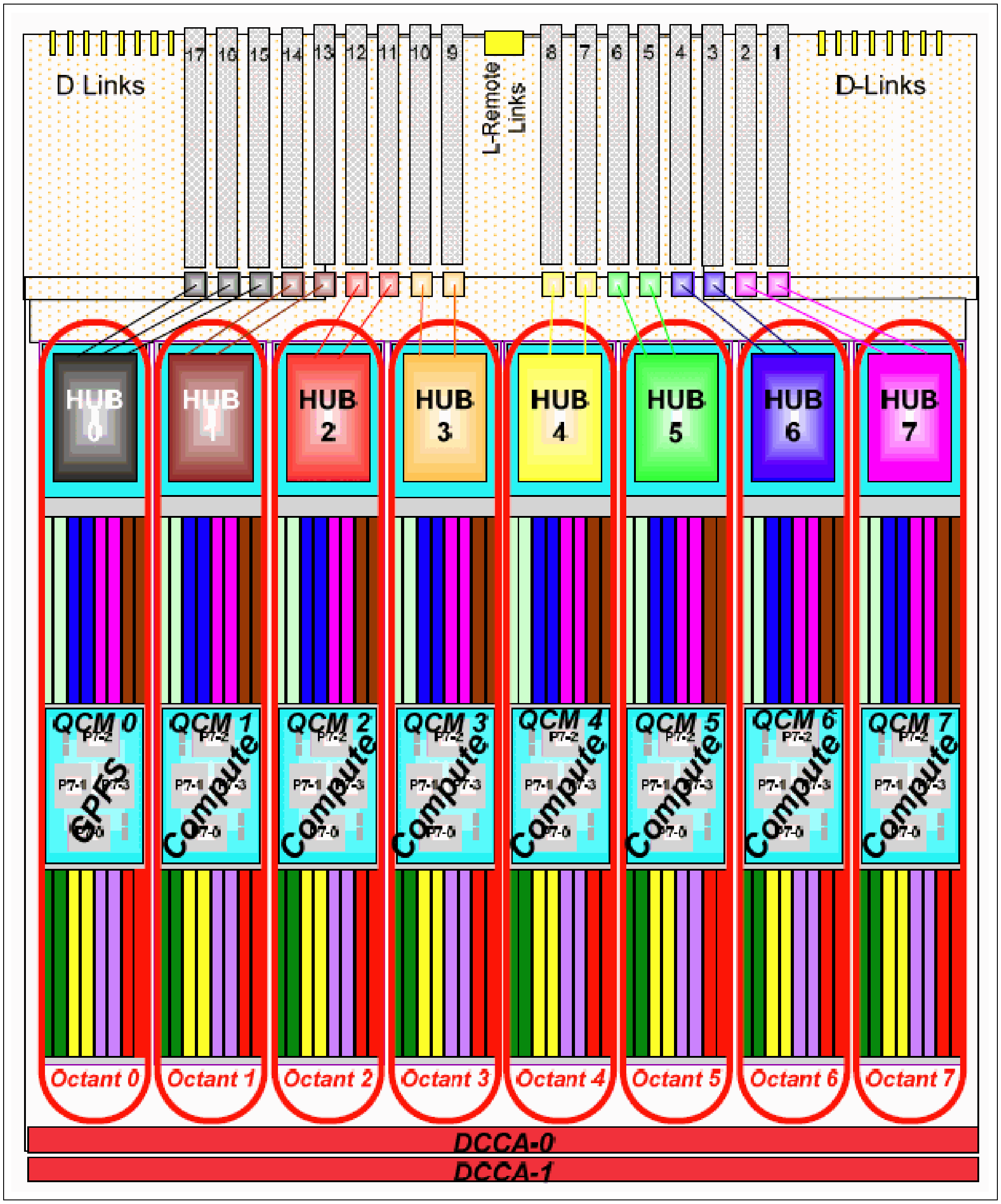


Figure A-4 Board layout

In this example, the Availability plus procedure is used to remove this resource from the current compute or non-compute node configuration. The resource is moved into the failed resources group.

For more information about the A+ procedures, see 5.4, “Power 775 Availability Plus” on page 297, or see this website:

http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Cluster_Recovery#P775_Fail_in_Place_.28FIP.29

The complete service path is in the Cluster Service Guide at this website:

https://www.ibm.com/developerworks/wikis/download/attachments/162267485/p775_service_guide.pdf?version=1

B

Command outputs

This appendix provides long command output examples.

The following topics are described in this appendix:

- General Parallel File System native RAID
- DB2

General Parallel File System native RAID

The following section provides examples for the General Parallel File System (GPFS) native RAID. An output from the **mmgetpdisktopology** command is shown in Example B-1.

