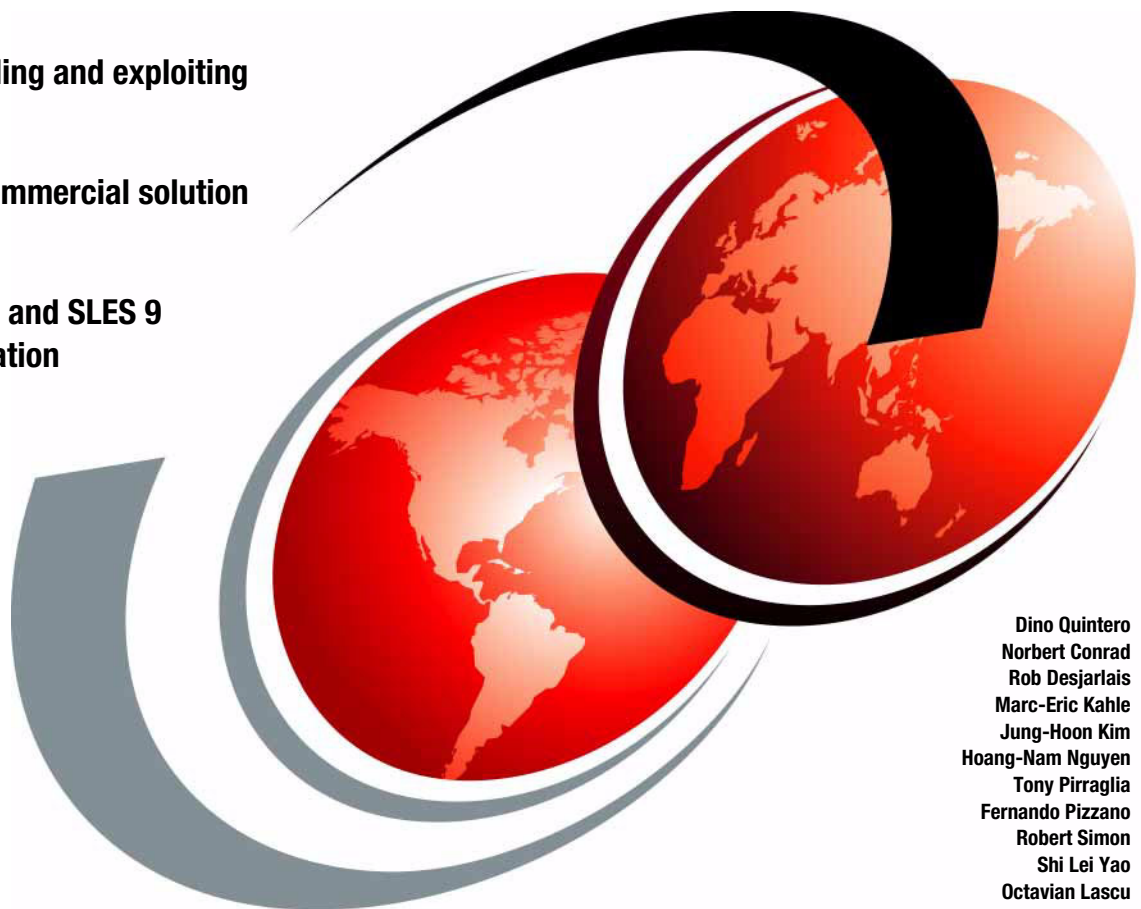


Implementing InfiniBand on IBM System p

Understanding and exploiting
InfiniBand

HPC and commercial solution
explored

AIX 5L V5.3 and SLES 9
implementation



Dino Quintero
Norbert Conrad
Rob Desjarlais
Marc-Eric Kahle
Jung-Hoon Kim
Hoang-Nam Nguyen
Tony Pirraglia
Fernando Pizzano
Robert Simon
Shi Lei Yao
Octavian Lascu



International Technical Support Organization

Implementing InfiniBand on IBM System p

September 2007

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

First Edition (September 2007)

This edition applies to Version 5, Release 3, Modification 4, APAR IY84006 of AIX 5L, of SUSE LINUX Enterprise Server 9 for POWER Service Pack 3, SLES 9 SP3 and Release 6, Version 1.0 and APAR MB01795 of the Hardware Management Console (HMC).

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this book	xi
Become a published author	xiv
Comments welcome	xv
Part 1. InfiniBand architecture	1
Chapter 1. Introduction	3
1.1 Introduction to InfiniBand	4
Chapter 2. Introduction to InfiniBand technology	7
2.1 A technical introduction to InfiniBand	9
2.2 Markets	10
2.3 Application clustering	11
2.4 I/O architectures: fabric versus bus	12
2.4.1 Shared bus architecture	13
2.4.2 New interconnects compliment InfiniBand	13
2.4.3 Bandwidth out of the box	14
2.5 InfiniBand technical overview	14
2.6 InfiniBand layers	16
2.6.1 Physical layer	16
2.6.2 Link layer	17
2.6.3 Network layer	19
2.6.4 Transport layer	20
2.6.5 Upper layers	21
2.6.6 InfiniBand elements	22
2.7 InfiniBand architecture	23
2.7.1 Channel adapters	23
2.7.2 The IB switch	23
2.8 InfiniBand components	24
2.8.1 Router	24
2.8.2 Subnet manager	25
2.8.3 Management infrastructure	25
2.9 InfiniBand support for the Direct Access Programming Library (DAPL) ..	26
2.10 Adapter sharing	26
2.11 Summary	28

Chapter 3. InfiniBand hardware overview and implementation	29
3.1 Limitations and considerations	30
3.2 Features and benefits of InfiniBand on System p	30
3.2.1 AIX supported environments	33
3.2.2 Linux on System p: SLES9 SP3 supported environments	34
3.3 Hardware requirements	35
3.4 Hardware Management Console (HMC)	37
3.4.1 Cluster Ready Hardware Server (CRHS) mode	38
3.4.2 Why move the DHCP server	39
3.5 Supported System p servers	40
3.6 Supported host channel adapters (HCA)	41
3.6.1 Sharing the host channel adapter (HCA)	47
3.7 Logical partitioning (LPAR)	48
3.7.1 Make it smaller (micro partitions)	49
3.8 Cisco InfiniBand switches	50
3.8.1 Cisco SFS 7000P InfiniBand Server Switch	50
3.8.2 Cisco SFS 7008P InfiniBand Server Switch	51
3.8.3 Using the switch: user and passwords	53
3.9 InfiniBand cables	53
3.9.1 Cabling with octopus cables	55
3.10 Management server	57
3.10.1 A private network DHCP IP configuration versus a static IP configuration	58
3.11 IBM Network Manager (IBM NM)	59
Part 2. Implementation	71
Chapter 4. InfiniBand on AIX 5L	73
4.1 InfiniBand on AIX	74
4.1.1 Hardware requirements	74
4.1.2 AIX software requirements	74
4.1.3 Overview of cluster software components	74
4.2 InfiniBand on System p5 running AIX	86
4.2.1 Implementation of InfiniBand architecture (IBA) on System p5	86
4.2.2 IP over InfiniBand (IPoIB) implementation	89
4.2.3 AIX InfiniBand filesets and components	91
4.3 Test cluster layout and description	94
4.3.1 Planning for installation	94
4.3.2 Our environment	95
4.4 Installation and configuration of the AIX CSM Management server	97
4.4.1 Installation of AIX	97
4.4.2 Installing the AIX 5L management server	98
4.4.3 NIM configuration	100

4.4.4	Updating NIM	103
4.4.5	Verify InfiniBand filesets	103
4.5	Installation and configuration of AIX nodes	104
4.5.1	Pre-installation tasks	104
4.5.2	Get network adapter information	110
4.5.3	Further configuration	112
4.5.4	Preparing NIM for nodes (clients) installation	113
4.5.5	Verification of the AIX installation	115
4.5.6	Configuring InfiniBand adapters on AIX nodes	117
4.5.7	Verification of the InfiniBand configuration	121
4.6	GPFS installation and configuration	131
4.6.1	Communication considerations for GPFS	131
4.6.2	GPFS installation	132
4.6.3	Monitoring GPFS over InfiniBand	138
 Chapter 5. IBM System p cluster with InfiniBand and SUSE SLES 9 ..		141
5.1	InfiniBand considerations for SLES 9	142
5.1.1	Supported hardware	142
5.1.2	Software components and versions	143
5.1.3	InfiniBand implementation on SLES 9	145
5.2	Introduction of cluster software components for SLES 9	154
5.2.1	HPC cluster overview	155
5.2.2	System management software for SLES9 clustering	155
5.2.3	Software packages for High Performance Computing	158
5.3	Installation and configuration	165
5.3.1	Planning	165
5.3.2	Sample SLES 9 Cluster layout and description	169
5.3.3	Installation steps for setting up a SLES 9 cluster	170
 Part 3. Support		201
 Chapter 6. Problem determination		203
6.1	IB switch troubleshooting	204
6.1.1	Physical layer issues	204
6.1.2	InfiniBand switch firmware upgrade process	204
6.2	System p troubleshooting	205
6.2.1	HCA troubleshooting	205
6.2.2	Logs available for troubleshooting	206
6.2.3	HMC/IBM Network Manager troubleshooting	206
6.3	AIX troubleshooting	207
6.4	Troubleshooting IB on SLES 9	209
6.4.1	The dmesg tool	209
6.4.2	eHCA device driver version	211
6.4.3	Troubleshooting IP over InfiniBand issues in Linux	221

6.4.4 Troubleshooting an HPC User Space issue under Linux	225
6.5 Application layer troubleshooting	227
6.5.1 LoadLeveler issues	227
6.5.2 CSM issues	230
Chapter 7. Best practices	233
7.1 CSM	234
7.1.1 CSM and NIM strategy	234
7.1.2 Back up CSM data	234
7.1.3 NIM and resolv.conf	234
7.1.4 Nodegroups	235
7.2 Automatic InfiniBand configuration for many nodes	237
7.2.1 Configuring IB adapters in CSM for AIX	238
7.2.2 SLES	242
7.3 PowerPC productivity tools for SLES	243
7.4 Physical server build considerations	246
Chapter 8. Monitoring tools for InfiniBand adapter	249
8.1 Monitoring tools for AIX 5L and SLES 9	250
8.1.1 Useful commands for AIX 5L	250
8.1.2 Monitoring tools for SLES 9	254
8.2 Useful commands for LoadLeveler with InfiniBand	261
Part 4. Appendices	267
Appendix A. InfiniBand security	269
IB Protocol layer	270
IP layer	271
Appendix B. Cluster Ready Hardware Server	273
Cluster Ready Hardware Server basics	274
CRHS and InfiniBand switches	275
Appendix C. Function cross table: Linux for AIX sysadmins	279
Major features	280
Common system files	281
Task-specific command comparison	282
Appendix D. Installing OFED and eHCA on Linux Kernel 2.6.16, 2.6.17, and 2.6.18	285
Steps to install OFED and eHCA on Kernel 2.6.16, 2.6.17, and 2.6.18	286
Abbreviations and acronyms	295
Related publications	299

IBM Redbooks publications	299
Other publications	299
Online resources	300
How to get IBM Redbooks publications	300
Help from IBM	301
Index	303

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

AIX 5L™	IBM®	Redbooks®
AIX®	LoadLeveler®	Redbooks (logo)  ®
BladeCenter®	Micro-Partitioning™	RS/6000®
Blue Gene®	OpenPower™	System p™
Chipkill™	PowerPC®	System p5™
DB2®	POWER™	Tivoli®
eServer™	POWER4™	InfiniBand®
General Parallel File System™	POWER5™	InfiniBand Trade Association®
GPFS™	POWER5+™	
HACMP™	pSeries®	

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

InfiniBand Trade Association, InfiniBand, and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

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

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, other countries, or both.

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

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

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

Preface

InfiniBand is a powerful network interconnect technology designed to deliver I/O connectivity for large network infrastructures. InfiniBand is supported by all major OEM server vendors as a means to deliver the next generation I/O interconnect standard for servers.

This IBM® Redbooks® publication provides information about the InfiniBand standard and how the standard has been implemented to support High Performance Computing (HPC) clustered systems and large enterprise server environments.

This book will help you install, configure, and use the new IBM InfiniBand® adapters to form a high bandwidth, low latency communication network for applications that use either User Space Protocol or IP over InfiniBand (IPoIB).

This book presents the software and hardware components that have to be brought together to form the management and application foundation. We cover Cluster Systems Management, Network Install Manager, Reliable Scalable Clustering Technology, IBM compilers, IBM TWS LoadLeveler®, Parallel Environment, and various other AIX® and SUSE SLES tools.

This book will help you design, manage, and solve issues on the InfiniBand infrastructure in an IBM System p™ environment running either AIX or SUSE SLES 9 based on practical examples tested in our lab at ITSO Poughkeepsie, USA.

The team that wrote this book

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

Dino Quintero is a Senior Certified Consulting IT Specialist and the worldwide Technical Marketing Manager for the IBM System Blue Gene® Solution. Before joining the deep computing marketing team, he was a Clustering Project Leader for the ITSO. He worked as a clustering Performance Analyst for the Enterprise Systems Group focusing on industry benchmarks such as TPC-C, TPC-H, and TPC-W. He also worked as a Disaster Recovery Architect for IBM Global Services focusing on backup and recovery solutions for large customers. He has been with IBM since 1996, and in the IT industry since 1992. His areas of expertise include enterprise backup and recovery, disaster recovery planning

and implementation, and clustering architecture and solutions. He is an IBM Certified Specialist on System p Administration, System p clustering, and he is also an IBM Sr. Certified IT Specialist. Currently, he focuses on planning, influencing, leading, managing, and marketing IBM Blue Gene solution. He also delivers technical lectures worldwide.

Dr. Norbert Conrad is team leader for High Performance Computing in the Hochschulrechenzentrum at the Technical University of Darmstadt in Germany. He has worked with AIX for fifteen years. He holds a degree in physics from the University of Wuerzburg, Germany. His areas of expertise include High Performance Computing, High Availability, and Tivoli® Storage Manager.

Rob Desjarlais is a Principle Systems Analyst in Portsmouth NH. He has 15 years of experience in the IT field. His areas of expertise include system design and architecture, and systems support. This is his first published work.

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

Jung-Hoon Kim is a system P Advisory Service Representative at the IBM Global Technology Services in Daejeon, South Korea. He has six years of experience in System p. He has worked at IBM for six years. His areas of expertise include RS/6000 and pSeries hardware and he is also an AIX certified specialist. His main responsibility is customer support and the implementation of high-end pSeries, RS/6000, and SP solutions.

Hoang-Nam Nguyen is a software engineer at the IBM Systems and Technology Group in Boeblingen, Germany. He holds a master degree in Computer Science from the University of Karlsruhe, Germany. He has 12 years of experience in software and firmware development on various platforms. His areas of expertise include (Java™) service oriented architecture, network and system management, and high performance computing.

Tony Pirraglia is a senior engineer in the ISV Business Strategy and Enablement organization. He is based in Poughkeepsie, NY. He is currently involved with ISV application benchmarks in support of System p product launches, interfacing with the Blue Gene team regarding ISV support, and representing the ISV BSE organization on selected Product Development Teams. Tony has been involved with Scientific and Technical computing on IBM platforms in one form or another since 1988, when he first joined IBM. He received his Doctorate in Chemical Engineering from Columbia University.

Fernando Pizzano is a System Administrator in IBM Advanced Clustering Technology Development Lab, Poughkeepsie, New York. He has nine years of information technology experience. The last seven of those years have been with IBM. 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. His current position is in the Communication Protocols and Application Tools Development department.

Shi Lei Yao is a senior technical sales specialist of the IBM System p brand in China. He has eight years of experience in High Performance Computing (HPC), and three years of experience in System p and IBM eServer™ Cluster 1600 at IBM. He holds a PhD degree in Computing Fluid Dynamics (CFD). His areas of expertise include codes tuning and parallelizing for HPC applications.

Octavian Lascu is a Project Leader at the ITSO, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of IBM System p and Linux® clusters. Before joining the ITSO, Octavian worked in IBM Global Services Romania as a software and hardware Services Manager. He holds a Master's Degree in Electronic Engineering from the Polytechnical Institute in Bucharest and is also an IBM Certified Advanced Technical Expert in AIX/PSSP/HACMP™. He has worked with IBM since 1992.

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

Daniel Powell
IBM Poughkeepsie

Mark Atkins
IBM Boulder

Serban Maerean
IBM Poughkeepsie

Wade Wallace
International Technical Support Organization, Austin Center



Figure 1 Team members, back row, left to right: Dino Quintero, Shi Lei Yao, Dr. Norbert Conrad, Marc-Eric Kahle, Rob Desjarlais, Robert Simon; front row: Hoag-Nam Nguyen (medallion), Jung-Hoon Kim, Octavian Lascu, Fernando Pizzanno, Toni Pirraglia (medallion)

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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



Part 1

InfiniBand architecture

This section introduces the InfiniBand for IBM System p technology and presents some common implementations. We also provide information about the InfiniBand standard and its uses in HPC clustered systems and large enterprise server environments.

InfiniBand is a powerful network interconnect technology designed to deliver I/O connectivity for large network infrastructures. InfiniBand is supported by all major OEM server vendors as a means to deliver the next generation I/O interconnect standard for servers.



Introduction

This book will help system administrators understand the different technologies used to make up an InfiniBand solution on IBM System p Servers. Our team has put together a set of common solutions that would be used in many of the currently supported implementations of InfiniBand. This book is not intended to be all encompassing in its scope. This is a guide to the technology and common implementations of the technology as IBM has implemented it.

As you will see in the following pages, InfiniBand is a very useful and flexible technology, it is gaining maturity, and will help solve many of the problems that system administrators and designers face.

1.1 Introduction to InfiniBand

InfiniBand is an industry standard, high speed, and low latency network transport protocol used to connect system nodes for use in HPC, and Large Enterprise Server installations. This book provides information about InfiniBand and its use in these environments. This book discusses the use of AIX 5L and Linux installations in combination with System p servers and InfiniBand.

The InfiniBand high bandwidth fabric offers high-speed interconnects between servers in a cluster, as well as virtualization features that make a single GX adapter available to up to 16 LPARs. The IBM System p server combined with InfiniBand provides for the creation of powerful computing solutions.

System p Servers when combined with the IBM\Cisco 7000 series or Voltaire switches deliver an excellent combination of high bandwidth (up to 60 GBps with two 12X adapters in use) and low latency. As a result, a System p server cluster can deliver performance comparable to a single high-end monolithic server at lower costs, and allows for better scalability and reliability.