Example B-1 mmgetpdisktopology output example

```
hdisk1,hdisk97:0:/dev/rhdisk1,/dev/rhdisk97:140A0C0D4EA076F0|c101d1|000DE22BOT|140
A0C0D4EA07897|DA1:naa.5000C5001CE322D3:U78AD.001.000DE22-P1-C101-D1,U78AD.001.000D
E22-P1-C101-D1:01-00-00,03-00-00:/dev/hdisk1,/dev/hdisk97:IBM-ESXS:ST9300603SS
F:B536:3SE1EDVQ00009024Z1E7:42D0628:299999690752:78900111222331121,789001112223311
11:500507603e013643.500507603e0136c3:000DE22:P1-C76/P1-C77:P1-C101-D1
2SS6:5000c5001ce322d1.5000c5001ce322d2::
hdisk2,hdisk98:0:/dev/rhdisk2,/dev/rhdisk98:140A0C0D4EA07730|c101d2|000DE22BOT|140
A0C0D4EA07897|DA2:naa.5000C5001CE352E3:U78AD.001.000DE22-P1-C101-D2,U78AD.001.000D
E22-P1-C101-D2:01-00-00,03-00-00:/dev/hdisk2,/dev/hdisk98:IBM-ESXS:ST9300603SS
F:B536:3SE1DLV200009025TSY3:42D0628:299999690752:78900111222331121,789001112223311
11:500507603e013643.500507603e0136c3:000DE22:P1-C76/P1-C77:P1-C101-D2
2SS6:5000c5001ce352e1.5000c5001ce352e2::
hdisk3,hdisk99:0:/dev/rhdisk3,/dev/rhdisk99:140A0C0D4EA0775E|c101d3|000DE22BOT|140
A0C0D4EA07897|DA3:naa.5000C5001CE45DC7:U78AD.001.000DE22-P1-C101-D3,U78AD.001.000D
E22-P1-C101-D3:01-00-00,03-00-00:/dev/hdisk3,/dev/hdisk99:IBM-ESXS:ST9300603SS
F:B536:3SE1EXJ200009025QWSH:42D0628:299999690752:78900111222331121,789001112223311
11:500507603e013643.500507603e0136c3:000DE22:P1-C76/P1-C77:P1-C101-D3
2SS6:5000c5001ce45dc5.5000c5001ce45dc6::
[...]
hdisk667,hdisk763:0:/dev/rhdisk667,/dev/rhdisk763:140A0C0D4EA0773E|c056d2|000DE22B
OT|140A0C0D4EA07897|DA2:naa.5000C5001CE339BF:U78AD.001.000DE22-P1-C56-D2,U78AD.001
.000DE22-P1-C56-D2:02-00-00,04-00-00:/dev/hdisk667,/dev/hdisk763:IBM-ESXS:ST930060
3SS
F:B536:3SE195GJ00009025UAHR:42D0628:299999690752:78900111222331101,789001112223319
1:500507603e013343.500507603e0133c3:000DE22:P1-C28/P1-C29:P1-C56-D2
2SS6:5000c5001ce339bd.5000c5001ce339be::
hdisk668,hdisk764:0:/dev/rhdisk668,/dev/rhdisk764:140A0C0D4EA07788|c056d3|000DE22B
OT|140A0C0D4EA07897|DA3:naa.5000C5001CE3B803:U78AD.001.000DE22-P1-C56-D3,U78AD.001
.000DE22-P1-C56-D3:02-00-00,04-00-00:/dev/hdisk668,/dev/hdisk764:IBM-ESXS:ST930060
3SS
F:B536:3SE1EQ9900009025TWYG:42D0628:299999690752:78900111222331101,789001112223319
1:500507603e013343.500507603e0133c3:000DE22:P1-C28/P1-C29:P1-C56-D3
2SS6:5000c5001ce3b801.5000c5001ce3b802::
hdisk669,hdisk765:0:/dev/rhdisk669,/dev/rhdisk765:140A0C0D4EA077AA|c056d4|000DE22B
OT|140A0C0D4EA07897|DA4:naa.5000C5001CE351AF:U78AD.001.000DE22-P1-C56-D4,U78AD.001
.000DE22-P1-C56-D4:02-00-00,04-00-00:/dev/hdisk669,/dev/hdisk765:IBM-ESXS:ST930060
3SS
F:B536:3SE1949400009025TT6D:42D0628:299999690752:78900111222331101,789001112223319
1:500507603e013343.500507603e0133c3:000DE22:P1-C28/P1-C29:P1-C56-D4
2SS6:5000c5001ce351ad.5000c5001ce351ae::
host:-1:c250f10c12ap13-hf0:::::::::::::::::
mpt2sas0:31:/dev/mpt2sas0::78900111222331121:U78A9.001.1122233-P1-C12-T1:01-00:::7
637:1005480000:YH10KU11N428:74Y0500:::::::
mpt2sas1:31:/dev/mpt2sas1::78900111222331111:U78A9.001.1122233-P1-C11-T1:03-00:::7
637:1005480000:YH10KU11N277:74Y0500:::::::
mpt2sas2:31:/dev/mpt2sas2::78900111222331101:U78A9.001.1122233-P1-C10-T1:02-00:::7
637:1005480000:YH10KU11N830:74Y0500:::::::
```

```
mpt2sas3:31:/dev/mpt2sas3::7890011122233191:U78A9.001.1122233-P1-C9-T1:04-00:::763
7:1005480000:YH10KU11N057:74Y0500::::::
ses16:13:/dev/ses16::naa.500507603E013630:U78AD.001.000DE22-P1-C76:01-00-00::IBM:7
8AD-001:0154:YH10UE13G023:74Y02390::78900111222331121::000DE22:P1-C76::
ses17:13:/dev/ses17::naa.500507603E013670:U78AD.001.000DE22-P1-C76:01-00-00::IBM:7
8AD-001:0154:YH10UE13G023:74Y02390::78900111222331121::000DE22:P1-C76::
ses18:13:/dev/ses18::naa.500507603E013730:U78AD.001.000DE22-P1-C84:01-00-00::IBM:7
8AD-001:0154:YH10UE13G010:74Y02390::78900111222331121::000DE22:P1-C84::
[...]
ses45:13:/dev/ses45::naa.500507603E0132F0:U78AD.001.000DE22-P1-C21:04-00-00::IBM:7
8AD-001:0154:YH10UE13P019:74Y02390::7890011122233191::000DE22:P1-C21::
ses46:13:/dev/ses46::naa.500507603E0133B0:U78AD.001.000DE22-P1-C29:04-00-00::IBM:7
8AD-001:0154:YH10UE13G022:74Y02380::7890011122233191::000DE22:P1-C29::
ses47:13:/dev/ses47::naa.500507603E0133F0:U78AD.001.000DE22-P1-C29:04-00-00::IBM:7
8AD-001:0154:YH10UE13G022:74Y02380::7890011122233191::000DE22:P1-C29::
```

DB2

An output from the command **db2 get database configuration for <DB2_instance>** is shown in Example B-2.

Example B-2 Command to check details about DB2 database instance

```
# db2 get database configuration for xcatdb

       Database Configuration for Database xcatdb

 Database configuration release level                    = 0x0d00
 Database release level                                  = 0x0d00

 Database territory                                      = US
 Database code page                                      = 1208
 Database code set                                       = UTF-8
 Database country/region code                            = 1
 Database collating sequence                             = SYSTEM_819
 Alternate collating sequence              (ALT_COLLATE) =
 Number compatibility                                    = OFF
 Varchar2 compatibility                                  = OFF
 Date compatibility                                      = OFF
 Database page size                                      = 4096

 Dynamic SQL Query management           (DYN_QUERY_MGMT) = DISABLE

 Statement concentrator                      (STMT_CONC) = OFF

 Discovery support for this database       (DISCOVER_DB) = ENABLE

 Restrict access                                         = NO
 Default query optimization class         (DFT_QUERYOPT) = 5
 Degree of parallelism                      (DFT_DEGREE) = ANY
 Continue upon arithmetic exceptions   (DFT_SQLMATHWARN) = NO
 Default refresh age                   (DFT_REFRESH_AGE) = 0
 Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained     (NUM_FREQVALUES) = 10
 Number of quantiles retained            (NUM_QUANTILES) = 20

 Decimal floating point rounding mode  (DECFLT_ROUNDING) = ROUND_HALF_EVEN

 Backup pending                                          = NO

 All committed transactions have been written to disk    = NO
 Rollforward pending                                     = NO
 Restore pending                                         = NO

 Multi-page file allocation enabled                      = YES

 Log retain for recovery status                          = NO
 User exit for logging status                            = NO

 Self tuning memory                    (SELF_TUNING_MEM) = ON
 Size of database shared memory (4KB)  (DATABASE_MEMORY) = AUTOMATIC(320460)
 Database memory threshold               (DB_MEM_THRESH) = 10
 Max storage for lock list (4KB)              (LOCKLIST) = AUTOMATIC(56672)
 Percent. of lock lists per application       (MAXLOCKS) = AUTOMATIC(97)
 Package cache size (4KB)                   (PCKCACHESZ) = AUTOMATIC(83817)
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = AUTOMATIC(571)
 Sort list heap (4KB)                         (SORTHEAP) = AUTOMATIC(114)

 Database heap (4KB)                            (DBHEAP) = AUTOMATIC(2492)
 Catalog cache size (4KB)              (CATALOGCACHE_SZ) = 50
 Log buffer size (4KB)                        (LOGBUFSZ) = 98
 Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 50193
 Buffer pool size (pages)                     (BUFFPAGE) = 1000
 SQL statement heap (4KB)                     (STMTHEAP) = AUTOMATIC(8192)
 Default application heap (4KB)             (APPLHEAPSZ) = AUTOMATIC(256)
 Application Memory Size (4KB)             (APPL_MEMORY) = AUTOMATIC(40000)
 Statistics heap size (4KB)               (STAT_HEAP_SZ) = AUTOMATIC(4384)

 Interval for checking deadlock (ms)         (DLCHKTIME) = 10000
 Lock timeout (sec)                        (LOCKTIMEOUT) = -1

 Changed pages threshold                (CHNGPGS_THRESH) = 80
 Number of asynchronous page cleaners   (NUM_IOCLEANERS) = AUTOMATIC(31)
 Number of I/O servers                   (NUM_IOSERVERS) = AUTOMATIC(6)
 Index sort flag                             (INDEXSORT) = YES
 Sequential detect flag                      (SEQDETECT) = YES
 Default prefetch size (pages)         (DFT_PREFETCH_SZ) = AUTOMATIC

 Track modified pages                         (TRACKMOD) = OFF

 Default number of containers                            = 1
 Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32

 Max number of active applications            (MAXAPPLS) = AUTOMATIC(94)
 Average number of active applications       (AVG_APPLS) = AUTOMATIC(1)
 Max DB files open per application            (MAXFILOP) = 61440

 Log file size (4KB)                         (LOGFILSIZ) = 40048
 Number of primary log files                (LOGPRIMARY) = 10
```