A System p cluster with an InfiniBand solution delivers extremely high performance through the use of the inherent performance capabilities of InfiniBand. The combination of serial communication, high bandwidth, and low latency enables the creation of cost effective clusters of systems that deliver fast inter-processor communications across servers as well as among server clusters. High bandwidth and low latency also permit leveraging the capabilities of distributed databases such as DB2® Parallel edition and Oracle® Real Application Cluster (RAC), to deliver highly scalable and available clusters built on inexpensive server platforms.

This book delivers information about the use of InfiniBand and the IBM HPC software stack for AIX 5L, and Linux. AIX applications such as IBM Tivoli Workload Scheduler LoadLeveler, and General Parallel File System™ (GPFS™) deliver superb HPC capabilities and performance when combined with System p and InfiniBand. Linux applications such as MPI, LAPI, and the Linux implementations of Parallel Environment (PE) and Load Leveler allow for open source HPC solutions that deliver powerful computing solutions.

Data Center administrators can take advantage of the server consolidation and server I/O virtualization capabilities of System p servers with InfiniBand to create a high-bandwidth, low latency, and On Demand computing environment. An On Demand environment permits meeting the various workloads of an organization without overbuying hardware. On Demand computing also greatly reduces infrastructure complexity, simplifying management, reducing space, power and cooling requirements, and cable clutter. On Demand computing when combined

with Sub-Capacity software licensing can create substantial cost savings for many organizations.

An InfiniBand gateway unit allows seamless integration of the InfiniBand network into existing LAN and SAN topologies. A single I/O chassis can provide SAN and LAN capabilities to all systems attached to the InfiniBand network. In addition, the IBM Network Node Manager embeds policy and provisioning intelligence into the solution, enabling true data center resource virtualization. This allows administrators to manage systems from a central point, greatly simplifying operational management of the system. With this management capability comes the ability to add or remove servers from a central point, adjusting services on demand without disruption to service delivery.

The InfiniBand suite of technologies allow a wide array of flexible and capable solutions to be created. Technologies such as Channelization, IP over InfiniBand, RDMA (Remote Direct Memory Access), and SCSI RDMA Protocol (SRP), offer a powerful set of performance and reliability features that make using InfiniBand networks a very good performance solution, and a good value proposition too.



Introduction to InfiniBand technology

This chapter provides information about the InfiniBand standard and its uses in HPC clustered systems and large enterprise server environments.

InfiniBand is a powerful network interconnect technology designed to deliver I/O connectivity for large network infrastructures. InfiniBand is supported by all major OEM server vendors as a means to deliver the next generation I/O interconnect standard for servers. InfiniBand has been designed from the very beginning as being able to deliver excellent Reliability, Availability and Serviceability (RAS) functions and features. Data Center administrators can construct InfiniBand networks with multiple levels of redundancy for reliable operation of the networks. InfiniBand uses a channel based architecture that is inherently more reliable than conventional architectures. InfiniBand uses a message passing system to transfer data that is more efficient and reliable than traditional load and store type configurations.

For details about InfiniBand, see the InfiniBand Trade Association® web page at:
<http://www.InfiniBandta.org/about/>

For High Performance Computing (HPC) applications and Remote Direct Memory Access (RDMA) aware services, data can be copied directly from the memory of the CPU to the memory of the InfiniBand Host Channel Adapter (HCA). This effectively bypasses the kernel interactions, and IP stack on the server. Using these techniques allows for very low latency communications (less than 10 microseconds) between any two elements in an IB network.

2.1 A technical introduction to InfiniBand

InfiniBand is a switch-based serial I/O interconnect architecture. Serial architectures allow for higher speed and longer distance communications. When these high speed connections are combined with the channel architecture of InfiniBand very high throughput, and very low latencies are possible.

The standard nomenclature for InfiniBand transfer data rate speeds is described in Table 2-1.

Table 2-1 *InfiniBand terms and speeds*

Name	Speed	Data rate	Fully duplexed rate
1X	2.5 Gbps	2 Gbps	4 Gbps
4X	10 Gbps	8 Gbps	8 Gbps
12X	30 Gbps	24 Gbps	48 Gbps
1X DDR	5 Gbps	4 Gbps	8 Gbps
4X DDR	20 Gbps	16 Gbps	32 Gbps
12X DDR	60 Gbps	48 Gbps	96 Gbps
1X QDR	10 Gbps	8 Gbps	16 Gbps
4X QDR	40 Gbps	32 Gbps	64 Gbps
12X QDR	120 Gbps	96 Gbps	192 Gbps

The terminology for InfiniBand is set by the IBTA and currently consists of three standards: SDR, DDR, and QDR. SDR stands for Single Data Rate, and is represented by using 1X, without suffix. DDR stands for Double Data Rate, and delivers double the performance of basic InfiniBand. Naming convention specifies that when discussing DDR, to differentiate this from SDR, the DDR suffix must be added to the speed, for example, 1X DDR. Similarly QDR, or Quadruple Data Rate, has four times the performance of SDR, and the suffix to be added is QDR, for example, 1X QDR.

As of this writing, DDR is just coming into the market, and new switches and HCA cards are becoming available. As the QDR technology is still under development, there are future IBM plans to support QDR, but there is no estimated date for the announcement.

Using twisted pair cables up to 17 m in length or fiber cables up to 1000 m in length, InfiniBand networks can span to cover large data center and campus networking requirements. When combined with IP and SAN gateway devices, InfiniBand can be integrated into nearly any systems infrastructure. This integration allows for better utilization of existing LAN and SAN infrastructures. When an InfiniBand network and a traditional LAN and SAN environment are combined together, the high performance features of that LAN and SAN environment can be better leveraged to their fullest value. In other words InfiniBand's virtualization and integration capabilities will allow administrators to finally combine sufficient compute and workload resources to fully utilize their existing IP network and SAN capabilities.

For example, the IBM GX adapter based InfiniBand Host Card Adapter (HCA) allows for extremely high performance interconnects, and superior virtualization capabilities, allowing up to 16 LPARs access to a single HCA device and utilize the (up to) 320 Gbps bandwidth available to this card.

InfiniBand also offers excellent latency characteristics. The average large chassis Ethernet switch can have as much as 100 microseconds latency when delivering data from one port to another. InfiniBand offers much lower latency capabilities with many switches delivering data in 3 to 5 microseconds, from port to port.

2.2 Markets

Large data center managers and administrators continue to run into the problems of capacity and efficient resource utilization. As HPC systems and virtualization tools continue to consolidate workloads and deliver ever greater value and levels of service, InfiniBand allows for the continued extension and expansion of these technologies.

As processor capability and performance continue to grow, high performance interconnects like InfiniBand allow current and future processor technologies to be fed the ever increasing amounts of data they need to operate efficiently.

In environments such as clustered applications and inter-tier communication¹, the QoS and RAS features of InfiniBand will allow customers to better exploit these technologies. By reducing the time it takes for a transaction to transit the various elements of the infrastructure, more transactions can be done with the same amount of hardware.

¹ For example, multiple PCI I/O extension drawers connected to the same CPU.

Many embedded systems (including routers, storage systems, and intelligent switches) utilize the PCI bus architecture, often in the Compact PCI format, for their internal I/O architecture. Such systems are unable to keep up the high speed networking interconnects such as DWDM technologies; therefore, many companies are developing proprietary I/O interconnect architectures to keep up with these faster networks.

Building on the experience of developing Ethernet Local Area Networks (LAN) Fibre Channel Storage Area Networks (SAN) and Wide Area Network (WAN) interconnects, InfiniBand has been designed to exceed the needs of today's markets and provide a cohesive interconnect for a wide range of systems. This is accomplished with extensive attention being paid to items such as RAS, QoS, and scalability.

In its current iteration, InfiniBand has the ability to meet the aggressive demands of today's user community, with defined standards for up to 32X performance. InfiniBand has the ability to scale and grow well into the future.

2.3 Application clustering

The internet today has evolved into an immense global infrastructure supporting numerous applications and services. Each of these applications and services must support an ever increasing volume of data while delivering higher and higher availability. Service providers are, in turn, experiencing tremendous pressure to support these application requirements. At the same time as trying to deliver this improved performance, they are trying to create new offerings such as QoS, and improved security.

Service providers have appeared to support the outsourcing of e-commerce, e-marketing, and other e-business related activities, specializing in delivering these high demand internet-based applications. Providers must now be able to offer highly reliable services that offer the ability to dramatically scale the communication infrastructure capabilities in a short period of time to accommodate the explosive growth of application demand in situations such as major news events or new product releases. Clustering techniques have evolved as the preferred mechanism to support these requirements. A cluster is simply a group of servers connected by load-sharing/balancing elements working in parallel to serve a particular application.

InfiniBand simplifies application cluster connections by unifying networks with a feature rich managed architecture. InfiniBand's switched architecture provides native cluster connectivity, thus supporting scalability and reliability for clustering. Devices can be added and multiple paths can be utilized with the addition of switches to the fabric. High-priority transactions between devices can be processed ahead of lower-priority items through the QoS (Quality of Service) mechanisms built into InfiniBand.

Application cluster features such as Sockets Direct Protocol, and RDMA allow integrated products to communicate at much higher rates of speed.

2.4 I/O architectures: fabric versus bus

The shared bus architecture is the most commonly I/O interconnect today, despite numerous drawbacks, such as contention and congestion. Clusters and networks require systems with high speed, fault tolerant interconnects that cannot be supported properly using a bus architecture. Thus all bus architectures require network interface models to enable scalable network topologies. To keep pace with systems, an I/O architecture must provide a high speed connection with the ability to scale. Table 2-2 provides a simple feature comparison between a switched fabric architecture and a shared bus architecture.

Table 2-2 Fabric versus bus comparison

Feature	Fabric	Bus
Topology	Switched	Shared bus
Interconnect pin count	Low	High
Number of endpoints	Many (theoretically infinite)	Few (limited)
Max signal length	Kilometers (theoretically unlimited)	Centimeters
Reliability	Yes	No
Scalability	Yes	Limited
Fault tolerant	Yes	No

A fabric architecture allows for a many to many type of communication relationship to be created. This has many benefits: It is more resilient to a component failure, it offers the ability to communicate over more than one path at one time, and it allows for multiple endpoints to communicate with each other without interfering with each other.

Generally fabrics are more resilient and have less contention for resources. An important consideration in the design of these systems is that a fabric can be extended by adding another fabric unit to the whole, and with careful planning and implementation this can greatly enhance the capacity of the whole fabric. Using this feature, fabrics can scale in an almost limitless way, where bus configurations can only scale in a linear manner.

Fabric architectures are also more complex to operate, and as a result, they tend to be more expensive to acquire and manage. In most situations, cost and complexity are far outweighed by the performance and reliability of these devices.

2.4.1 Shared bus architecture

In a shared bus architecture, the bandwidth of the bus is divided among all devices on that bus. If any one device consumes all of the bandwidth on the bus, none of the other devices on that bus can communicate. Bus configurations can have severe electrical signal, mechanical, and power issues. For example, parallel bus requires many pins for each connection requiring large amounts of system board real estate. For example, 16X PCIe uses 164 pins, as described in the following Web document:

http://www.interfacebus.com/Design_Connector_PCI_Express.html

At high bus frequencies, the distance of each signal is limited to short traces on the system board. In a slot-based system with multiple card slots, bus termination is difficult to control and can cause problems if not designed properly. These limitations make shared bus architectures undesirable for applications that demand high performance and high scalability.

Shared buses can be extended by using a technique called *bridging*. This allows for the implementation of more slots, but can have negative bandwidth implications. Latency and congestion increase with each addition of a bus bridge.

2.4.2 New interconnects compliment InfiniBand

New interconnects such as PCI Express and the GX bus allow system architects and administrators to better exploit InfiniBand's capabilities. PCI Express architecture continues to be developed, and offers ever greater amounts of bandwidth to supply today's faster processors with the data they need to deliver faster performance and better user experiences. PCI Express is a standard 64-bit shared bus interconnect that in most implementations can support up to six endpoints (PCI devices) on a single bus.

GX bus adapters use the 64-bit fully meshed loop topology of IBM System p Servers to deliver high bandwidth and low latency communications. GX bus is theoretically unlimited in its ability to interconnect devices. The IBM System p model 595 has a maximum of 32 GX bus connections. The bandwidth of these systems is impressive, and its low latency makes this a natural choice for being the interface to use for InfiniBand adapters.

2.4.3 Bandwidth out of the box

A fundamental concept of the InfiniBand architecture is that of *bandwidth out of the box*. InfiniBand has the ability to deliver data to devices outside the server central electronic complex (CEC) at speeds generally only seen inside the server backplanes. This performance allows for new services to be offered, such as video on demand, and traditional services, like Web services, to deliver new levels of performance and serviceability.

Historically, bandwidth decreases as distance away from the CPU increases. An example of this is that typical CPU to memory communications are measured in the tens of gigabytes per second range of throughput, while network technologies commonly deliver only mega- to gigabits per second of throughput, and have substantial overhead that lowers their utility. InfiniBand delivers gigabyte per second range performance over distances measured in kilometers with only a small amount of overhead.

Current state of the art processors are able to communicate with memory at speeds up to 30 gigabytes per second, but PCI-X and other bus interconnects are also limited in their bandwidth, typically 5 to 10 gigabytes per second. The GX bus adapter of the System p servers can sustain transfer rates up to 320 gigabytes per second.

2.5 InfiniBand technical overview

InfiniBand is a point to point interconnect developed for today's systems with the ability to scale to meet the increasing bandwidth demands of today's computer users. Each individual link is based on a four wire 2.5 Gbps bidirectional connection. The architecture defines a layered hardware protocol as well as a software layer to manage initialization and the communication between devices. Each link can support multiple transport services for reliability and multiple prioritized virtual communication channels.

To manage the communication within an IB subnet, the architecture defines a communication management scheme that is responsible for configuring and maintaining each of the InfiniBand fabric elements. Management schemes are defined for error reporting, link failover, chassis management, and other services to ensure a cohesive communication environment.

The InfiniBand feature set includes:

- ▶ Layered protocol: Physical, link, network, transport, and upper layers
- ▶ Packet based communication
- ▶ Quality of Service
- ▶ Six link speeds 1X, 1X DDR, 4X, 4X DDR, 12X, and 12X DDR
- ▶ 1X - 2.5Gbps, 4 wire
- ▶ 4X - 10Gbps, 16 wire
- ▶ 12X - 30Gbps, 48 wire
- ▶ Copper or Fiber Cable interconnect
- ▶ Dual port adapters
- ▶ Subnet Management Protocol
- ▶ Remote DMA support (RDMA)
- ▶ Multicast and Unicast IP support
- ▶ Reliable transport methods: message queuing
- ▶ Communication flow control at the Link Layer and end to end

2.6 InfiniBand layers

The InfiniBand architecture is divided into multiple layers, where each layer operates independently of the other, as shown in Figure 2-1.

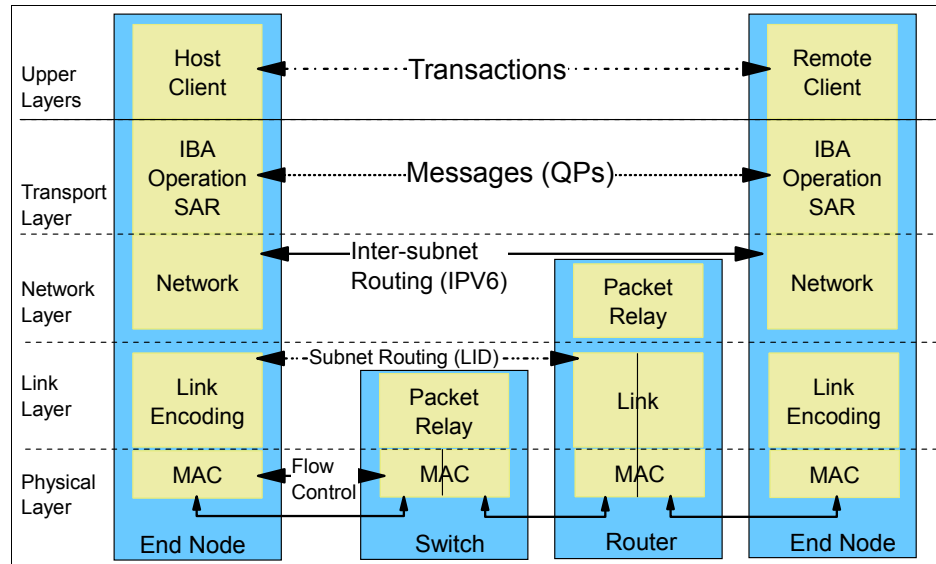


Figure 2-1 InfiniBand layers

2.6.1 Physical layer

InfiniBand is a comprehensive architecture that defines electrical and mechanical specifications for this technology. The specifications include cables, receptacles, and connectors and how they all work together, and how they should behave in certain situations such as when a part is hot-swapped.

Currently InfiniBand defines six link speeds at the physical layer: 1X, 1X DDR, 4X, 4X DDR, and 12X, and 12X DDR (QDR is still under development). Each individual link is a four wire serial differential connection (two wires in each direction) that provide a full duplex connection at 2.5 Gbps for Single Data Rate, and 5 Gbps for DDR (10 Gbps for the upcoming QDR). These links are illustrated in Figure 2-2 on page 17.

All IBM InfiniBand adapters have two ports on them so that they can be attached to two separate switches, allowing for architectures with no single points of failure.

Different InfiniBand cables are required for the different performance levels of InfiniBand. As you can see in Figure 2-2, higher bandwidth solutions require cables with more pairs of wire. The speed designation is based on the numbers of send and receive pairs in each interface. For example, a 1X InfiniBand cable has one pair of send and one pair of receive cables, while a 12X connection has 12 send and 12 receive pairs. The InfiniBand standard defines capabilities up to 32X, but the only available solutions on the market today are 12X or lower.