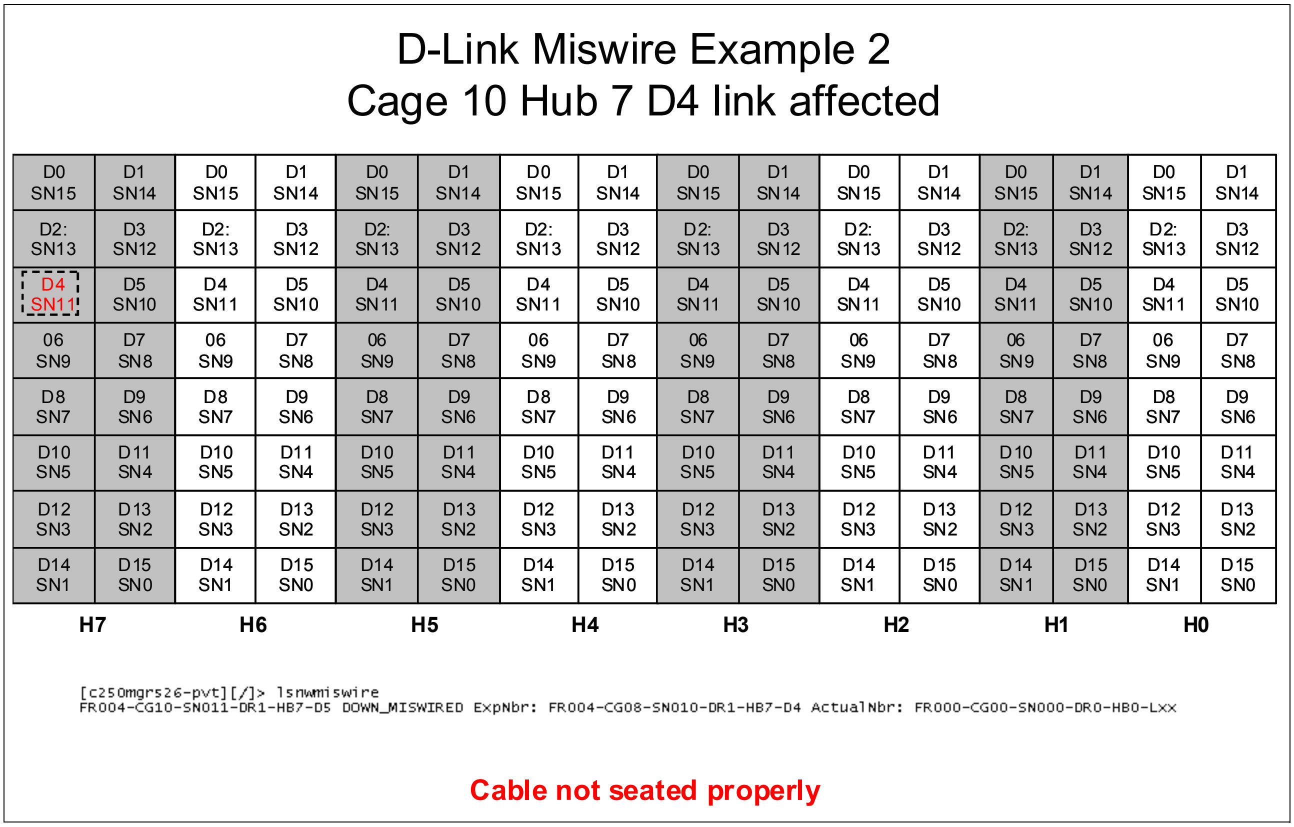
```
 Number of secondary log files               (LOGSECOND) = 20
 Changed path to log files                  (NEWLOGPATH) =
 Path to log files                                       =
/db2database/db2/xcatdb/NODE0000/SQL00001/SQLOGDIR/
 Overflow log path                     (OVERFLOWLOGPATH) =
 Mirror log path                         (MIRRORLOGPATH) =
 First active log file                                   =
 Block log on disk full                (BLK_LOG_DSK_FUL) = NO
 Block non logged operations            (BLOCKNONLOGGED) = NO
 Percent max primary log space by transaction  (MAX_LOG) = 0
 Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

 Group commit count                          (MINCOMMIT) = 1
 Percent log file reclaimed before soft chckpt (SOFTMAX) = 120
 Log retain for recovery enabled             (LOGRETAIN) = OFF
 User exit for logging enabled                (USEREXIT) = OFF

 HADR database role                                      = STANDARD
 HADR local host name                  (HADR_LOCAL_HOST) =
 HADR local service name                (HADR_LOCAL_SVC) =
 HADR remote host name                (HADR_REMOTE_HOST) =
 HADR remote service name              (HADR_REMOTE_SVC) =
 HADR instance name of remote server  (HADR_REMOTE_INST) =
 HADR timeout value                       (HADR_TIMEOUT) = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = NEARSYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 0

 First log archive method                 (LOGARCHMETH1) = OFF
 Options for logarchmeth1                  (LOGARCHOPT1) =
 Second log archive method                (LOGARCHMETH2) = OFF
 Options for logarchmeth2                  (LOGARCHOPT2) =
 Failover log archive path                (FAILARCHPATH) =
 Number of log archive retries on error   (NUMARCHRETRY) = 5
 Log archive retry Delay (secs)         (ARCHRETRYDELAY) = 20
 Vendor options                              (VENDOROPT) =

 Auto restart enabled                      (AUTORESTART) = ON
 Index re-creation time and redo index build  (INDEXREC) = SYSTEM (RESTART)
 Log pages during index build            (LOGINDEXBUILD) = OFF
 Default number of loadrec sessions    (DFT_LOADREC_SES) = 1
 Number of database backups to retain   (NUM_DB_BACKUPS) = 12
 Recovery history retention (days)     (REC_HIS_RETENTN) = 366
 Auto deletion of recovery objects    (AUTO_DEL_REC_OBJ) = OFF

 TSM management class                    (TSM_MGMTCLASS) =
 TSM node name                            (TSM_NODENAME) =
 TSM owner                                   (TSM_OWNER) =
 TSM password                             (TSM_PASSWORD) =

 Automatic maintenance                      (AUTO_MAINT) = ON
   Automatic database backup            (AUTO_DB_BACKUP) = OFF
   Automatic table maintenance          (AUTO_TBL_MAINT) = ON
     Automatic runstats                  (AUTO_RUNSTATS) = ON
       Automatic statement statistics  (AUTO_STMT_STATS) = ON
     Automatic statistics profiling    (AUTO_STATS_PROF) = OFF
```

```
      Automatic profile updates         (AUTO_PROF_UPD) = OFF
    Automatic reorganization               (AUTO_REORG) = OFF

Auto-Revalidation                          (AUTO_REVAL) = DEFERRED
Currently Committed                        (CUR_COMMIT) = ON
CHAR output with DECIMAL input        (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations        (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes)     (WLM_COLLECT_INT) = 0
Monitor Collect Settings
Request metrics                       (MON_REQ_METRICS) = BASE
Activity metrics                      (MON_ACT_METRICS) = BASE
Object metrics                        (MON_OBJ_METRICS) = BASE
Unit of work events                      (MON_UOW_DATA) = NONE
Lock timeout events                   (MON_LOCKTIMEOUT) = NONE
Deadlock events                          (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events                         (MON_LOCKWAIT) = NONE
Lock wait event threshold               (MON_LW_THRESH) = 5000000
Number of package list entries         (MON_PKGLIST_SZ) = 32
Lock event notification level         (MON_LCK_MSG_LVL) = 1

SMTP Server                               (SMTP_SERVER) =
SQL conditional compilation flags         (SQL_CCFLAGS) =
Section actuals setting               (SECTION_ACTUALS) = NONE
Connect procedure                        (CONNECT_PROC) =
```

The `lsnwmiswire` command

The **`lsnwmiswire`** command lists miswired links. Miswires sample figure output that was captured during the residency is shown in the figures on the following pages.

Figure B-1 on page 329 shows the D-Link miswire example in cage 10, hub 7, D4 link affected.



D-Link Miswire Example 2
Cage 10 Hub 7 D4 link affected

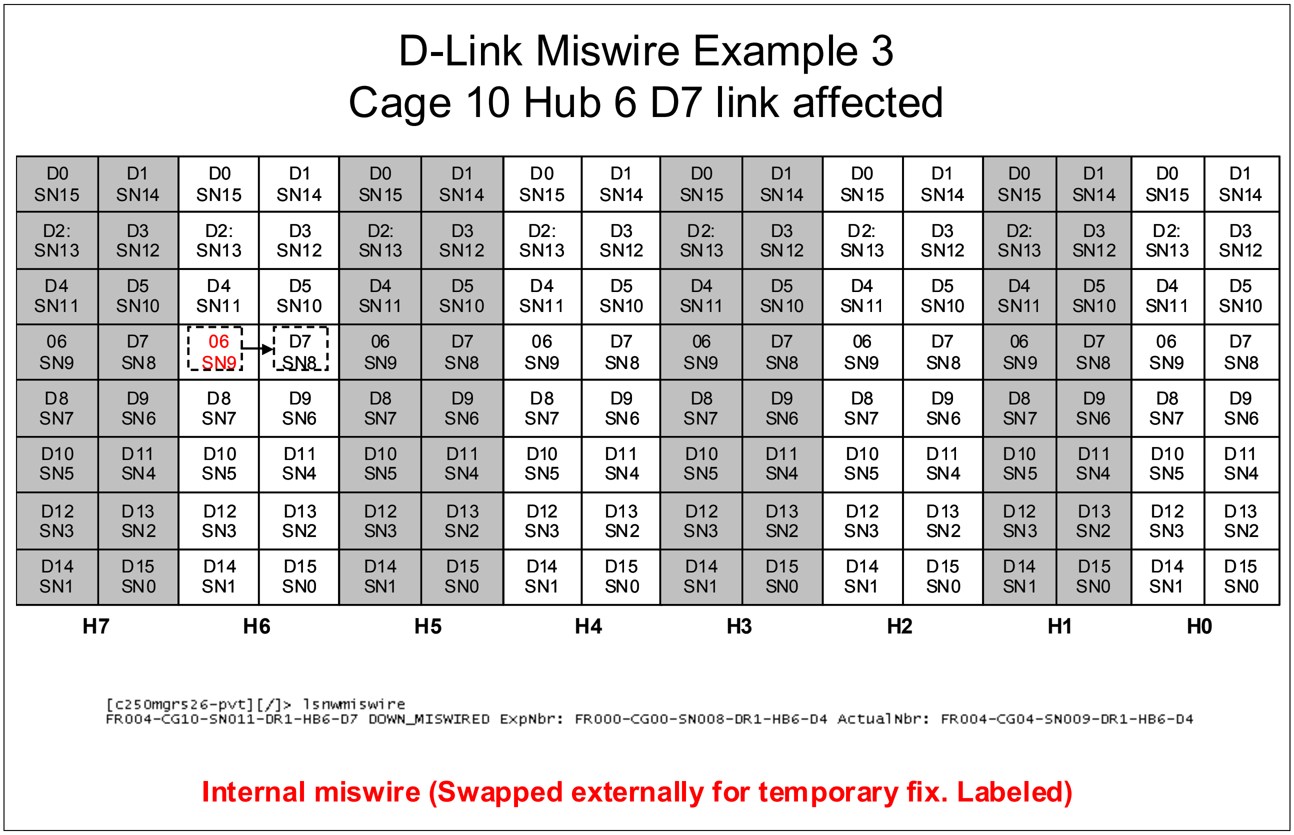
D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		**H6**		**H5**		**H4**		**H3**		**H2**		**H1**		**H0**	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG10-SN011-DR1-HB7-D5 DOWN_MISWIRED ExpNbr: FR004-CG08-SN010-DR1-HB7-D4 ActualNbr: FR000-CG00-SN000-DR0-HB0-Lxx
```

Cable not seated properly

Figure B-1 Miswire sample figure 2: Cable not seated properly

Figure B-2 on page 330 shows the D-Link miswire example in cage 10, hub 6, D7 link affected.



D-Link Miswire Example 3
Cage 10 Hub 6 D7 link affected

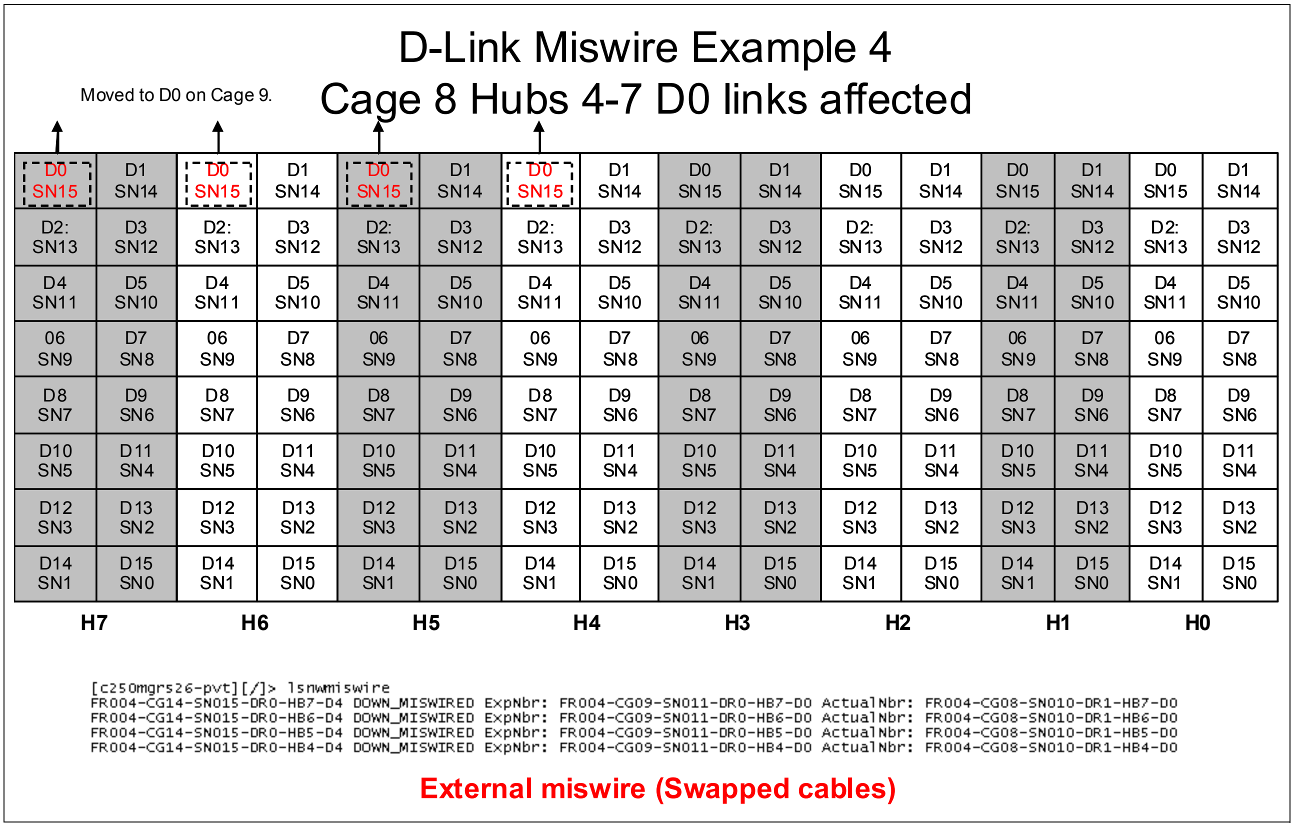
D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		**H6**		**H5**		**H4**		**H3**		**H2**		**H1**		**H0**	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG10-SN011-DR1-HB6-D7 DOWN_MISWIRED ExpNbr: FR000-CG00-SN008-DR1-HB6-D4 ActualNbr: FR004-CG04-SN009-DR1-HB6-D4
```

Internal miswire (Swapped externally for temporary fix. Labeled)

Figure B-2 Miswire sample figure 3 - Internal miswire swapped externally for temporary fix

Figure B-3 on page 331 shows the D-Link miswire example in cage 8, hubs 4-7, D0 links affected.