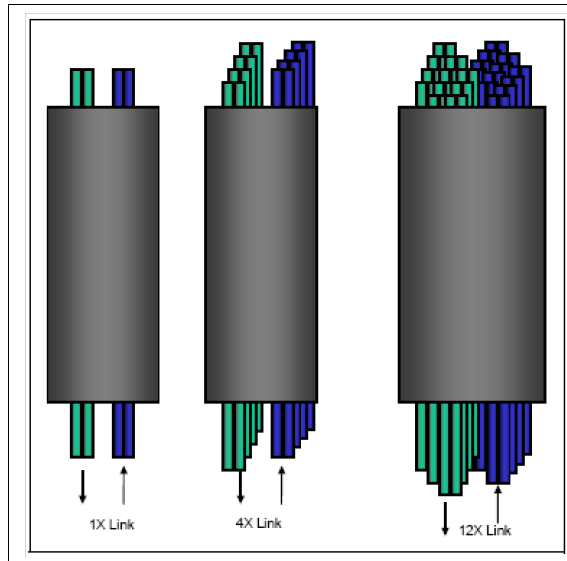


Figure 2-2 InfiniBand cable configurations

2.6.2 Link layer

The link layer (along with the transport layer) is the heart of the InfiniBand architecture. The link layer encompasses packet layout, point-to-point link operations, and switching within a local subnet. The link layer (LL) is responsible for the orderly handling of data packets, link creation and management, Virtual Lane (VL) operations, packet flow control, and packet error checking. The serial nature of the protocol ensures that data will arrive in the appropriate order.

Virtual Lanes

InfiniBand allows the use of multiple independent data streams (flows) over the same physical link. This feature is called Virtual Lanes (VLs). VLs are separate logical flows that have their own buffering and flow control. This allows for more efficient and rapid communication between devices as no one buffer or task can slow all the communications on the physical connection. Current implementation allows for 16 Virtual Lanes (0...15), Virtual Lane 15 being the highest priority virtual lane, and is reserved for all management operations on the InfiniBand network.

Packets

There are two types of packets in the link layer: management and data packets. Management packets are used for link maintenance and configuration. They are also used to determine device information, such as virtual lane support. Data packets carry the actual payload data that is used by the applications on the systems. Data packets can carry up to 4 KB of data.

Switching

Within a subnet, packet forwarding and switching is handled at the link layer. All devices on a subnet have a 16-bit local identifier (LID) assigned by the subnet manager. All packets sent within a subnet use the LID for addressing. Link level switching forwards the packets to the device specified by a destination LID with a local router header (LRH) in the packet. The LRH is present in all packets.

Please note that currently IBM IB solution does not support the cascading of switches.

QoS

Quality of Service is supported by InfiniBand through the Virtual Lanes feature. Virtual Lanes (VLs) are separate logical communication links that share a single physical link. Each link can support up to 15 Standard VLs and one (1) Management lane (VL15). VL 15 is the highest priority VL and VL 0 is the lowest priority VL. Management packets use VL15 exclusively. Each device must support as a minimum VL0 and VL15 while other VLs (1...14) are optional.

As a packet traverses a subnet, a Service Level (SL) is divided to ensure its QoS level. Each link along a path can have a different VL, and the SL provides each link with the desired priority of communication. Each switch/router has an SL-to-VL mapping table that is set by the Subnet Manager to keep the proper priority with the number of VLs supported on each link. Therefore, the InfiniBand Architecture can ensure end-to-end QoS through switches and routers.

Credit based flow control

Link level flow control is used to manage data flow between two point-to-point links. Flow control is handled on a per-VL basis, allowing separate virtual fabrics to maintain communication utilizing the same physical media. Each receiving end of a link supplies credits to the sending device on the link to specify the amount of data that can be received without the loss of data. Credit passing between each device is managed by a dedicated link packet to update the number of data packets the receiver can accept. Data is not transmitted unless the receiver advertised credits indicate that receive buffer space is available.

Data integrity

At the link level, CRCs per packet are used to ensure data integrity. These are named Variant CRC and Invariant CRC. The 16-bit VCRC includes all fields in the packet and is recalculated at each hop. The 32-bit ICRC covers only fields that do not change from hop to hop. The VCRC provides link level data integrity between the two hops and the ICRC provides end-to-end data integrity. In a protocol such as Ethernet, which defines only one CRC, an error can be introduced within a device that then recalculates the CRC. The check at the next hop would reveal a valid CRC, even though the data has been corrected. InfiniBand included the ICRC so that when a bit error is introduced, the error will always be detected.

2.6.3 Network layer

The network layer handles routing of packets from one subnet to another. (Within a subnet, the network layer is not required.) Packets that are sent between subnets contain a global route header (GRH). The GRH contains the 128-bit IPv6 address for the source and destination of the packet. The packets are forwarded between subnets through a router, based on each device's 64-bit Globally Unique Identifier (GUID). The router modifies the Local Route Header (LRH, 16-bit) with the proper local address within each subnet. Therefore, the last router in the path replaces the LID in the LRH with the LID of the destination port. Within the network layer, InfiniBand packets do not require the network layer information and header overhead when used within a single subnet (which is a likely scenario for InfiniBand system area networks).

2.6.4 Transport layer

The transport layer is responsible for in-order packet delivery, partitioning, channel multiplexing, and transport services (reliable connection, reliable datagram, unreliable connection, unreliable datagram, and raw datagram). The transport layer also handles transaction data segmentation at sending, and reassembly at receiving. Based on the maximum transfer unit (MTU) of the path, the transport layer divides the data into packets of the proper size. The receiver reassembles the packets based on a Base Transport Header (BTH) that contains the destination queue pair and packet sequence number. The receiver acknowledges the packets and the sender receives these acknowledgements and updates the completion queue with the status of the operation.

InfiniBand architecture offers a significant improvement for the transport layer over other technologies such as Ethernet: all functions are implemented in the hardware. InfiniBand specifies multiple transport services for data reliability. Table 2-3 describes each of the supported services. For a given queue pair, one transport level is used.

Table 2-3 Transport services

Class of service	Description
Reliable connection	Acknowledged - connection oriented
Reliable datagram	Acknowledged - multiplexed
Unreliable connection	Unacknowledged - connection oriented
Unreliable datagram	Unacknowledged - connectionless
Raw datagram	Unacknowledged - connectionless

Reliable and unreliable datagrams

InfiniBand defines two basic data types: reliable datagrams and unreliable datagrams. Reliable datagrams are used when both of the following requirements are in place: The upper layer protocol does not deliver data reliability, and communication needs to be extremely reliable. Reliable datagrams deliver good performance, but pose a performance limitation as each packet must be checked as it arrives, and placed in the correct order, or a resend must be performed. In most situations where reliable datagrams would be used, administrators generally choose to use IP over InfiniBand.

Unreliable datagrams are used by applications that have built-in data checking capabilities. Most HPC applications use unreliable datagrams, relying on the upper layer application to recognize when data is missing, and requesting a re-submission of the missing packets. Unreliable datagrams are useful because of the channelized nature of InfiniBand. The channel structures deliver robust communications to the point where unreliable datagrams can be used for most services and applications.

2.6.5 Upper layers

There are several upper layer services offered by InfiniBand, such as Sockets Direct Protocol (SDP), SCSI RDMA Protocol (SRP), and many others. These services are used to enable applications to communicate with various devices attached to the IB fabric.

The discussion in the following paragraphs is far from being exhaustive, as these services vary in their deployments and implementations from vendor to vendor.

SRP

SRP is used by InfiniBand attached devices to communicate with disk storage systems that are attached to the InfiniBand fabric. These devices can be attached either directly with a Target Channel Adapter (TCA), or through an InfiniBand to SAN bridge. SRP allows InfiniBand attached servers to exploit the high bandwidth and low latency of InfiniBand and to improve server performance access to data storage devices like SAN attached disk and tape subsystems. SRP allows InfiniBand attached servers to seamlessly integrate into existing SAN infrastructures, and leverage existing investments in storage systems and technologies.

SDP

Sockets Direct Protocol (SDP) is used by InfiniBand attached servers to communicate directly with different elements in the InfiniBand fabric. SDP utilizes RDMA type functionality to allow devices to communicate streaming data to each other. SDP is implemented as an Application Programming Interface (API) that can be used to allow application access to the power of the InfiniBand architecture.

SDP facilitates the direct mapping of stream type connections to InfiniBand reliable connections or channels. For more information, refer to the following document:

http://infiniband.sourceforge.net/archive/LinuxSAS_SDP.pdf

Remote Direct Memory Access

Remote Direct Memory Access (RDMA) is used to allow different servers on the InfiniBand fabric to access the memory of another server directly. An example for this would be a database server cluster. The database server cluster uses an RDMA agent to be added to its core functionality, which allows two database instances running on different nodes to communicate directly with each other, bypassing all of the kernel level communication operations, thus reducing the number of times the data is copied from persistent storage into the RAM memory of the cluster nodes.

RDMA allows data to move from the main memory in the address space of a user process on the server directly to the memory buffer on the HCA, which then delivers the data to the remote HCA, and then from the HCA directly into the memory of the remote system. This is a tremendous improvement in the communication efficiency over TCP based systems, as all of the (TCP generated) kernel calls are skipped, and all of the CRC computations in the system CPU are passed to the HCA for both sending and receiving sides.

By off-loading the CRC operations to the HCA, the system CPU can deliver more productive work. RDMA implementations make servers more efficient to operate, and allow the exploitation of ever faster processor systems.

Because of the high performance nature of InfiniBand, this allows for systems like database servers to deliver faster responses and more efficient operations.

2.6.6 InfiniBand elements

The InfiniBand Architecture defines multiple devices for system communication: a channel adapter, switch, router, and subnet manager. Within a subnet, there must be at least one channel adapter for each end node and a subnet manager to set up and maintain the link. All channel adapters and switches must contain a Subnet Management Agent (SMA) required for handling communication with the subnet manager (SM).

The subnet manager creates the queue pairs that define the channels between two devices on the IB network. A HCA has a limit of 64,000 queue pairs. and a Queue Pair is the information that defines an InfiniBand channel.

Every HCA has at least one Queue Pair between the subnet manager and the HCA. Each time an HCA talks to another HCA, a QP is created between the two HCAs.

2.7 InfiniBand architecture

In this section, we discuss the two items that make up an InfiniBand Network Fabric, Channel Adapters, and Switches.

2.7.1 Channel adapters

A channel adapter is the physical device that connects two InfiniBand devices. The InfiniBand standard defines two types of channel adapters: Host Channel Adapters (HCA), and Target Channel Adapters (TCA). For the purposes of this book, we will focus only on HCAs.

An HCA provides an interface to a host device and supports all of the “verbs” supported by InfiniBand. Verbs are an abstract representation that defines the required interface between the client software and the functions of the HCA. Verbs are items (defined by the IBTA standards committee) that allow the device driver and the hardware to work together.

A TCA is a specialized type of HCA. A TCA would be used in a data storage device, and generally does not have the full functionality and resources of an HCA. Based on the application, a TCA may have various set of features, for example, a TCA in a storage device would have a different set of features than a TCA in a printer.

Each HCA has a physical network address (128-bit wide). The physical address of each card follows the same format and rules as IPv6. This address is known as the Globally Unique Identifier (GUID).

2.7.2 The IB switch

Switches are the fundamental components of an InfiniBand fabric. A switch contains several InfiniBand ports that are allowed to move traffic between each other. A switch does not consume or generate data traffic, other than management traffic. Switches can be configured to forward either unicast packets or multicast packets.

Switch fabrics are the foundation for the utility of an InfiniBand network. Switches allow many servers to communicate with one another, and deliver enormous amounts of work. Currently supported implementations allow up to 96 servers to work together in a single fabric. The general topology of an IB fabric is shown in Figure 2-3.

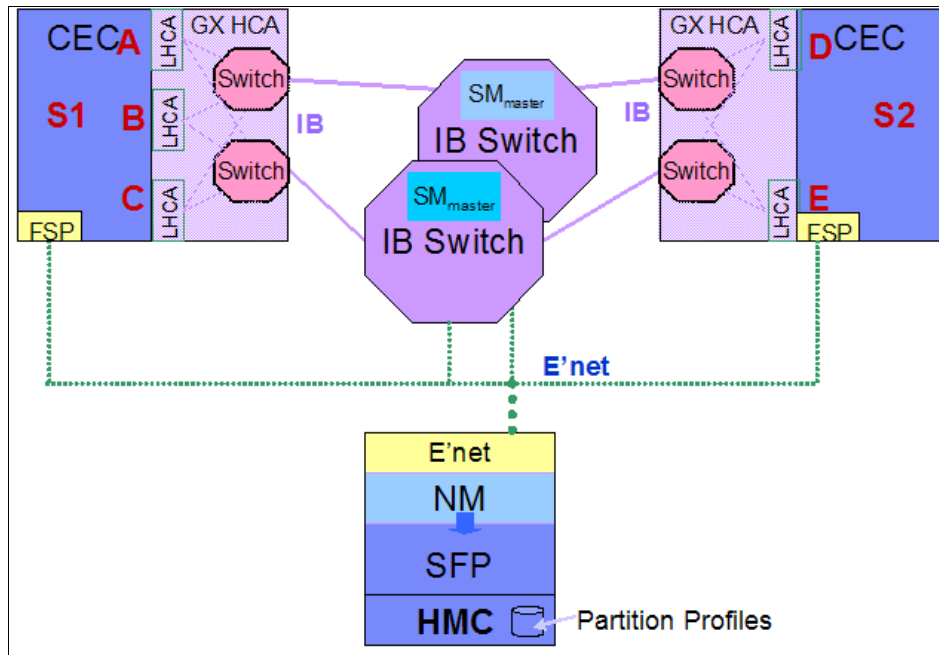


Figure 2-3 Sample InfiniBand network diagram

2.8 InfiniBand components

This section presents the management components of an IB network.

2.8.1 Router

InfiniBand routers forward packets from one subnet to another without consuming or generating packets. Unlike a switch, a router reads the global route header of a packet, and redirects that packet to the appropriate subnet based on its IPv6 network layer address. The router rebuilds each packet with the proper Logical Identifier (LID) on the next Subnet. The router keeps a table of the current conversations that are underway, and ensures that the packets that are going to and coming from the next hop arrive back at the correct device.

2.8.2 Subnet manager

The subnet configures the local subnet and ensures its continued operation. There is always at least one Subnet Manager in each subnet to monitor and maintain operations between all of the InfiniBand devices in the subnet. The Subnet Manager controls all switch and router setups when a new device comes into the subnet. The subnet manager communicates to all devices on the subnet through the Subnet Manager Adapter native to each InfiniBand device. This is a regular component of every InfiniBand port.

There can be multiple subnet managers on a subnet, but only one can be the master at any one time. All other subnet managers on the subnet are slaved to the master until the master becomes unavailable. Slave subnet managers keep a full copy of the master subnet manager forwarding and routing information to prevent the subnet from going down if the master should go down.

2.8.3 Management infrastructure

The InfiniBand architecture defines two methods of system management for handling all operations (link-up, link-down, maintenance, and general service functions) associated with a subnet. These methods, Subnet Management, and General Services Interface, has a dedicated Queue Pair that is supported by all devices on the subnet to distinguish management traffic from all other traffic.

Subnet management

Subnet management is handled by the subnet manager device generally operating on the InfiniBand switch. This processor is the “brain” of the InfiniBand switch and creates all of the automatic configurations that the protocols define, such as Link creation, Queue Pair operations, LID assignments, Link Failover, and so on. By handling these operations in the background, all configuration management is kept separate from client software and operations. This simplifies operations and provides better reliability and security.

All subnet management operates over queue pair 0, which is handled exclusively on Virtual Lane 15 to ensure the highest priority in the network. No other resources can use VL15 or QP0. VL15 uses the unreliable datagram service and does not follow the same flow control consideration as other VLs in the network. This feature ensures that the InfiniBand network is reliable and resilient to single device failures on the InfiniBand network.

General services interface

The second method defined by the InfiniBand Standard is the General Services Interface (GSI). This port delivers chassis management operations and information, and uses regular Virtual Lanes, such as VL1, and are subject to the availability of switch resources.

2.9 InfiniBand support for the Direct Access Programming Library (DAPL)

The Direct Access Programming Library is a distributed messaging technology that is both hardware-independent and compatible with current network interconnects. The architecture provides an API that can be utilized to provide high-speed and low-latency communications among peers in clustered applications.

InfiniBand was developed with the DAPL architecture in mind. InfiniBand offloads traffic control from the software client through the use of execution queues. These queues, called work queues, are initiated by the client, then left for InfiniBand to manage. For each communication channel between devices, a work queue pair (WQP - send and receive queues) is assigned at each end. The client places a transaction into the work queue (work queue entry (WQE)), which is then processed by the channel adapter from the send queue and sent to the remote device. When the remote device responds, the channel adapter returns its status to the client through a completion queue or event.

The client can post multiple WQEs, and the channel adapter's hardware handles each of the communication requests. The channel adapter then generates a completion queue entry (CQE) to provide status for each WQE in the proper prioritized order. This enables the client to continue with other activities while the transactions are being processed.

2.10 Adapter sharing

A key competitive advantage for InfiniBand is its ability to divide resources among multiple tasks or partitions. The GX adapter can support up to 64,000 simultaneous queue pairs. This can be divided into 64 slices of 1000 Queue Pairs each. The IBTA standard allows for millions of queue pairs, but most implementations are scaled down for simplicity, supportability, and price reasons.

Adapter sharing is a feature specific to the IBM GX bus InfiniBand adapter. Up to 16 LPARs can access the resources on one GX Adapter. Each LPAR can receive a Virtual Adapter from the GX HCA card. Each HCA can deliver one virtual adapter to each LPAR. The HCA is capable of creating up to 64 Virtual adapters, but it there is little value in adding more than one virtual interface from the same HCA to the same LPAR.

Each Virtual Adapter is given a unique GUID. This is created on the HMC when a new LPAR is created. Please refer to Chapter 3, “InfiniBand hardware overview and implementation” on page 29 for more information about how to add an HCA to a partition.

When running an HCA under Linux, it is possible to run both IP-over-IB and SCSI (SRP) over the same Virtual Adapter. Each protocol uses one Queue Pair.

Adapter sharing divides up the resources of the InfiniBand adapter between the different LPARs on a frame. Each adapter can be assigned a priority of low, medium, or high. This priority assignment defines the quantity of resources, memory and Queue pairs, on the HCA that each LPAR can consume. This has performance impacts and should be carefully considered prior to implementation. If a change is made to the priority of the HCA, then the LPAR must be rebooted for the change to take effect. It is possible to assign all of the resources on one HCA to one LPAR, but this effectively eliminates the ability to share the HCA.