D-Link Miswire Example 4
Cage 8 Hubs 4-7 D0 links affected

Moved to D0 on Cage 9.

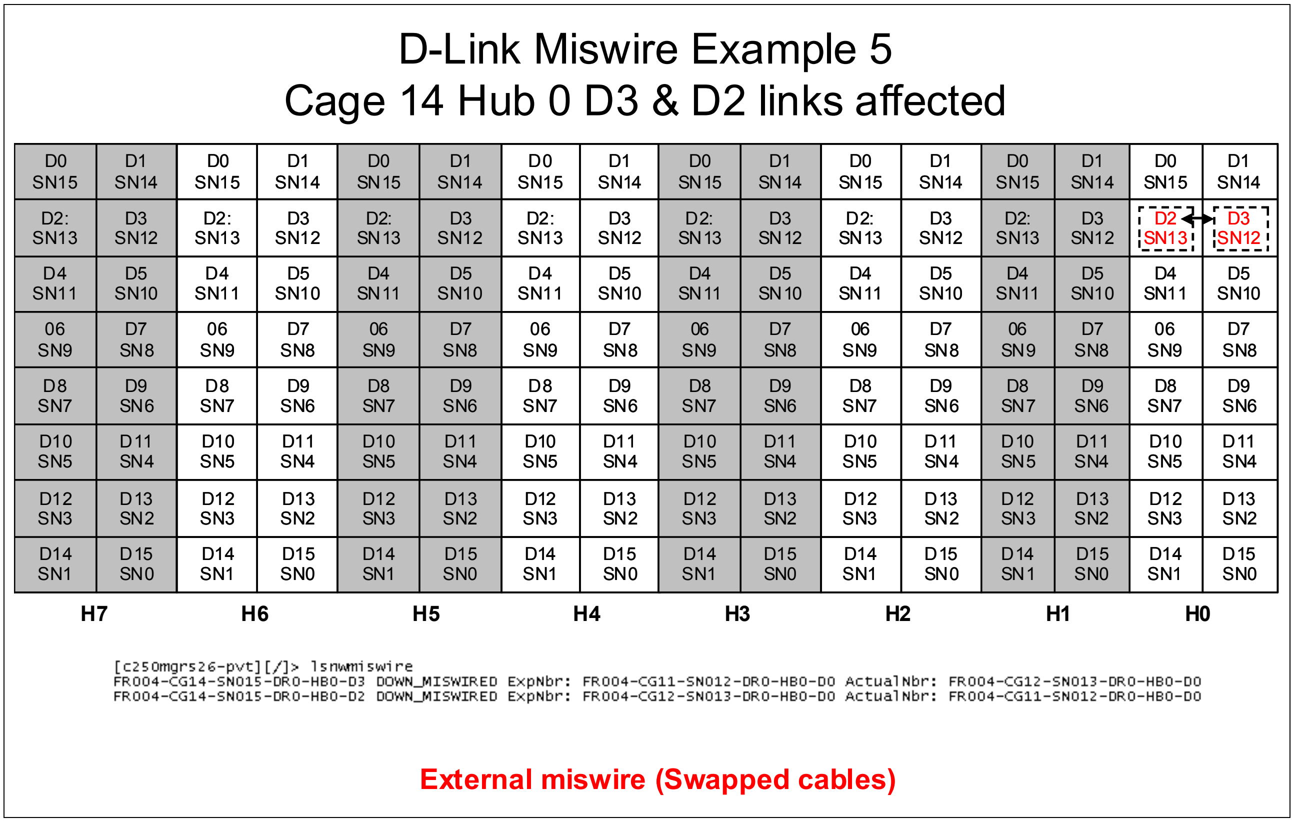
D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		**H6**		**H5**		**H4**		**H3**		**H2**		**H1**		**H0**	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG14-SN015-DR0-HB7-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB7-D0 ActualNbr: FR004-CG08-SN010-DR1-HB7-D0
FR004-CG14-SN015-DR0-HB6-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB6-D0 ActualNbr: FR004-CG08-SN010-DR1-HB6-D0
FR004-CG14-SN015-DR0-HB5-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB5-D0 ActualNbr: FR004-CG08-SN010-DR1-HB5-D0
FR004-CG14-SN015-DR0-HB4-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB4-D0 ActualNbr: FR004-CG08-SN010-DR1-HB4-D0
```

External miswire (Swapped cables)

Figure B-3 Miswire sample figure 4 - External miswire (swapped cables)

Figure B-4 on page 332 shows a D-Link miswire example on cage 14, hub 0, and D3 and D2 links affected.



D-Link Miswire Example 5
Cage 14 Hub 0 D3 & D2 links affected

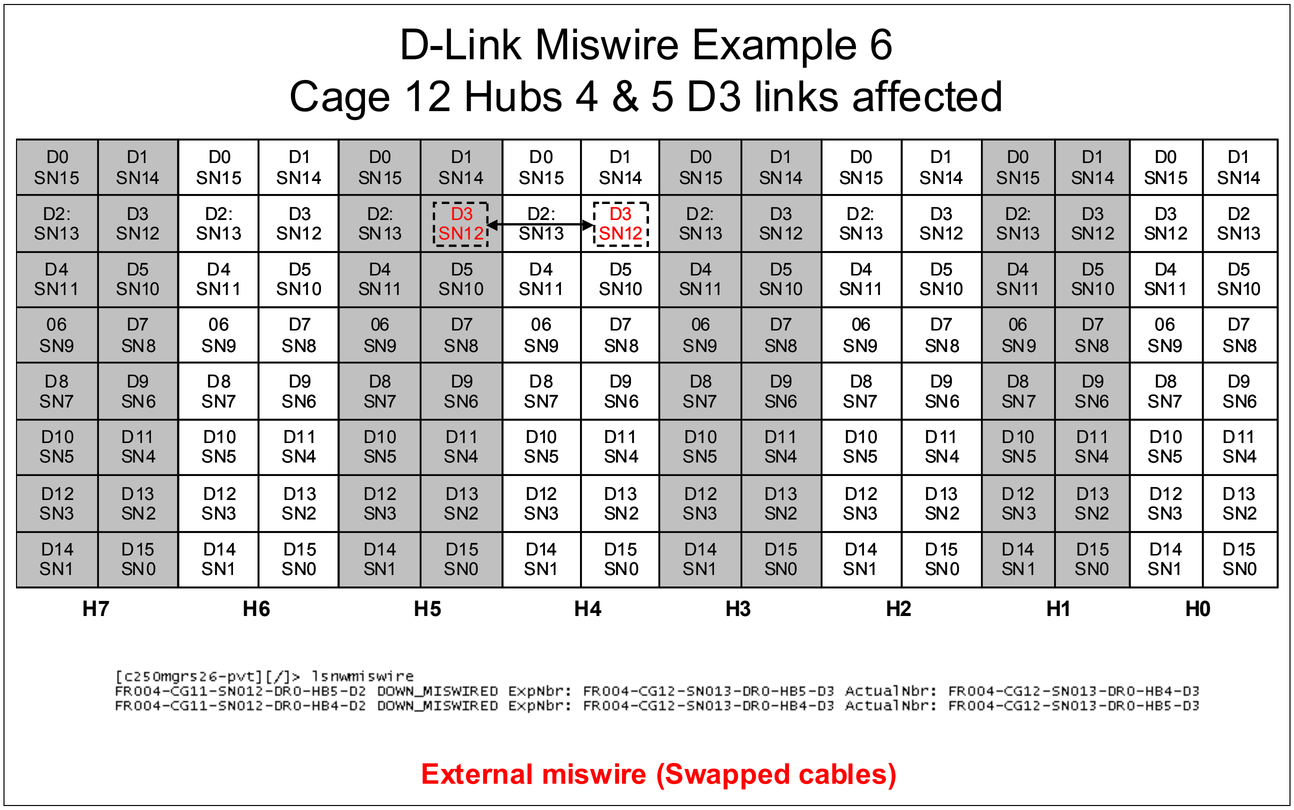
D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2 SN13	D3 SN12
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		H6		H5		H4		H3		H2		H1		H0	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG14-SN015-DR0-HB0-D3 DOWN_MISWIRED ExpNbr: FR004-CG11-SN012-DR0-HB0-D0 ActualNbr: FR004-CG12-SN013-DR0-HB0-D0
FR004-CG14-SN015-DR0-HB0-D2 DOWN_MISWIRED ExpNbr: FR004-CG12-SN013-DR0-HB0-D0 ActualNbr: FR004-CG11-SN012-DR0-HB0-D0
```