HCA resource allocation for an LPAR partition (using HMC) is shown in Figure 2-4:

- ▶ Dedicated HCA = 100% of HCA resources (complete HCA)
- ▶ High utilization = 25% of HCA resources (1/4 of an HCA)
- ▶ Medium utilization = 12.5% of HCA resources (1/8 of an HCA)
- ▶ Low utilization = 6.25% of HCA resources (1/16 of an HCA)

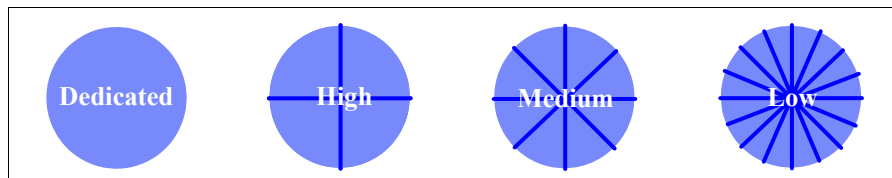


Figure 2-4 HCA resource division in a shared adapter

Careful planning is needed when using adapter sharing. It is possible to negatively impact performance by dividing the resources too finely between the different LPARs on a system. The key element is to understand your workload, and how many queue pairs will be needed by your implementation. In an HPC environment, each task will consume one queue pair, so as more MPI resources are used, more queue pairs will be needed. Alternately, if you are using a commercial computing setup, with just IP over IB, then only one queue pair is needed, and the low, resource division is quite appropriate. Section 6.2.1, “HCA troubleshooting” on page 205 discusses some of the issues that may arise if you have incorrectly set up your adapter sharing configuration.

2.11 Summary

The networking industry is converging on InfiniBand as the future of High Performance Networking. The only real competition for InfiniBand that is still being developed is 10 Gb Ethernet. With InfiniBand implementations that will soon exceed the 100 Gbps mark, and improving virtualization support, InfiniBand will continue to grow in implementation and support. As current technologies expand and mature, so too will the number of InfiniBand implementations.

The IBTA has a vision to improve and simplify the data center through InfiniBand technology and the fabric it creates as an interconnect for server, communications, and storage. This vision when combined with current growth trends in the technology industry means that InfiniBand has a healthy future before it.

When looked at from a holistic perspective, the InfiniBand suite of technologies offer a remarkable set of opportunities for high performance application services and protocols. If developed to their natural limits, InfiniBand products and technologies can help system designers and administrators deliver high performance server solutions.

IBM System p Servers when combined with InfiniBand networks can deliver remarkable performance and value to organizations and businesses.



InfiniBand hardware overview and implementation

This chapter provides information about InfiniBand hardware and the different components needed to use InfiniBand on IBM System p. The following topics are discussed in this chapter:

- ▶ Limitations and considerations
- ▶ Features and benefits of InfiniBand on System p
- ▶ Hardware requirements
- ▶ Hardware Management Console (HMC)
- ▶ Supported System p servers
- ▶ Supported host channel adapters (HCA)
- ▶ Logical partitioning (LPAR)
- ▶ Cisco InfiniBand switches
- ▶ InfiniBand cables
- ▶ Management server
- ▶ IBM Network Manager (IBM NM)

3.1 Limitations and considerations

The current available InfiniBand solutions provided by IBM have some limitations and considerations that must be observed. Refer to Table 3-1 for a list of limitations and considerations.

Table 3-1 *Limitations and considerations*

Limitations and considerations
<ul style="list-style-type: none">▶ A maximum of 32 endpoints for AIX and 96 endpoints for Linux on System p.▶ Supported operating systems include AIX 5L V5.3 and SLES9 SP3 on Linux on System p. See the tables of supported hardware and software.▶ A maximum of one switch per subnet except in a BladeCenter® cascaded switch configuration.▶ A maximum of four subnets (above two networks through RPQ 8A1573) in a Galaxy1 configuration.▶ A maximum of one subnet for Blades.▶ Only homogeneous OSI configurations in a single subnet (that is, AIX or Linux on System p).▶ P5 servers must be installed on a private network.▶ For clusters with a single HMC and not using Cluster Ready Hardware Server, the HMC must be configured to provide DHCP services.▶ For clusters with multiple HMCs, a Cluster Systems Management - Management Server (CSM MS) is required. The CSM MS serves as the DHCP server and Cluster Ready Hardware Server must be used.▶ If using Cluster Ready Hardware Server, the switches must be set to use static-IP. Otherwise, for the case with a single HMC in a cluster and no Cluster Ready Hardware Server in use, the switches need to be set to use the DHCP service.▶ Configurations with multiple subnets are required to use different GID-prefix IDs for switches on different subnets.▶ A maximum of 512 tasks or 1024 tasks in p5 SMT mode.
<ul style="list-style-type: none">▶ Cluster configuration must use IBM provided InfiniBand (IB) cables.▶ IBM Network Manager (IBM NM) supports only IBM 7048-120, IBM 7048-270, Cisco SFS 7000P, and Cisco SFS 7008P switches.▶ Non-IBM NM interfaces are not supported.▶ Preemption is not supported on Tivoli Workload Scheduler (TWS) LoadLeveler for Linux when running in an InfiniBand cluster environment.

3.2 Features and benefits of InfiniBand on System p

Besides the current limitations and considerations, the InfiniBand technology on System p still offers lots of benefits. When combined with IBM System p Servers, InfiniBand allows for immense bandwidth and IO performance in a compact and power efficient package. The IBM Power architecture combines power processor

technologies, with large memory bandwidth, superior reliability and serviceability capabilities, and excellent IO performance.

When solution designers use IBM Power CPU based systems with InfiniBand, it becomes possible to exploit the features and capabilities of those systems to the full extent. Some examples are InfiniBand's high bandwidth and low latency allows the network to keep up with the data demands of the POWER5+™ CPUs. InfiniBand supports the high memory bandwidth capabilities of the POWER™ architecture by again keeping memory fed with up to date data from other systems on the fabric. Adapter sharing allows for easier and more sophisticated uses of micro partitioning and shared processor pool resources.

Refer to Table 3-2 for an overview of the features and benefits of System p and the InfiniBand technology working together.

Table 3-2 Features and benefits

Features and benefits of InfiniBand on System p
<ul style="list-style-type: none"> ▶ POWER5+ microprocessor <ul style="list-style-type: none"> – Designed to provide excellent application performance and high reliability – Includes simultaneous multithreading to help increase commercial system performance and processor utilization – High memory / I/O bandwidth – Fast processors wait less for data to be moved through the system – Delivers data faster for the needs of HPC and other memory-intensive applications
<ul style="list-style-type: none"> ▶ Shared processor pool <ul style="list-style-type: none"> – Provides the ability to transparently share processing power between partitions – Helps balance processing power and helps make sure the high priority partitions get the processor cycles they need
<ul style="list-style-type: none"> ▶ Micro-Partitioning™ <ul style="list-style-type: none"> – Allows each processor in the shared processor pool to be split into as many as 10 partitions – Fine tune processing power to match workloads

Features and benefits of InfiniBand on System p

- ▶ Linux operating system
 - Enables access to 32- and 64-bit open source applications
 - Provides a common operating environment across IBM hardware and software platforms
- ▶ Virtual I/O Server
 - Helps save cost and ease systems administration by sharing expensive resources
- ▶ Virtual LAN
 - Helps speed internal communication between partitions at memory speeds
- ▶ Dynamic logical partitioning
 - Allows reallocation of system resources without rebooting affected partitions
 - Offers greater flexibility in using available capacity and more rapidly matching resources to changing business requirements
- ▶ Mainframe-inspired RAS features
 - Delivers exceptional system availability using features usually found on much larger, more expensive systems including service processor, redundant service processor, Chipkill™ memory, First Failure Data Capture, dynamic deallocation of selected system resources, hot-plug PCI-X slots, hot-swappable disk bays, redundant hot-plug power and cooling, hot-add I/O drawers, dynamic firmware updates, and more
- ▶ Scale-out with CSM support
 - Allows for more granular growth so user demands can be readily satisfied
 - Provides centralized management of multiple interconnected systems
 - Provides ability to handle unexpected workload peaks by sharing resources
- ▶ Multiple operating system support
 - Allows clients the flexibility to select the right operating system and the right application to meet their needs
 - Provides the ability to expand applications choices to include many open source applications
- ▶ AIX 5L operating system
 - Designed to deliver maximum throughput for mixed workloads without complex system configuration or tuning
 - Delivers integrated security features designed for system protection
 - Extends application choices with Linux affinity

3.2.1 AIX supported environments

The following environments are supported with AIX:

- ▶ IP Protocol family over IPoIB
- ▶ MPI User-Space over IB

The following considerations and limitations are known for the supported hardware and software:

- ▶ A maximum of 64 endpoints for AIX on System p
- ▶ A maximum of 64 Operating System Images (OSIs) for AIX System p in a single cluster
- ▶ A maximum of 64 servers for AIX on System p in a single cluster

Table 3-3 shows the current (as of July 2007) supported software and cluster capabilities for System p nodes running AIX.

Table 3-3 AIX and IB supported software stack

Component	p5 52A	p5 55A	p5 570	p5 575	p5 590/5	JS21
AIX						
CSM	X	X	X	X	X	X
RSCT	X	X	X	X	X	X
PE		X		X		X
LoadLeveler	X	X	X	X	X	X
GPFS	X	X	X	X	X	X
ESSL		X		X		X
PESSL		X		X		X
Node scaling	32	32	32	32	32	64
Task scaling	512	1024	512	1024	512	512
Subnets	1	2	1	4	2	1

3.2.2 Linux on System p: SLES9 SP3 supported environments

The following environments are supported with Linux:

- ▶ MPI User-Space over IB
- ▶ MPI over IPoIB
- ▶ IP Protocol family over IPoIB.

The following considerations and limitations are known for the supported hardware and software:

- ▶ A maximum of 96 endpoints for Linux on System p.
- ▶ A maximum of 96 OSIs for Linux on System p in a single cluster.
- ▶ A maximum of 96 Servers for Linux on System p in a single cluster.
- ▶ Preemption is not supported on TWS LoadLeveler for Linux when running in an InfiniBand cluster environment.

Table 3-4 shows the current (as of July 2007) supported software and cluster capabilities for System p nodes running SUSE SLES 9 SP3.

Table 3-4 SLES 9 on System p with IB supported software stack

Component	p5 52A	p5 55A	p5 570	p5 575	p5 590/5	JS21
SLES 9 SP3						
CSM	X	X	X	X	X	
RSCT	X	X	X	X	X	
PE		X		X		
LoadLeveler	X	X	X	X	X	
GPFS	X	X	X	X	X	
ESSL		X		X		
PESSL		X		X		
Node scaling	32	32	32	32	32	
Task scaling						
Subnets	1	2	1	4	2	

For the latest news and support, check the following Web page:

<http://www14.software.ibm.com/webapp/set2/sas/f/networkmanager/home.htm>
1

3.3 Hardware requirements

This section presents the hardware prerequisites and the recommended code levels used in our test environment.

Global firmware (GFW) and HMC minimum levels

Firmware Level SF235_160 and HMC code Version 5, Release 1.0 is the minimum requirement.

The IBM System p Series machines that are supported with the InfiniBand adapters need a minimum level of system firmware installed to be fully functional. For POWER5™ servers, the minimum required firmware Level is SF235_160. The latest level will differ from that level. Make sure you use always the correct, supported level for your POWER5 system. The firmware can be downloaded from the following Web site:

<http://techsupport.services.ibm.com/server/mdownload2/download.html>

Current levels (at the date of this writing):

- ▶ Fix Pack SF240_261 for Global Firmware
- ▶ Fix Pack BP240_197 for Power Subsystem Microcode

See also POWER5 Firmware Readme at:

http://www14.software.ibm.com/webapp/set2/sas/f/hps/related/Power5_Firmware_ReadMe.html

The minimum HMC code level is Version 5, Release 1.0. The current level for HMCs is Version 6, Release 1.0 (for System p5™). This level can be downloaded from the following Web site:

<http://www14.software.ibm.com/webapp/set2/sas/f/hmc/power5/download/v61.Recovery.html>

Refer to the README for details on the installation/upgrade process. Always check for prerequisites and notes on the page. Sometimes additional BIOS upgrades on the HMC itself has to be performed.

You can use the HMC GUI or the HMC command line to verify the currently installed release level with the `lshmc -V` command. Refer to Example 3-1 for a view of the command output.

Example 3-1 lshmc -V command

```
hscroot@hmcib01:~> lshmc -V
"version= Version: 6
  Release: 1.0
HMC Build level 20060801.1
MH00833: Required fixes for V6R1.0 (09-01-2006)
","base_version=V6.1.0
"
```

You can use one of the following methods to verify the currently installed firmware level:

- ▶ On a running AIX (Linux needs the IBM diagnostics RPM package installed) partition, run the `lsmcode` command, as shown in Example 3-2.

Example 3-2 lsmcode command

```
[ibaix1] [/]> lsmcode
DISPLAY MICROCODE LEVEL
IBM,9131-52A
```

```
The current permanent system firmware image is SF240_258
The current temporary system firmware image is SF240_261
The system is currently booted from the temporary firmware image.
```

- ▶ When an HMC is attached to the POWER5 system, the installed firmware level can be verified from the HMC GUI. Refer to the InfoCenter topic “Getting fixes and upgrades” at:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iph5/iph5_applying_fixes.htm

IBM Network Manager (IBM NM)

This NM release requires that the HMC is installed at the V6.1.0 level. IBM Network Manager fixes can be downloaded from the Hardware Management Console (HMC) service located at:

<http://www14.software.ibm.com/webapp/set2/sas/f/hmc/home.html>

- ▶ Click **HMC V6.1.0** and follow the instructions for installing this HMC level.
- ▶ Select the **HPSNM/IBNM fixes** tab.
- ▶ Download APAR MB01385, PTF MH00696.

Cisco/Topspin switch firmware

Driver levels can be obtained at the following Cisco Web site:

<http://www.cisco.com/public/sw-center/website>

You will be required to register at least one of the Cisco/Topspin hardware components to obtain access to the Cisco Web site.

The latest code level that has been tested in IBM labs (at the time of this writing) is:

- ▶ SFS 7000P/TS120 Driver Level -Topspin OS-2.7.0
- ▶ SFS 7008P/TS270 Driver Level -Topspin OS-2.7.0

After initial switch installation and recognition of the switch by the IBM Network Manager, you may check the level of code on the switch using the following procedure, which is documented in the section “Checking switch software levels” in the “Guide to clustering systems using InfiniBand hardware Installation and service information”, which may be found in the HMC Information Center.

3.4 Hardware Management Console (HMC)

The Hardware Management Console (HMC) is used to control the System p servers connected to a private network. Control functions such as power on/off, dynamic logical partitioning (DLPAR), logical partitioning (LPAR), and capacity upgrade on demand (CUoD) are handled by the HMC. In the InfiniBand environment, the HMC has an important function: it runs the IBM Network Manager (IBM NM) that manages the InfiniBand network. Refer to 3.11, “IBM Network Manager (IBM NM)” on page 59 for more information.

Even if the HMC does not need to be a DHCP server, we recommend using the HMC as a DHCP server for the attached System p machines, especially when used together with the InfiniBand switches. The IBM Network Manager (IBM NM) uses the DHCP information to assign the management IP address of the switch. To add a switch manually to the IBM NM, you must call the next level of support to get a guided way to do that. The use of DHCP is different when Cluster Ready Hardware Server (CRHS) is used. Please refer to 3.4.1, “Cluster Ready Hardware Server (CRHS) mode” on page 38 for more information about this topic.

The HMC uses a Ethernet connection (a private LAN) to connect to the attached System p machines Flexible Service Processor (FSP). The FSP stores all the information about LPARs, CUoD, and other configuration information. The FSP offers two Ethernet ports that can be connected to two HMCs in a redundant HMC configuration. Both HMCs can access the server through the FSP and when one HMC has to be maintained, repaired, upgraded, or updated, the second HMC is still able to manage the attached servers.

For more information about the HMC, please refer to the following Web page:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp?topic=/iphai/hardwaremanagementconsolehmc.htm>

3.4.1 Cluster Ready Hardware Server (CRHS) mode

The term Cluster Ready Hardware Server (CRHS) refers to a set of software that enhances the ability to control HMC-managed System p5 nodes (including OpenPower™ nodes). In single HMC environments, the CRHS is not needed, but it can be used. CRHS is useful to manage multiple HMCs, as this software enables control functions and advantages like:

- ▶ Setting up passwords and installation of the managed servers
- ▶ A hardware server daemon
- ▶ Consolidation of service networks into a maximum of two VLANs for redundancy
- ▶ Reduced HMC requirements for recovery (N + 1, where N is the minimum number of HMCs required for your environment)
- ▶ Automated discovery of System p5 server and HMCs, database registration initialization, and automatic hardware configuration updates
- ▶ Automatic association of System p5 575, 590 and 595 server with frames
- ▶ Ease of movement of System p5 servers between HMCs
- ▶ Shared database of System p cluster hardware information

Refer to Figure 3-1 for an example of the Cluster Service Network Topology.

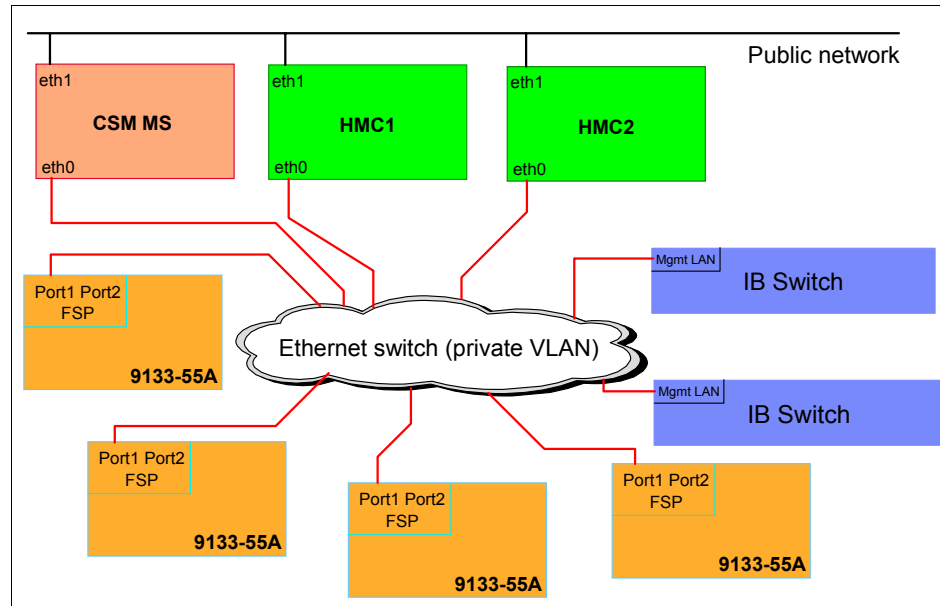


Figure 3-1 Cluster service network topology

For more information about CRHS and how to configure it is either Linux or AIX please refer to Appendix B, “Cluster Ready Hardware Server” on page 273.

3.4.2 Why move the DHCP server

When Cluster Ready Hardware Server (CRHS) is used, the HMC is no longer the main control point for the cluster. When you have multiple HMCs, or have chosen to use CRHS with CSM, the CSM Management Server is required to be the DHCP server for the service network.

The HMCs no longer can serve as DHCP servers, as it is not possible to have multiple DHCP servers configured (on HMCs) on the same IP subnet (this could lead to unpredictable results). You need to change the HMCs network settings by disabling the DHCP server.

3.5 Supported System p servers

Using InfiniBand technology on System p requires selected machines. Only GX adapters are supported. Refer to Table 3-5 for a overview of machines supported with InfiniBand adapters.

Table 3-5 supported System p and IB adapter feature codes (FC)

Machine type	Speed	Adapter feature code (FC)	Maximum number of adapters per system
9131-52A	4x	1812	1
9113-550	4x	1809	1
9133-55A			1
9116-561	4x	1810	1
9117-570			1
9118-575	4x	1811	1
9119-590	12x	7820	4
9119-595			4

The GX adapters offer higher bandwidth than PCI adapters. They plug directly into the GX bus, which is only available on selected System p5 models. The GX bus is directly connected to the system backplane (together with the processors and the memory. This bus is also used for RIO-2 connectivity to connect expansion I/O drawers (that also contains PCI slots). The GX slots are used for the 4x and 12x InfiniBand adapters.

In order to get the GX bus activated on System p systems 9113-550 and 9117-570, a second processor card must be installed. It is not mandatory to have all processors activated (processors installed as CUoD is OK); the only requirement is to have them physically installed into the CEC.

Currently, only the 9119-590/595 offers support for the IBM GX Dual-port 12x IB HCA adapter, which uses different cables than the 4x adapters used to connect to the InfiniBand switch.

For the latest information, check the manual *System p and eServer pSeries: Clustering systems using InfiniBand hardware*, which can be retrieved online at:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphau/infinbdpdf.pdf>

3.6 Supported host channel adapters (HCA)

The GX adapters available and supported for the IBM System p show up as Host Channel Adapters (HCAs). Several feature codes (FC) are available for the System p machines. These adapters have been designed by and are currently manufactured by IBM.

The InfiniBand technology itself is a connection oriented fabric. To control and manage the connections, the Subnet Manager (SM) comes into play. The SM is managing the InfiniBand network in conjunction with the Subnet Administration (SA). The SM discovers the network and has a total overview of the InfiniBand network at all times, and the SM sets the routes for all connections through the InfiniBand fabric (network).

Each adapter has a Globally Unique Identifier (GUID), which is similar to the MAC address of the Ethernet NICs, except it is 64 bits long (for example: 00:05:ad:00:00:02:61:2c). This is an IEEE defined 64-bit identifier assigned to a device. The EUI-64 is a 64-bit identifier is created by concatenating a 24-bit company_id value and a 40-bit extension identifier. The company_id is assigned by the IEEE Registration Authority and the extension identifier is assigned by the organization with the assigned company_id. For more information, see:

<http://www.standards.ieee.org/regauth/oui/tutorials/EUI64.html>

The GUID numbers assigned to the LPARs can be checked on the HMC with the **lshwres** command, as shown in Example 3-3.

Example 3-3 GUIDs assigned/unassigned

```
hscroot@hmcib01:~> lshwres -r hca -m ib01 --level sys
adapter_id=230001ff,phys_loc=U787F.001.DPM18BL-P1-C21,allocation_allowe
d=1,state=1,"unassigned_guids=255005000273c, 255005000273b,
255005000273a, 2550050002739, 2550050002738, 2550050002737,
2550050002736, 2550050002735, 2550050002734, 2550050002733,
2550050002732, 2550050002731, 2550050002730, 255005000272f,
255005000272e, 255005000272d, 255005000272c, 255005000272b,
255005000272a, 2550050002729, 2550050002728, 2550050002727,
2550050002726, 2550050002725, 2550050002724, 2550050002723,
2550050002722, 2550050002721, 2550050002720, 255005000271f,
255005000271e, 255005000271d, 255005000271c, 255005000271b,
255005000271a, 2550050002719, 2550050002718, 2550050002717,
2550050002716, 2550050002715, 2550050002714, 2550050002713,
2550050002712, 2550050002711, 2550050002710, 255005000270f,
255005000270e, 255005000270d, 255005000270c, 255005000270b,
255005000270a, 2550050002709, 2550050002708, 2550050002707,
2550050002706, 2550050002705, 2550050002704, 2550050002703,
```

```
2550050002702, 2550050002701,  
2550050002700",assigned_guids=255005000273d
```

Since each GUID must be different in a network, the IBM HCA gets a subsequent GUID assigned by the firmware. You can choose the offset that should be used for the logical HCA. This information is also stored in the LPAR profile on the HMC.

Attention: You need to update the partition profiles with the new GUID when an adapter is replaced. If this step is not performed, the HCA is not available to the operating system.

The originally used GUID index for the replaced adapter cannot be used again for the new adapter. When the LPAR ID value was 2 and you used the same value as the GUID index value, it cannot be used again. When the GUID index values 1-4 were previously used, only a GUID index higher than 4 can be used.

IBM IB adapters

The following section presents the IBM IB adapters available at the date of this writing.

12x InfiniBand adapter FC 7820

Figure 3-2 on page 43 presents the 12x InfiniBand adapter feature code 7820.

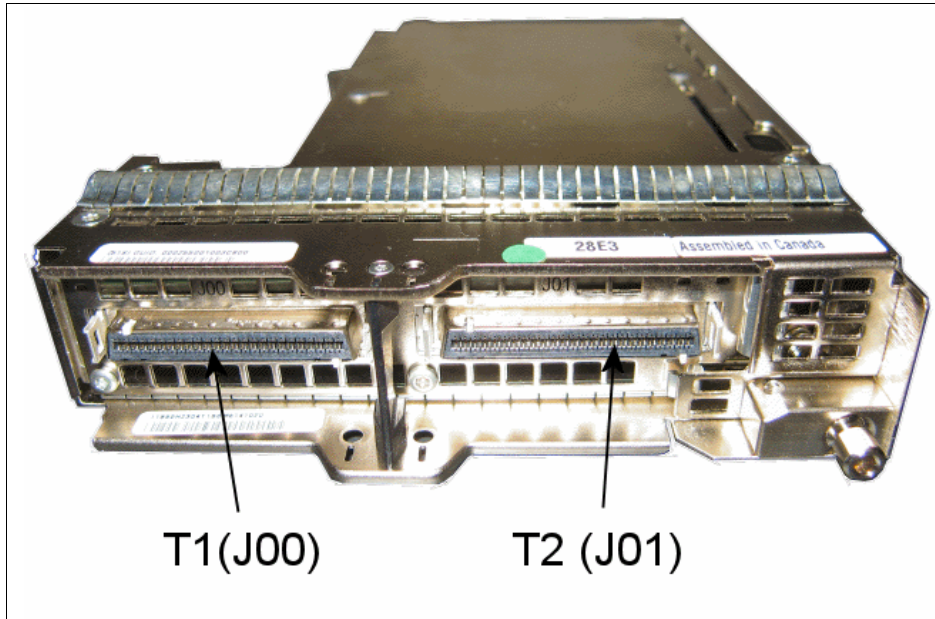


Figure 3-2 Feature 7820 12x InfiniBand adapter

4x InfiniBand adapter FC 1811

Figure 3-3 shows the 4x InfiniBand adapter feature code 1811.

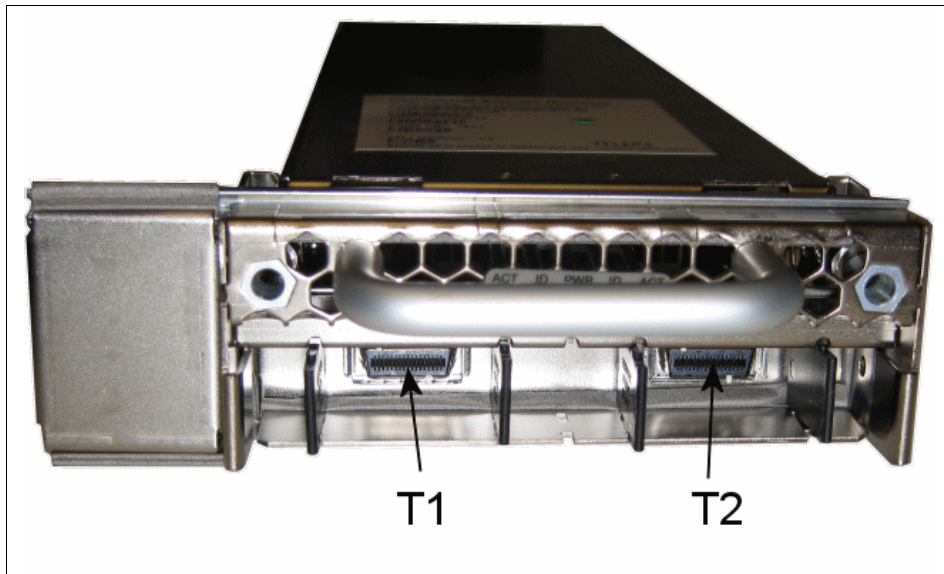


Figure 3-3 Feature 1811 4x InfiniBand Adapter

4x InfiniBand adapter FC 1809

Figure 3-4 shows the 4x InfiniBand adapter feature code 1809.



Figure 3-4 Feature 1809 4x InfiniBand Adapter

4x InfiniBand adapter FC 1810

Figure 3-5 shows the 4x InfiniBand adapter feature code 1810.

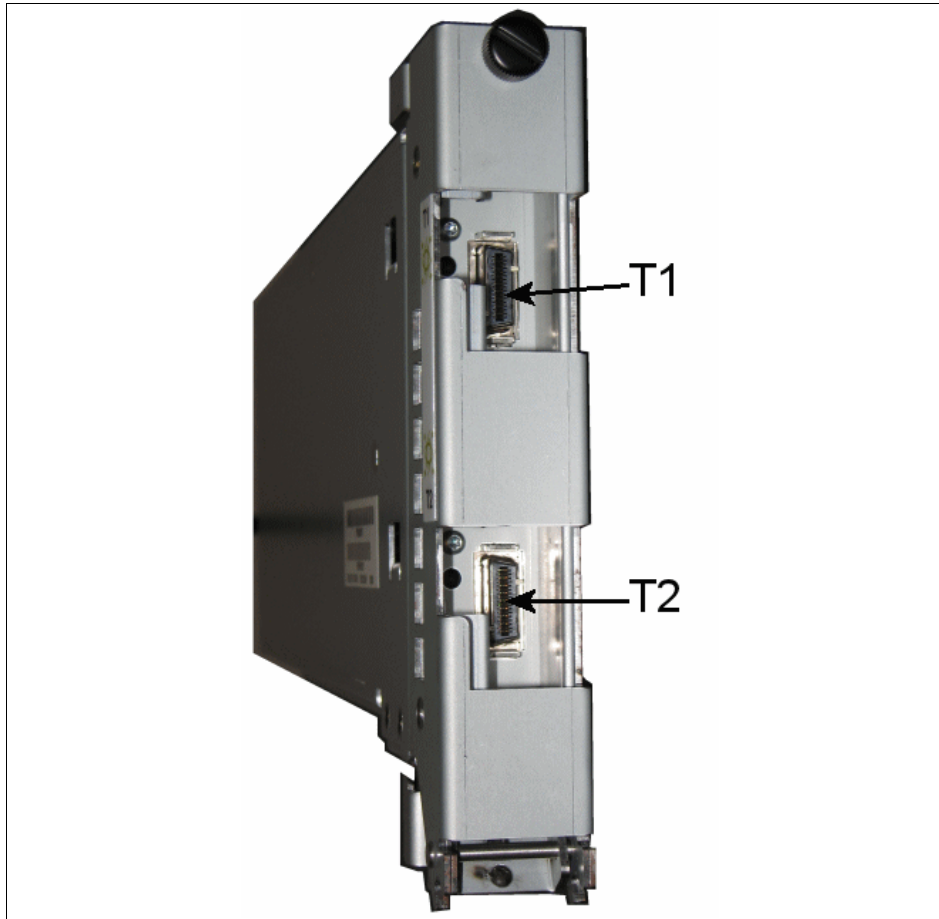


Figure 3-5 Feature 1810 4x InfiniBand Adapter

4x InfiniBand adapter FC 1812

Figure 3-6 shows the 4x InfiniBand adapter feature code 1812.

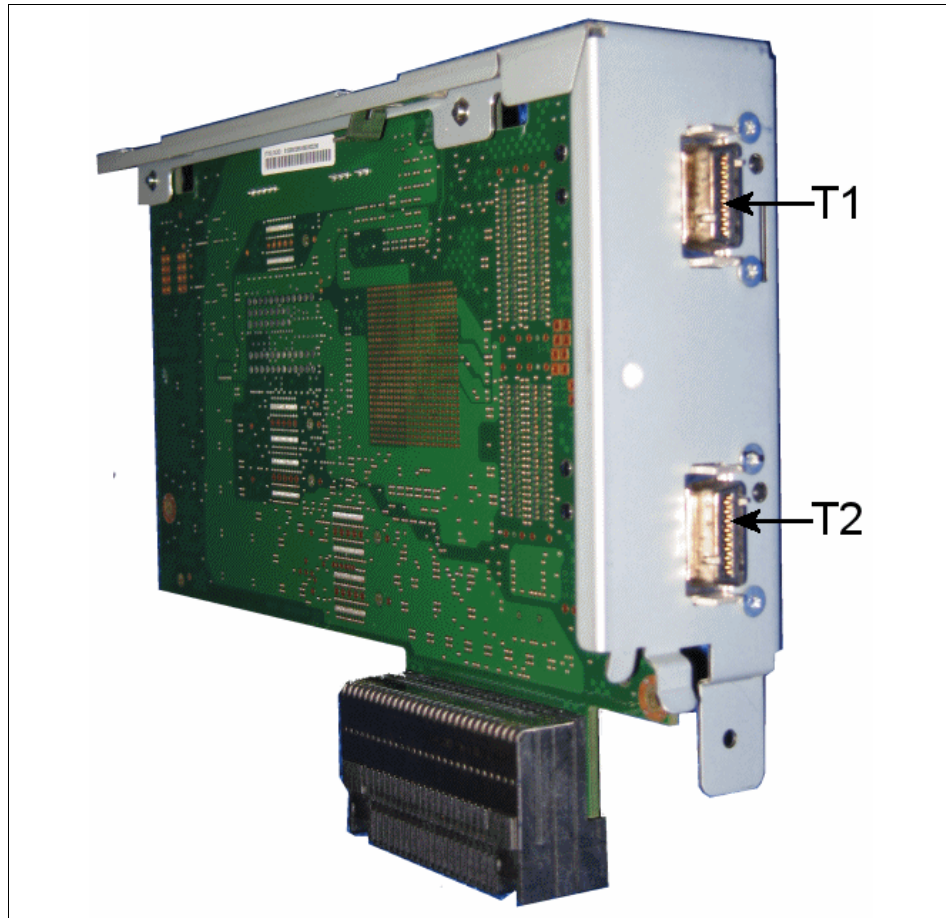


Figure 3-6 Feature 1812 4x InfiniBand adapter

HCA installation orientation

Refer to Table 3-6 on page 47 for information containing HCA installation orientation, location code, and port ID information.

All available GX adapters are located in the back of each Machine Type, except the GX adapter in M/T 9119-59x, which is located in the front of the machine.

Table 3-6 HCA installation orientation

Machine type	HCA orientation	HCA port ID	HCA location code	IBM NM port ID	IBM NM location code
9131-52A	Vertical	1,2	T1,T2	1,2	T1,T2
9133-55A	Vertical	1,2	T2,T1	1,2	T2,T1
9117-570	Vertical	2,1	T1,T2	2,1	T1,T2
9118-575	Horizontal	1,2	T1,T2	1,2	T1,T2
9119-59x	Horizontal	2,1	T1,T2	2,1	T1,T2

The following specifications apply to the information provided in Table 3-6:

- ▶ Vertically oriented HCA ports are numbered from top to bottom.
- ▶ Horizontally oriented HCA ports are numbered from left to right.
- ▶ The Port IDs (Logical port) representations are used to configure the AIX and Linux IB device drivers.
- ▶ The location codes (Physical port) representations are used to cable the two ports of the HCA to the IB switch or switches.

3.6.1 Sharing the host channel adapter (HCA)

The IBM GX Dual-port adapter can be shared between partitions. Therefore, a Hardware Management Console (HMC) is required to use and control the IBM GX Dual-port InfiniBand adapter. A maximum of 16 LPARs sharing the one adapter is supported. When a Host Channel Adapter is installed, the SM is communicating with the Subnet Management Agent (SMA) that resides in the Power Hypervisor. The SM itself will see two logical switches and sixteen (0-15) logical HCAs. Each partition is only aware of its assigned logical HCA. For each partition profile, a GUID is selected with a logical HCA. The GUID is programmed in the adapter itself and cannot be changed.

The GX Dual-port InfiniBand adapter will show up in the HMC GUI Properties tab under HCA. There you can see the current usage of the adapter, GUID, capability, and other information. The resource allocation options define the usage of each GX dual port adapter. There are four possible usage modes:

- ▶ Dedicated: 100% of the HCA resources are allocated.
- ▶ High: 25% of the HCA resources are allocated (1/4 of the HCA).
- ▶ Medium: 12.5% of the HCA resources are allocated (1/8 of the HCA).
- ▶ Low: 6.25% of the HCA resources are allocated (1/16 of the HCA).

This allocation will only apply to the adapter resources.

For more information about adapter sharing, please refer to 2.10, “Adapter sharing” on page 26.

3.7 Logical partitioning (LPAR)

When you define your Logical Partition (LPAR), it is useful that the configuration for the Host Channel Adapter (HCA) uses the same GUID index value as the Partition ID.