External miswire (Swapped cables)

Figure B-4 Miswire sample figure 5 - External miswire (swapped cables)

Figure B-5 on page 333 shows a D-Link miswire example on cage 12, hubs 4 and 5, and D3 links affected.



D-Link Miswire Example 6
Cage 12 Hubs 4 & 5 D3 links affected

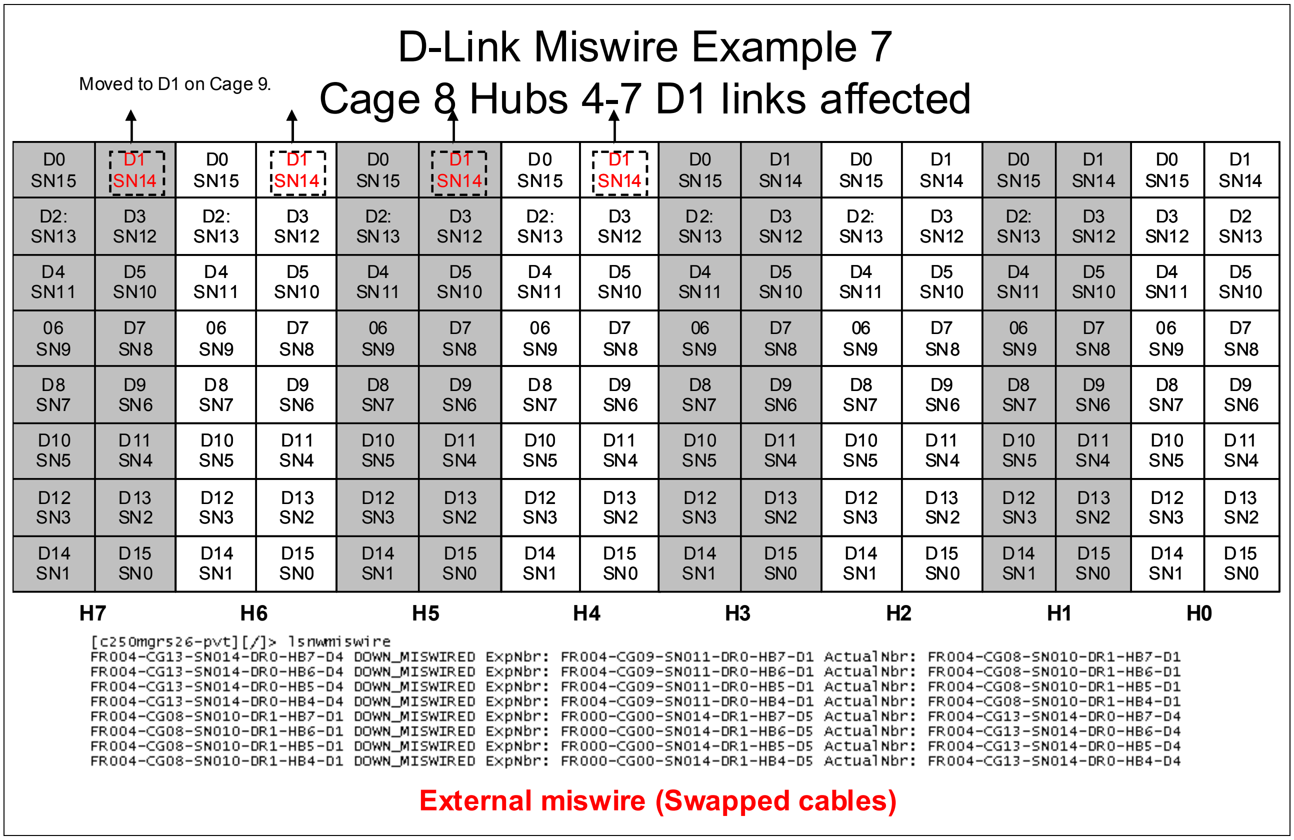
D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D3 SN12	D2 SN13
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		H6		H5		H4		H3		H2		H1		H0	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG11-SN012-DR0-HB5-D2 DOWN_MISWIRED ExpNbr: FR004-CG12-SN013-DR0-HB5-D3 ActualNbr: FR004-CG12-SN013-DR0-HB4-D3
FR004-CG11-SN012-DR0-HB4-D2 DOWN_MISWIRED ExpNbr: FR004-CG12-SN013-DR0-HB4-D3 ActualNbr: FR004-CG12-SN013-DR0-HB5-D3
```

External miswire (Swapped cables)

Figure B-5 Miswire example 6 - External miswire (swapped cables)

Figure B-6 on page 334 shows a D-Link miswire example on cage 8, hubs 4-7, and D1 links affected.



D-Link Miswire Example 7
Cage 8 Hubs 4-7 D1 links affected

Moved to D1 on Cage 9.

D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14	D0 SN15	D1 SN14
D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D2: SN13	D3 SN12	D3 SN12	D2 SN13
D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10	D4 SN11	D5 SN10
06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8	06 SN9	D7 SN8
D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6	D8 SN7	D9 SN6
D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4	D10 SN5	D11 SN4
D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2	D12 SN3	D13 SN2
D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0	D14 SN1	D15 SN0
H7		H6		H5		H4		H3		H2		H1		H0	