For example, let us assume you have an LPAR with the Partition ID 3 and the value for the GUID index is usually a number between 0-62. To keep track of the LPARs and the GUIDs, it makes sense to use the same value for both index and ID. So, choosing a GUID index of 3 (same as Partition ID) should be used. Refer to Figure 3-7 on page 49 for screen capture of the HMC GUI configuration window.

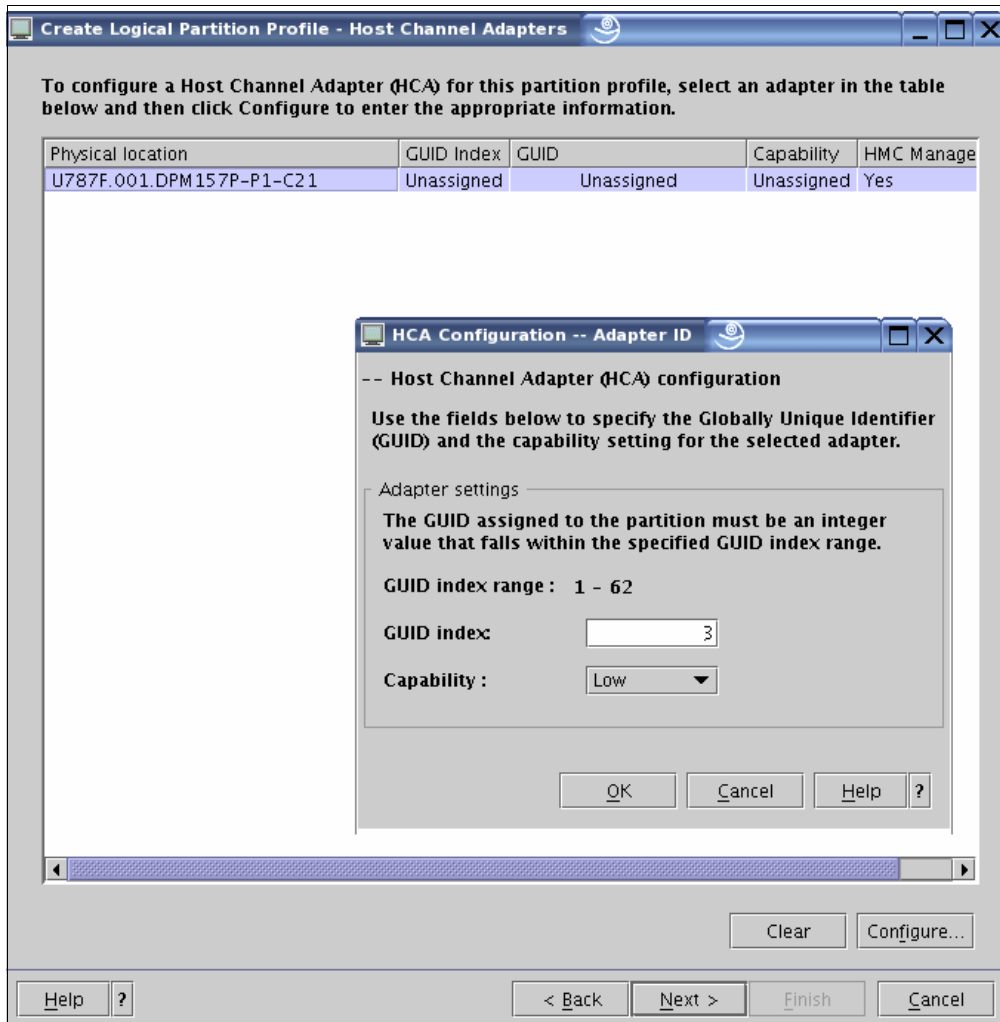


Figure 3-7 GUID Index configuration window

3.7.1 Make it smaller (micro partitions)

The System p architecture allows you to use small chunks of the hardware resources. Each shared processor can be split between as many as 10 partitions. In certain cases, considering the workload on your system, this might be an optimal solution. The InfiniBand GX adapter can be shared between 16 partitions. Thus, it is possible to use 0.1 processor (10%) and 1/16 of the IB HCA (6.25%). However, no more than 16 partitions can share an IB HCA adapter.

3.8 Cisco InfiniBand switches

The Cisco InfiniBand Server Switch Models SFS 7000p and SFS 7008P are currently supported for configurations with IBM InfiniBand Hardware¹. The Cisco InfiniBand Server Switch SFS 7000P (7048-120) is a compact, cost-effective 1U rack-mount unit that provides 24 ports of non-blocking 4x (10 Gbps, full duplex) InfiniBand connectivity.

The Cisco InfiniBand Server Switch SFS 7008P (7048-270) is a 6 U rack-mount unit that offers 48 to 96 ports at 4x connectivity. Both switches offer hot-plug, redundant power and cooling, one RS-232 serial port, and one Ethernet management port. The Model SFS 7008P provides many other features designed to minimize downtime, including hot-swappable field replaceable units (FRUs) with automatic failover capability.

3.8.1 Cisco SFS 7000P InfiniBand Server Switch

The Cisco SFS 27000P InfiniBand Server Switch is an ideal combination of price, performance, and packaging for building high-performance computing clusters. Fully nonblocking with embedded subnet management, it offers 24 ports of 10 Gbps connectivity in a single, compact 1U chassis.

The technical specifications of the SFS 7000P switch are:

- ▶ Width: Standard 19-inch rack-mountable
- ▶ Height: 1 U (1.75 inch) 22-inch depth
- ▶ Depth: 22-inch depth
- ▶ Weight: <30 lbs
- ▶ Air Flow: Front-to-rear
- ▶ Temperature operating: 0 degrees C to 40 degrees C
- ▶ Storage: -25 degrees C to 65 degrees C
- ▶ Altitude, operating: 10,000 feet
- ▶ Storage: 35,000 feet
- ▶ Humidity, operating: 20 percent to 80 percent non-condensing
- ▶ Storage: Five percent to 95 percent non-condensing
- ▶ Shock, operating: 5 G, 11 ms half-sine wave
- ▶ Storage: 10 G, 11 ms half-sine wave
- ▶ Vibration, operating: 0.25 G, 5 to 300 Hz, 15 min.
- ▶ Storage: 0.5 G 5-300 Hz, 15 min.
- ▶ Power 90 to 264 V AC auto-ranging, 47 to 63 Hz
- ▶ Power dissipation: 100 W

¹ These switches are no longer available as a IBM Machine Type 7048-120 and 7048-270.

Refer to Figure 3-8 on page 51 for a rear view of the SFS 7000P InfiniBand switch

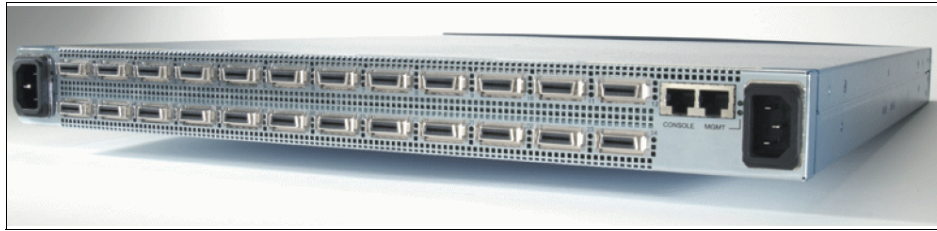


Figure 3-8 Rear view of the SFS 7000P InfiniBand switch

3.8.2 Cisco SFS 7008P InfiniBand Server Switch

The Cisco SFS 7008P InfiniBand Server Switch provides an InfiniBand switch designed to maximize reliability, availability, and serviceability (RAS). With fully redundant, hot-pluggable components and hitless failover, the Cisco SFS 7008P provides a director-class, 96-port switch for building scalable, highly available clusters. The switch comes with 48 ports installed in the base configuration. The switch can be upgraded to 96 ports.

The technical specifications of the SFS 7008P switch are:

- ▶ Width: Rack-mountable mounting in a standard 19-inch EIA rack
- ▶ Depth: 24-inch depth
- ▶ Height: 6 U
- ▶ Weight: 60-110 lbs
- ▶ Air flow: Front-to-rear
- ▶ Temperature operating: 0 degrees C to 42 degrees C
- ▶ Storage: -40 degrees C to 70 degrees C
- ▶ Altitude Operating: 10,000 feet
- ▶ Storage: 40,000 feet
- ▶ Humidity operating: Eight percent to 80 percent non-condensing
- ▶ Storage: Five percent to 90 percent non-condensing
- ▶ Shock Operating: 5 G, 11 ms half-sine wave
- ▶ Storage: 10 G, 11 ms half-sine wave
- ▶ Vibration operating: 0.25 G, 5 to 300 Hz, 15 min.
- ▶ Storage: 0.5 G, 5 to 300 Hz, 15 min.
- ▶ Power 90 to 264 V AC auto-ranging, 47 to 63 Hz
- ▶ Power dissipation: <600 W

Features:

- ▶ Redundant and hot-pluggable FRUs enable no single point of failure.
- ▶ Redundant, synchronized embedded subnet managers.

- ▶ Auto-detection of and auto-recovery from system errors.

Rapid Service Architecture

The Cisco SFS 7008P is designed to eliminate downtime. In large clusters, a key to minimizing downtime is the ability to rapidly service hardware and software upgrades. With fully redundant power, cooling, and control processors, every system field replaceable unit (FRU) is hot-swappable and supports auto-failover. In the Cisco 7008P Rapid Service Architecture, a passive mid-plane design isolates all active electronics on the front of the chassis, with all cables connected to the back. This enables switch modules to be replaced without detaching a single cable, a powerful concept when all ports are cabled and a board needs to be quickly replaced. Since the switch has two controller cards, one card remains inactive in Hot Standby state. In case the active card fails, the second card can do a failover without performing a reboot or reset of the switch or any of its cards.

Refer to Figure 3-9 for a rear view of the SFS 7008P InfiniBand switch

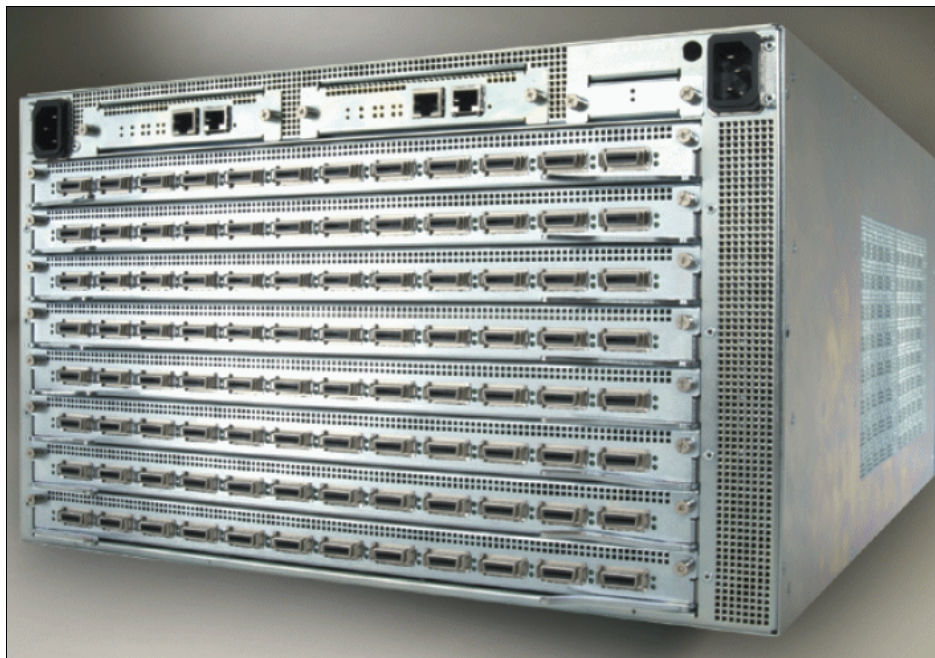


Figure 3-9 Rear view of the SFS 7008P InfiniBand switch

3.8.3 Using the switch: user and passwords

To configure the switch, you can log in through a serial (RS232/TTY) connection with one of the following predefined user/password combinations shown in Table 3-7 on page 53.

Table 3-7 User and password combinations

User	Password
guest	guest
admin	admin
super	super

3.9 InfiniBand cables

The currently supported System p servers and the M/T 7048 Topspin switches use cables in certain lengths to interconnect the servers to the switches. The length of the cables offered by IBM are 1.5 m, 3 m, and 8 m. Longer cables are available from other vendors, but they are *NOT SUPPORTED* for configurations offered by IBM with System p servers. Refer to Table 3-8 for currently supported cable types, length, feature codes, and part numbers.

Table 3-8 InfiniBand cables

InfiniBand cable type	Length	Feature code	Part number
4x	3 m	1835	42R6244
4x	8 m	1836	42R6245
12x to three 4x cables	8 m	1838	42R6248
4x	1.5 m	1839	42R6243

The Feature code 1838 is for System p Server 590 and 595 only.

Identifying IBM supported cables

IBM supported cables can be identified by the label carrying the part number (P/N) and also by the connector color. Figure 3-10 shows the 4x cable connector.



Figure 3-10 4x InfiniBand connector

Figure 3-11 shows the 12x cable connector.



Figure 3-11 12x InfiniBand connector

3.9.1 Cabling with octopus cables

Octopus cables are cables with a 12x connector on one end and three 4x connectors on the other end of the cable. This enables a 12x device to connect to a switch using a group of three switch port connectors. The bandwidth of 30 Gbps per adapter port can be achieved with the octopus cable. The three ports grouped together on the switch side will operate as a single port 12x.

Using static 12x mode

In a group of three ports operating in static 12x mode, one port is assumed to be the control port of the group. This port is the port you will see as active and reports a operating speed of 30 Gbps.

Important: The 12x groups will only work in 12x speed and will not degrade to 4x or 1x speed. Either the link is up and running at 12x speed or the link will be down.

The octopus 4x cable ends are labeled with numbers 16, 17, and 18 and they have a special significance and orientation.

- ▶ The lowest numbered port in a group is the configuration port that determines if the group is running in 12x static mode. This is also the only port that can be enabled/disabled. The other two ports configuration properties are irrelevant.
- ▶ The controlling port in a group can be different for each group. It is either the lowest numbered port or the highest numbered port, so it may be different from the lowest numbered configuration port. It will always be connector 16 of the octopus cable.

For the cable groups, refer to Table 3-9 on page 56 for details on the SFS 7000P switch and refer to Table 3-10 on page 56 for details on the SFS 7008P switch.

Grouping the ports for 12x cables

There are 24 4x switch ports on a SFS 7000P in two rows. They are logically divided into groups of three ports. Refer to Table 3-9 for groups and how to determine which ports are used for static-12x mode.

Table 3-9 Groups for SFS 7000P switch

Groups for static 12x mode on a SFS 7000P											
Group 1			Group 2			Group 3			Group 4		
1	2	3	4	5	6	7	8	9	10	11	12
Group 5			Group 6			Group 7			Group 8		
13	14	15	16	17	18	19	20	21	22	23	24

Grouping the ports for 12x cables

On a SFS 7008P switch, each 4x Line Interface Module (LIM) has 12 ports. These ports are grouped together into three ports each. Refer to Table 3-10 for a look at the groups in each LIM and how to determine the possible configuration.

Table 3-10 Groups for SFS 7008P switch

Groups for static 12x mode on a SFS 7008P											
Group 1			Group 2			Group 3			Group 4		
1	2	3	4	5	6	7	8	9	10	11	12

Cabling 4x connections for octopus cables

The use of the 4x connectors on the octopus cables connected to the switch ports needs a special ordering. As explained already, the 4x connectors are labeled 16, 17, and 18. This numbering is very important and in case the correct sequence is not used, the link will not pass any data across. Refer to Table 3-11 on page 57 for a description of how the sequence must be on a SFS 7000P switch or refer to Table 3-12 on page 57 for the SFS 7008P InfiniBand switch.

Table 3-11 SFS 7000P cable ordering for 12x octopus cable

SFS 7000P cable ordering for 12x octopus cable			
Top ports of switch			
Ports 1-3	Ports 4-6	Ports 7-9	Ports 10-12
16,17,18	16,17,18	16,17,18	16,17,18
Bottom ports of switch			
Ports 13-15	Ports 16-18	Ports 19-21	Ports 22-24
18,17,16	18,17,16	18,17,16	18,17,16

A cabling sequence for the SFS 7008P switch is shown in Table 3-12.

Table 3-12 SFS 7008P cable ordering for 12x octopus cable

SFS 7008P cable ordering for 12x octopus cable				
LIM	Ports 1-3	Ports 4-6	Ports 7-9	Ports 10-12
1	18,17,16	18,17,16	16,17,18	16,17,18
2	16,17,18	16,17,18	18,17,16	18,17,16
3	16,17,18	16,17,18	18,17,16	18,17,16
4	18,17,16	18,17,16	16,17,18	16,17,18
5	18,17,16	18,17,16	16,17,18	16,17,18
6	16,17,18	16,17,18	18,17,16	18,17,16
7	16,17,18	16,17,18	18,17,16	18,17,16
8	18,17,16	18,17,16	16,17,18	16,17,18

3.10 Management server

When Cluster Systems Management (CSM) is used to control, manage, or maintain a cluster of nodes that can run AIX 5L or Linux, a management server (MS) is needed to operate, monitor, and maintain the rest of the cluster.

The management server usually is a stand-alone machine that runs the code but also the use of an LPAR as a management server is possible. However, you should be aware of the limitations when an LPAR is used as a CSM MS.

Considerations: Limitations for an LPAR used as a management server:

- ▶ The CSM management server can be brought down inadvertently by a user on the HMC who deactivates the LPAR. Even if a user does not have access to the CSM management server, a user with access to the HMC can power off the management server or move resources, such as CPU or I/O, from the LPAR.
- ▶ If the firmware needs to be upgraded, the LPAR management server might also go down when the system is quiesced. However, bringing the CEC back up returns the system to normal.
- ▶ There is no manual hardware control of the CSM management server. You must use the HMC for power control of the management server.
- ▶ An LPAR management server cannot have an attached display. This limitation can affect the performance of your CSM GUIs.
- ▶ Do not define an LPAR management server as a managed node.

A cluster that is installed and configured can still function even if the management server goes down. For example, cluster applications can continue to run, and nodes in the cluster can be rebooted. However, tasks including monitoring, automated responses for detecting problems in the cluster, and scheduled file and software updates cannot occur while the management server is down.

Generally, a stand-alone Management Server should be used for both AIX and Linux platforms.

A Linux Management Server cannot be used to control AIX nodes. An AIX Management Server can be used to control both AIX and Linux nodes in a cluster.

For more information about CSM, refer to:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.csm.doc/clusterbooks.html>

3.10.1 A private network DHCP IP configuration versus a static IP configuration

The general recommendation for IBM System p machines managed by a HMC is to use the HMC as DHCP server to ease the administrative tasks for a multi server environment. Also, for low- and midrange servers like 9133-55A and 9117-570, the use of static IP addresses for the FSP is okay.

However, the use of static IP addresses is not possible in the following configurations:

- ▶ When a redundant Flexible Service Processor (FSP) is available (on 9117-570 and 9119-59x).
- ▶ When machines that have a Frame with Bulk Power Controllers (BPC) installed (9118-575, 9119-590, and 9119-595). BPCs need also an IP address, and this can be obtained only through DHCP.

When these machine types are used, DHCP must be enabled either on the HMC (only one!) or on the CSM Management Server.

If you must use DHCP, or if you choose to use CSM and Cluster Ready Hardware Server (CRHS), we recommend that the DHCP server be on the CSM Management Server, and all HMCs must have their DHCP server capability disabled.

When you are in a single HMC environment, the HMC is the DHCP server for the service Ethernet (private) network.

The CSM Management Server must be the DHCP server in Cluster Ready Hardware Server (CRHS) environment in order to recognize the servers, BPAs, HMCs and InfiniBand switches in the cluster.

Note: While we recommend that you have two service networks on different IP subnets to support redundancy in IBM servers, BPCs, and HMCs, the InfiniBand switches only support a single service network (even though some InfiniBand switch models have multiple Ethernet connections).

3.11 IBM Network Manager (IBM NM)

To manage a GX-Adapter based InfiniBand network, you can only use the IBM Network Manager. The PCI based InfiniBand adapters are not available on IBM NM². The IBM Network Manager enables you to manage the InfiniBand network from the Hardware Management Console (HMC).

Use the IBM Network Manager to manage InfiniBand switches, update switch software, view network topology information, and view and modify management properties. You can also view the Network Manager event logs. You must enable the Network Manager from the HMC before you can use it to manage your InfiniBand network. This section explains how to use the IBM Network Manager tasks.

² PCI InfiniBand adapters for System p systems are no longer available from IBM.

More information about IBM Network Manager is available here:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp?topic=/ipha1/usingibmnetmgr.htm>

Enabling and disabling the IBM Network Manager

Before you can use the IBM Network Manager to manage your InfiniBand network switches and system, you must enable it from the HMC. When you enable the IBM Network Manager, it starts IB fabric discovery and begins providing data about the status of your InfiniBand switch network.

Note: If you decide to move the IBM Network Manager functions from one HMC host to another HMC host, you must first disable the IBM Network Manager on the first HMC host before enabling the Switch Management folder.

To enable or disable the IBM Network Manager, complete the following steps:

1. Check that the HMC network is configured properly (as a DHCP server when no CRHS is used; if CRHS is used, the HMC should get a static IP address).

Note: Any HMC user can enable or disable the IBM Network Manager.

2. Check that the Topspin switch is configured as a DHCP client.

Note: You can configure the Cisco InfiniBand switch Ethernet port with a static IP or DHCP IP by using the Serial Port of the switch. However, for the use with IBM Network Manager, this setting must be DHCP.

3. In the Navigation area, expand the **Switch Management** folder.
4. Click **IBM Network Manager**.
5. In the Network Manager menu, select **Enable IBM Network Manager Software**. To disable the IBM Network Manager, select **Disable IBM Network Manager Software**.

The HMC menu is shown in Figure 3-12 on page 61.

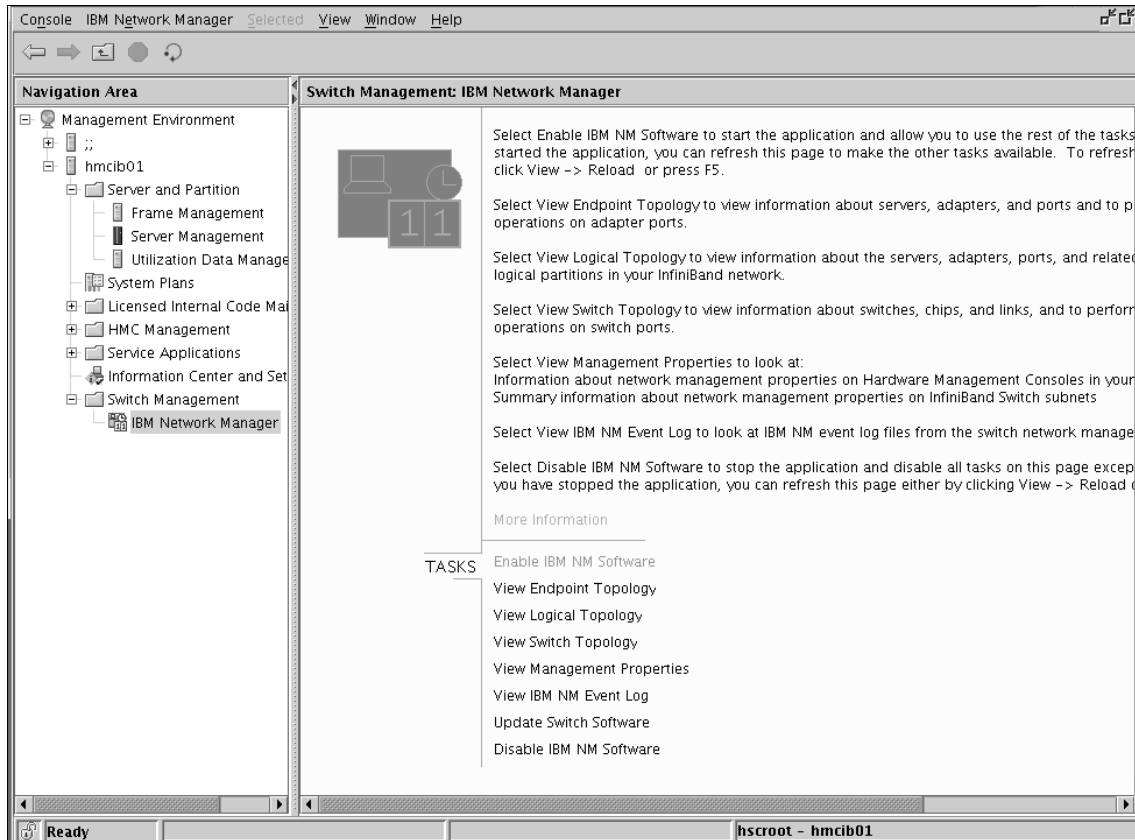


Figure 3-12 IBM NM Main window

Viewing switch topology information in an InfiniBand network

You can view information about the switches in your InfiniBand network, which identifies the physical location and connection activity for all connected switch devices. Any user can view the switch topology information.

To view the switch topology information, follow these steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View Switch Topology**. The HMC displays a table with a list of the switches in the IB network. By default, the information presented shows the switch device name (or identifier), location code, frame information, power on/off status, GUID, port status, and neighbor port (link) information.

To change the columns displayed in the list of information, select **View** → **Show Columns** and select or clear the check box for the columns that you want to designate as the default columns. See Figure 3-13.

Name	Location Code	Connectivity	Node GUID	Port State	Neighbor Location Code	Neighbor Name
Topspin-12...	U7048.120.0200035	Responsive				
	U7048.120.0200035-P1-C1					
	U7048.120.0200035-P1-C1-T1		00:05:AD:00:00:02:13:8C	Active	U787F.001.DPM188L-P1-C21-T1	ib01
	U7048.120.0200035-P1-C1-T2		00:05:AD:00:00:02:13:8C	Active	U787F.001.DPM157P-P1-C21-T1	ib02
	U7048.120.0200035-P1-C1-T3		00:05:AD:00:00:02:13:8C	Active	U787F.001.DPM1894-P1-C21-T1	ib03
	U7048.120.0200035-P1-C1-T4		00:05:AD:00:00:02:13:8C	Active	U787F.001.DPM18DC-P1-C21-T1	ib04
	U7048.120.0200035-P1-C1-T5		00:05:AD:00:00:02:13:8C	Down	-	-

Last Auto-Update: 10/12/06 10:52:38 AM Filter Off

Figure 3-13 View the switch topology

Viewing server topology information in an InfiniBand (IB) network

You can view information about the servers, adapter, and ports in your InfiniBand (IB) network. Any user can view the server topology information.

To view the server topology information, follow these steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View End-Point Topology**. The HMC displays a table with a list of the servers in the InfiniBand cluster. By default, the information presented shows the server name, location code, power on/off status, frame/cage information, adapter status, port status, and neighbor port (link) information. To change the columns displayed in the list of information, select **View** → **Show Columns** and select or clear the check box for the columns that you want to designate as the default columns.

The HMC menu looks like the window shown in Figure 3-14.

Server Name	Location Code	Power	Adapt...	Adapter St...	GID Prefix	Port State	Neighbor Location Code	Neighbor Name
CSM-Linux	9124-720*100195A	Up	0/0					
ib01	9131-52A*10391FG	Up	1/1					
	U787F.001.DPM188L-P1-C21			Functional				
	U787F.001.DPM188L-P1-C21-T1				FE:80:00:00:00:00:02	Active	U7048.120.0200035-P1-C1-T1	T1
	U787F.001.DPM188L-P1-C21-T2				FE:80:00:00:00:00:00	Active	UTS1204XCP4XCP.US20502001...	T2
ib02	9131-52A*10391EG	Up	1/1					
ib03	9131-52A*10391DG	Up	1/1					
ib04	9131-52A*103920G	Up	1/1					

Last Auto-Update: 10/12/06 11:01:57 AM Filter Off

Figure 3-14 IBM NM view endpoint topology

Viewing logical topology information in an InfiniBand (IB) network

You can view information about the servers, adapters, ports, and related status for the logical partitions in your InfiniBand (IB) network. Any user can view the logical topology information.

To view the logical topology information, complete the following steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View Logical Topology**. The HMC displays a table with a list of the servers and partitions in the InfiniBand (IB) configuration. By default, the information presented shows the server name, location code, system or logical partition status, adapter status, type, the GUID, port status, and neighbor port (link) information. To change the columns displayed in the list of information, select **View** → **Show Columns** and select or clear the check box for the columns that you want to designate as the default columns. See Figure 3-15.

Server Name	Location Code	State	Adapters U...	Type	GUID	LPAR Name	Port ID	Port State	GUID Prefix
CSM-Linux	9124-720*100195A	Up	0/0						
ib01	9131-52A*10391FG	Up	1/1						
	U787F.001.DPM188L-P1-C21	Functional		PHCA					
	U787F.001.DPM188L-P1-C21			LHCA	00:02:55:00:50:00:27:3D	ib01			
	U787F.001.DPM188L-P1-C...				00:02:55:00:50:00:27:3D		1	Active	FE:80:00:00:00:00
	U787F.001.DPM188L-P1-C...				00:02:55:00:50:00:27:7D		2	Active	FE:80:00:00:00:00
	U787F.001.DPM188L-P1-C21			LSW	00:02:55:00:50:00:27:80				
	U787F.001.DPM188L-P1-C21			LSW	00:02:55:00:50:00:27:81				
ib02	9131-52A*10391EG	Up	1/1						
ib03	9131-52A*10391DG	Up	1/1						
ib04	9131-52A*103920G	Up	1/1						

Figure 3-15 IBM NM view logical topology

In Figure 3-15, you can also see the configuration shown by the IBM Network Manager of the GX Dual-port adapters attached to an InfiniBand switch. Here are some of the abbreviations to notice:

- ▶ PHCA: Physical Host Channel Adapter
- ▶ LHCA: Logical Host Channel Adapter
- ▶ LSW: Logical Switch

Viewing IBM Network Manager properties

You can view and modify the IBM Network Manager properties and the switch-management properties for your InfiniBand (IB) network.

For example, use this procedure to change the IBM Network Manager default name assigned to the managed switches. Using switch names provides a convenient way to keep track of the switches you are managing (particularly when frame numbers or cage numbers are not readily available).

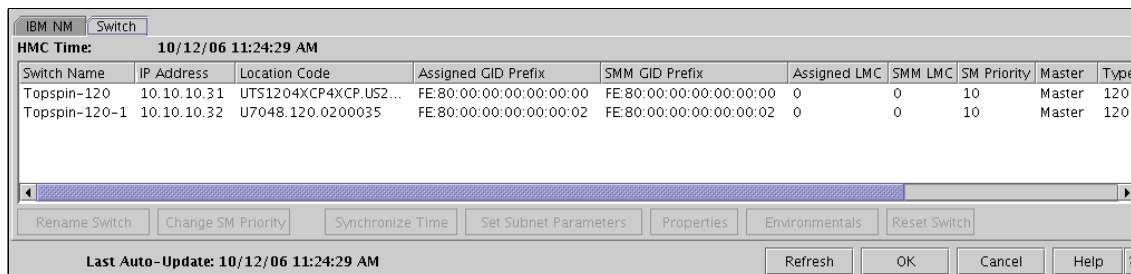
You can also change the switch-management priority (change the subnet manager master and succession order), synchronize the switch local time with the HMC time (NTP time is not currently supported), and view the switch topology.

Note: Any HMC user can view the Network Manager properties.

To view the properties, complete the following steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View Management Properties**.
4. Click the **IBNM** tab to view the IBM Network Manager properties. The HMC displays the HMC host name, the IP address, and version number of the currently enabled IBM Network Manager.
5. Click the **Switch** tab to view and modify information about the switches in your InfiniBand (IB) environment.

Refer to Figure 3-16 for view of the management properties.



Switch Name	IP Address	Location Code	Assigned GID Prefix	SMM GID Prefix	Assigned LMC	SMM LMC	SM Priority	Master	Type
Topspin-120	10.10.10.31	UTS1204XCP4XCP.US2...	FE:80:00:00:00:00:00:00	FE:80:00:00:00:00:00:00	0	0	10	Master	120
Topspin-120-1	10.10.10.32	U7048.120.0200035	FE:80:00:00:00:00:00:02	FE:80:00:00:00:00:00:02	0	0	10	Master	120

Figure 3-16 IBM NM view management properties

Assigning the Global ID prefix (GID) and LID Mask Count (LMC)

The GID prefix is a 64-bit value that identifies the InfiniBand subnet on the InfiniBand network. The GID prefix and the Global Unique Identifier (GUID) for each adapter port forms a Global ID (GID) for the adapter port. The GID is used to address packets that are sent between subnets.

The Local ID (LID) mask Count (LMC) represents the number of bits that are masked in the PID on the subnet. Therefore, the LMC indicates the number of LIDs that belong to each endpoint port on this subnet.

To get redundant connections and enhance the throughput, more than one LID per endpoint port can exist. The subnet manager modifies the switch routing tables to specify a different path to each endpoint port.

The default value for LMC is 0, which is one LID/path per endpoint. For IBM HPC purposes, the value is 2, which gives four LIDs/paths per endpoint port.

To set the LMC value, you have to perform the following steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View Management Properties**.
4. Click the **Switch** tab to view and modify information about the switches in your InfiniBand (IB) environment.
5. Select the appropriate switch and click **Set Subnet Parameters** to change the GID prefix and, if needed, the LMC value

Refer to Figure 3-17 for a view of the Set Subnet Parameters GUI.

Switch Name:	IP Address:	Location Code:	GID Prefix:
Topspin-120	10.10.10.31	UTS1204XCP4XCP.US2050200 145	FE:80:00:00:00:00:00:00

Assign GID Prefix

Assign From Available GID Prefixes:

FE:80:00:00:00:00:00:00
FE:80:00:00:00:00:00:02

Assign New GID Prefixes:

Assign LMC:

0 = Default ▼

?

Figure 3-17 IBM NM Set Subnet Parameters

Viewing the IBM Network Manager log

The IBM Network Manager maintains an event log that you can view to track the IBM Network Manager activities.

Note: Any HMC user can view the event logs.

To view the IBM Network Manager event log, complete the following steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View IBM NM event log**. The HMC displays the contents of the log file.

Refer to Figure 3-18 on page 67 for a view of the event log.

Invoke TI...	App. Name	Board MTMS	Network	Type	Chip	Port	Message
9/29/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start EXPLORE on device ID 10000"
9/29/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start EXPLORE on device ID 10001"
9/29/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start EXPLORE on device ID 10002"
9/29/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start EXPLORE on device ID 10003"
9/29/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start EXPLORE on device ID 10004"
10/2/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start DevAsync PORT_STATE_CHANGE on device ID 10001"
10/2/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start DevAsync PORT_STATE_CHANGE on device ID 10002"
10/2/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start DevAsync PORT_STATE_CHANGE on device ID 10003"
10/2/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start DevAsync PORT_STATE_CHANGE on device ID 10004"
10/2/06...	NM_IBNMFSPD	0000#0000	0	0	0	0	"Start DevAsync CEC_STATE_CHANGE on device ID 10001"

Filter Off

Figure 3-18 IBM NM view event log

Updating the switch software

You can update multiple switches simultaneously for a single software version, or choose a switch and the version of software to apply for the update of a particular switch. You can start another switch installation before a switch update procedure that is currently running has finished; however, if both installations run concurrently and target the same switch, the subsequent installation action might fail.

Note: An Import option allows you to add more software versions from a DVD. Any user can update the switch software.

To update your switch software, complete the following steps:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **Update Switch Software**.
4. Select the switches that you want to update.
5. Select the software version to install for the selected switch or switches.
6. Click **OK** to start the software update.

Note: To uninstall the last update, select the **Return switch to previous software**, as shown in Figure 3-19 on page 68.

For a view of how the HMC menu looks like, refer to Figure 3-19.

Select switches to update:

Switch Name	Location Code	IP Address	Current Version	Previous Version
Topspin-120	UTS1204XCP4XCP.US2050200145	10.10.10.31	TopspinOS-2.7.0/build014	-
Topspin-120-1	U7048.120.0200035	10.10.10.32	TopspinOS-2.7.0/build014	-

Select All Deselect All

Select switch software to install or delete. Click import to add software from DVD.

Switch Management Version	Date
Cisco-SF57000P-TopspinOS-2.7.0-build014.img	Oct 3, 2006

Import... Delete...

Return switch to previous software.

Last Auto-Update: 10/12/06 11:31:43 AM

OK Cancel Help ?