```
[c250mgrs26-pvt][/]> lsnwmiswire
FR004-CG13-SN014-DR0-HB7-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB7-D1 ActualNbr: FR004-CG08-SN010-DR1-HB7-D1
FR004-CG13-SN014-DR0-HB6-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB6-D1 ActualNbr: FR004-CG08-SN010-DR1-HB6-D1
FR004-CG13-SN014-DR0-HB5-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB5-D1 ActualNbr: FR004-CG08-SN010-DR1-HB5-D1
FR004-CG13-SN014-DR0-HB4-D4 DOWN_MISWIRED ExpNbr: FR004-CG09-SN011-DR0-HB4-D1 ActualNbr: FR004-CG08-SN010-DR1-HB4-D1
FR004-CG08-SN010-DR1-HB7-D1 DOWN_MISWIRED ExpNbr: FR000-CG00-SN014-DR1-HB7-D5 ActualNbr: FR004-CG13-SN014-DR0-HB7-D4
FR004-CG08-SN010-DR1-HB6-D1 DOWN_MISWIRED ExpNbr: FR000-CG00-SN014-DR1-HB6-D5 ActualNbr: FR004-CG13-SN014-DR0-HB6-D4
FR004-CG08-SN010-DR1-HB5-D1 DOWN_MISWIRED ExpNbr: FR000-CG00-SN014-DR1-HB5-D5 ActualNbr: FR004-CG13-SN014-DR0-HB5-D4
FR004-CG08-SN010-DR1-HB4-D1 DOWN_MISWIRED ExpNbr: FR000-CG00-SN014-DR1-HB4-D5 ActualNbr: FR004-CG13-SN014-DR0-HB4-D4
```

External miswire (Swapped cables)

Figure B-6 Miswire example 7 - External miswire (swapped cables)

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publication provides additional information about the topic in this document (this publication might be available only in softcopy):

- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615

You can search for, view, download or order this document and other Redbooks, Redpapers, Web Docs, draft and additional materials, at this website:

`http://www.ibm.com/redbooks`

Other publications

These publications are also relevant as further information sources:

- *Parallel Environment Runtime Edition for AIX: PAMI Programming Guide*, SA23-2273-00
- *ESSL for AIX V5.1 and ESSL for Linux on POWER V5.1Guide and Reference,* SA23-2268-01
- *ESSL for AIX V5.1 Installation Guide*, GA32-0767-00
- *ESSL for Linux on POWER V5.1 Installation Guide*, GA32-0768-00
- *Parallel ESSL for AIX V4.1 Installation*, SA23-1367-00
- *Parallel ESSL for AIX V4.1 Guide and Reference*, GA32-0903-00
- *Tivoli Workload Scheduler LoadLeveler: Using and Administering*, SA22-7881
- *IBM Parallel Environment Runtime Edition: MPI Programming Guide*, SC23-6783
- *IBM Parallel Environment Runtime Edition: PAMI Programming Guide,* SA23-2273

Online resources

These websites are also relevant as further information sources:

- IBM Power Systems hardware information

  `http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp`

- ESSL for AIX V5.1/ESSL for Linux on POWER V5.1

  `http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.essl.doc/esslbooks.html`

- Cluster Products Information Center

  `http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.pe.doc/pebooks.html`

- IBM Fix Central

  `http://www.ibm.com/support/fixcentral`

- Parallel Tools Platform

  http://www.eclipse.org/ptp/

Help from IBM

- IBM Support and downloads

  http://www.ibm.com/support
- IBM Global Services

  http://www.ibm.com/services

Index

IBM Power Systems 775 for AIX and Linux HPC Solution

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

IBM Power Systems 775 for AIX and Linux HPC Solution

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM Power Systems 775 for AIX and Linux HPC Solution

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM Power Systems 775 for AIX and Linux HPC Solution

(0.2”spine)
0.17”<->0.473”
90<->249 pages

(0.1”spine)
0.1”<->0.169”
53<->89 pages

IBM Power Systems 775 for AIX and Linux HPC Solution

(2.5” spine)
2.5”<->nnn.n”
1315<-> nnnn pages

IBM Power Systems 775 for AIX and Linux HPC Solution

(2.0” spine)
2.0” <-> 2.498”
1052 <-> 1314 pages

IBM Power Systems 775 for AIX and Linux HPC Solution

Unleashes computing power for HPC workloads

Provides architectural solution overview

Contains sample scenarios

This IBM Redbooks publication contains information about the IBM Power Systems 775 Supercomputer solution for AIX and Linux HPC customers. This publication provides details about how to plan, configure, maintain, and run HPC workloads in this environment.

This IBM Redbooks document is targeted to current and future users of the IBM Power Systems 775 Supercomputer (consultants, IT architects, support staff, and IT specialists) responsible for delivering and implementing IBM Power Systems 775 clustering solutions for their enterprise high-performance computing applications.

SG24-8003-00 ISBN 073843731X