Figure 3-19 IBM NM update switch software

Gathering the IBNM snap data

When you need InfiniBand related switch data for support-related activities, IBM NM gives you the possibility to gather IBNM snap data. This data can either be saved to the local disk on the HMC into `/var/hsc/log` or it can be saved onto a DVD-RAM. The steps for saving the IBNM snap data with the GUI are:

1. In the Navigation area, expand the **Switch Management** folder.
2. Click **IBM Network Manager**.
3. In the Network Manager menu, select **View Management Properties**.
4. Click the **IBNM** tab to view the IBM Network Manager properties. The HMC displays the HMC host name, the IP address, and version number of the currently enabled IBM Network Manager.
5. Click the **Save logs (snap)** button.
6. Either select the **Hard Drive (/var/hsc/log)** or **DVD Drive** as the target location.

Refer to Figure 3-20 on page 69 for view of the GUI.

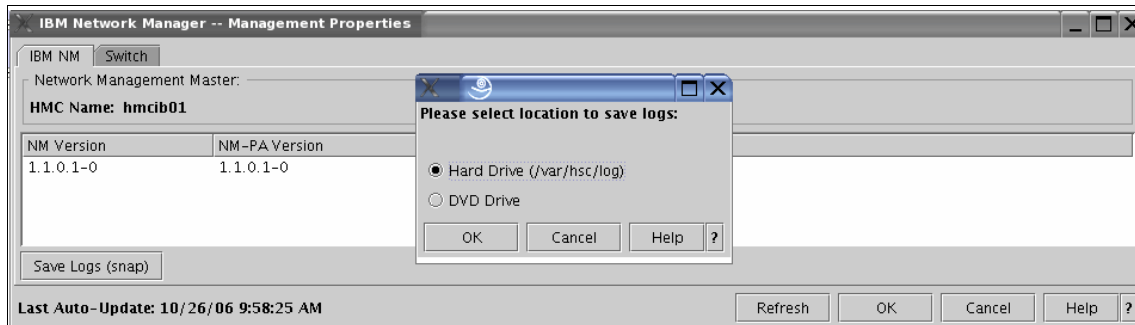


Figure 3-20 IBM NM save logs

In case you select **Hard Drive (/var/hsc/log)**, you can transfer the data from the HMC to a different server with the `sendfile` command. You need to be `root` to transfer the data. To get the normal root access, follow these steps:

1. Log in first as `hscroot`.
2. Verify that the `hscpe` user is available with the `lshmcusr` command; if not, generate the user using the `mkhmcusr -u hscpe -a hmcpe` command and proceed. Refer to Example 3-4 for user verification.

Example 3-4 lshmcusr command

```
hscpe@hmcib01:~> lshmcusr
name=hscroot,taskrole=hmcsuperadmin,description=HMC Super
User,pwage=99999,resourcerole=ALL:
name=hscpe,taskrole=hmcpce,description=HMC
User,pwage=99999,resourcerole=
name=root,taskrole=hmcsuperadmin,description=root,pwage=99999,resourcer
ole=ALL:
```

3. Log in with `ssh hscpe@localhost` as the `hscpe` user.
4. Now perform `su -` to log in as the normal root user with the standard password, which is `passw0rd` with a zero instead of the letter o (if it has not already been changed).

Now you can follow these steps to both generate and transfer the log data to a different server:

1. If you already generated the snap file with the GUI, go to step 3; otherwise, issue the **ibnm.snap** command to generate the log file. Refer to Example 3-5 for the command output.

Example 3-5 ibnm.snap command

```
hscroot@hmcib01:~> ibnm.snap
0
```

2. Verify with the **ls -ltr /var/hsc/log/*.gz** command that you have a log file saved in the /var/hsc/log directory. Refer to Example 3-6 for the command's output.

Example 3-6 ls command verifies log data saved

```
hscroot@hmcib01:~> ls -ltr /var/hsc/log/*.gz
-rw-rw-r-- 1 root ccfw 26860613 2006-10-26 13:34
/var/hsc/log/hmcib01.2006-10-26.13.33.42.snap.tar.gz
```

3. From the command output, you can see the log file with ~26 MB of data. Now you can transfer the data with the **sendfile** command to a different target server in your preferred network. The syntax is **sendfile -f local-file-name -h remote-system -d remote-directory -u remote-user-id**; refer to Example 3-7.

Example 3-7 sendfile command

```
sendfile -f /var/hsc/log/hmcib01.2006-10-26.13.33.42.snap.tar.gz -h
192.168.100.164 -d /tmp -u root
```

4. After verifying the file has been saved on the target server, you can forward it to IBM Support.



Part 2

Implementation

This provides a basic guide for installation and configuration of InfiniBand in an IBM System p p5 server environment. Currently, IBM supports InfiniBand with AIX and SUSE SLES 9 operating systems. We describe the software components that are used to make an IB fabric operational. Operating system, device drivers, clustering products for management, availability, and job scheduling come together to form a solution for today's computing environments.



InfiniBand on AIX 5L

This chapter provides a basic guide for installation and configuration of the IBM System p InfiniBand on AIX, and also covers some administration topics. The following sections are included:

- ▶ InfiniBand on AIX
- ▶ InfiniBand on System p5 running AIX
- ▶ Test cluster layout and description
- ▶ Installation and configuration of the AIX CSM Management server
- ▶ Installation and configuration of AIX nodes
- ▶ GPFS installation and configuration

4.1 InfiniBand on AIX

In order to install the InfiniBand on AIX nodes, it is important to know and understand the software components of the IB device driver on AIX 5L and how these are installed. In addition, hardware and software planning considerations must be understood and followed. This section contains three parts:

- ▶ Hardware requirements: The hardware prerequisites needed for installing the InfiniBand on AIX 5L
- ▶ AIX software requirements: The AIX general software requirements of the InfiniBand on AIX 5L
- ▶ Overview of cluster software components: An overview of cluster software products used to assist in installing and configuring InfiniBand in an AIX cluster.

4.1.1 Hardware requirements

For specific details on each hardware component, see 3.3, “Hardware requirements” on page 35.

4.1.2 AIX software requirements

This section briefly presents the general software prerequisites and the recommended code levels we have also used in our test environment.

AIX 5L V5.3

Customers should at a minimum download AIX 5L Version 5.3 with the 5300-04 Technology Level (APAR IY84006).

APARs can be download from the IBM Electronic Fix Distribution service at:

<http://www-03.ibm.com/servers/eserver/support/unixservers/aixfixes.htm>

4.1.3 Overview of cluster software components

This section contains an overview of cluster software products that are used to help install and configure the InfiniBand network or that use the InfiniBand network. This section includes the following:

- ▶ Cluster Systems Manager (CSM)
- ▶ Reliable Scalable Cluster Technology (RSCT)
- ▶ General Parallel File System (GPFS)

Cluster Systems Manager (CSM)

Fundamentally, since AIX currently only supports IPoIB, the InfiniBand network could be set up without using Cluster Systems Manager (CSM) or Reliable Scalable Cluster Technology (RSCT). For example, you could manually configure NIM to install the LPARs with the necessary filesets and configure the InfiniBand adapters (secondary adapters, IP configuration).

That having been said, we recommend using CSM to install and manage your InfiniBand cluster. CSM (with the infrastructure provided by RSCT) provides a consistent interface for managing both AIX and SLES nodes and has also the flexibility to manage across multiple hardware platforms, various network topologies, and different geographic sites.

CSM is designed to scale up to a large number of servers and to protect performance by providing very efficient monitoring and reduced network traffic. Automatic error detection is another key feature of CSM that can help with problem avoidance, rapid resolution, and recovery.

CSM software provides a distributed system management solution that allows a system administrator to set up and maintain a cluster of nodes that run the AIX or SLES operating system. CSM simplifies cluster administration tasks by providing management from a single point-of-control.

CSM management structure

Management server	The machine that is designated to operate, monitor, and maintain the rest of the cluster.
Install servers	The machines that are used to install (OS and additional SW) the nodes. By default, the management server is the install server.
Managed nodes	Instances of the operating system that you can manage in the cluster.
Managed devices	The non-node devices for which CSM supports power control and remote console access.

CSM benefits

CSM enables system administrators to address a number of system management challenges. Some of the tasks you can perform from the management server include:

- ▶ Installing and updating software on the cluster nodes
- ▶ Running distributed commands across the cluster
- ▶ Synchronizing files across the cluster
- ▶ Run user-provided customization scripts during node installation or updates
- ▶ Monitoring the cluster nodes and devices

- ▶ Controlling cluster hardware
- ▶ Managing node or device groups
- ▶ Running diagnostic tools
- ▶ Configuring additional network adapters

CSM software requirements

When using CSM to install and manage InfiniBand, the minimum required level of CSM is 1.5.1.2 (APAR IY84922).

To verify this level, you can either issue the `instfix -ik IY84922` command, or you can verify that the following filesets are present (`ls1pp -L csm.*`):

- ▶ csm.client 1.5.1.1
- ▶ csm.core 1.5.1.2
- ▶ csm.dsh 1.5.1.2
- ▶ csm.server 1.5.1.2

Note: Further detail on planning for CSM clusters can be found in the *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344, *CSM for AIX 5L and Linux Administration Guide*, SA23-1343, and *CSM for AIX 5L and Linux Command and Technical Reference*, SA23-1345.

Reliable Scalable Cluster Technology (RSCT)

RSCT is a set of software components that work together to provide a comprehensive clustering environment for AIX and Linux (in our case SUSE SLES). RSCT is a component of AIX 5L and is the infrastructure used by a variety of IBM products to provide clusters with improved system availability, scalability, and ease of use. RSCT is well proven to provide clustering infrastructure for CSM, various Distributed Resource Managers, and High Availability Cluster Multi-Processing (HACMP).

Resource Monitoring and Control (RMC)

RMC is a component of Reliable Scalable Cluster Technology (RSCT). The Resource Monitoring and Control (RMC) subsystem is the scalable backbone of RSCT that provides a generalized framework for managing resources within a single system or a cluster. Its generalized framework is used by cluster management tools to monitor, query, modify, and control cluster resources.

RMC provides a single monitoring/management infrastructure for both RSCT peer domains (RPD - where the infrastructure is used by the Configuration resource manager) and management domains (where the infrastructure is used by HMCs and CSM).

RMC can also be used on a single machine, enabling you to monitor/manage the resources of that machine. However, when a group of machines, each running RMC, are clustered together (into management domains/peer domains), the RMC framework allows a process on any node to perform an operation on one or more resources on any other node in the domain.

Management domain

In a management domain (see Figure 4-1), a management node (server) is aware of all nodes it is managing but the nodes themselves know nothing of each other.

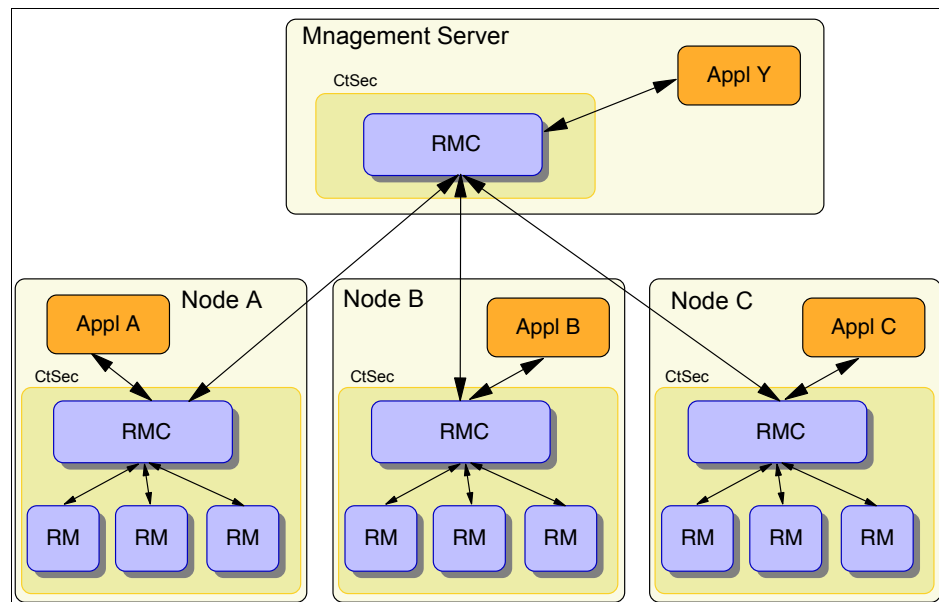


Figure 4-1 Management domain

Managed nodes in a management domain know only about their local resources and the resources located on the Management server (the ones which are advertised by the MS).

Management domains are established and administered by the CSM product.

Key subsystems (Resource managers (RMs)) in a Management Domain include:

- ▶ IBM.CSMAgentRM (only runs on Managed nodes)
- ▶ IBM.DMSRM (only runs on Management Server node)

Since remote actions can be executed between management server nodes and their respective managed nodes, authentication and authorization mechanisms must be provided with along with RMC to ensure that only nodes that have been authorized by the system administrator to participate in the domain can use these remote commands.

The authentication and authorization mechanisms must be configured on each node before it can join a management domain. The following two files are used for authentication, respective authorization:

- ▶ authentication = /var/ct/cfg/ct_has.thl file
- ▶ authorization = /var/ct/cfg/ctrmc.acls file

Filesets needed for management domain

The following filesets are installed by default and are required to establish and administer a management domain:

- | | |
|-------------------|--|
| rsct.core.rmc | - RSCT Resource monitor and control |
| rsct.core.sr | - RSCT Registry |
| rsct.core.utils | - RSCT Utilities |
| rsct.core.errm | - RSCT Event Response Resource Manager |
| rsct.core.auditrm | - RSCT Audit Log Resource Manage |
| rsct.core.fsrn | - RSCT File System Resource Manager |
| rsct.core.gui | - RSCT Graphical User Interface |
| rsct.core.hostrn | - RSCT Host Resource Manage |
| rsct.core.sec | - RSCT Security |

Peer domain

In a peer domain (see Figure 4-2), all nodes are considered equal and any node can monitor and control (or be monitored and controlled) by any other node.

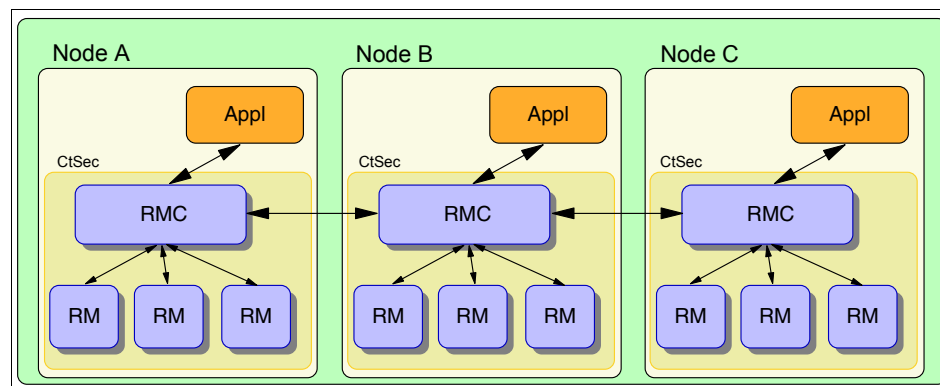


Figure 4-2 RSCT Peer Domain (RPD)

The RMC subsystem and all core Resource Managers are used to manage cluster resources (no CSM involvement in a peer domain).

Key subsystems in a Peer Domain include:

- ▶ IBM.ConfigRM
- ▶ cthats (clustering technology high availability topology services)
- ▶ cthags (clustering technology high availability group services)

A node can belong to several peer domains; however, at one point in time, the node can only belong to *ONLY ONE online* peer domain. Membership to a peer domain is a dynamic feature. A node can be added or removed to/from an existing peer domain.

Since remote commands can be executed between nodes in the peer domain, authentication and authorization mechanisms are provided along with RMC to ensure that only nodes that have been authorized by the system administrator to participate in the domain can use these remote commands. The authentication and authorization mechanisms must be configured on each node before it can join a peer domain.

ctcsd The daemon responsible for administering and enforcing CtSec (clustering technology security) rules.

Authentication The `/var/ct/cfg/ct_has.thl` file.

Authorization The `/var/ct/cfg/ctrmc.acls` file.

Filesets needed for peer domain

The RMC components required for a stand-alone configuration are installed automatically upon the AIX 5L installation by default; therefore, once the AIX 5L installation is complete, RMC will be active and ready to be used. However, even though some resources may be monitored “out-of-the-box”, no resources are controlled (RMC will not trigger any events) until you have configured the system to do so.

For systems that do not require cluster operation, no further filesets are required. If you are creating a peer domain cluster on an AIX 5L Version 5.3 system, you must install the following three filesets in addition to those filesets installed by default:

- ▶ `rsct.basic.rte`
- ▶ `rsct.compat.basic.rte`
- ▶ `rsct.compat.clients.rte`

RMC resources

A Resource is a physical (hardware) or logical (software) entity that provides services to some other component.

- ▶ All resources support a common set of operations (for example, Get/Set Attributes, Online / Offline, and so on).
- ▶ All resources within a class have a common set of Attributes and each resource has a unique set of attribute values.
- ▶ All resources are globally accessible within the cluster.
- ▶ Resource class specific operations can be defined and are called Actions.

A Resource Class is a collection of resources that have similar characteristics.

- ▶ Each resource class supports a common set of operations (for example, Define / Undefine Resource, Get/Set Attributes).
- ▶ Each resource class has a set of Attributes.
- ▶ Each resource class is globally accessible within the cluster.

Resource Managers (RM)

Resource Managers are agents for implementing resource specific functions and therefore rely on the stability of the RMC daemon. (RMC subsystem is known as ctrmc.)

Resource Managers receive requests from RMC, process them, and generate responses.

Currently, all RMs are separate daemons and support any number of resource classes.

RMC daemon automatically starts a subset of RMs at system start, while other RMs are started on demand.

RMC domain and how CSM manages it

CSM uses a management domain, and tells the RMC daemon on the management server about the nodes in the cluster and updates it whenever a node joins or leaves the cluster. This is managed by IBM.DMSRM, which does the following:

- ▶ When DMSRM starts up, it looks at the list of Managed nodes in IBM.ManagedNodes (not the PreManaged, or MinManaged) and gives the list of host names, IP addresses, and the RMC node IDs to the RMC daemon using the daemon API (DAPI). It also gives RMC the node group definitions, and IBM.HWCTRLRM give it the devices and device groups.
- ▶ The RMC daemon stores this domain info in `/var/ct/cfg/ctrmc.mntbl`.

- ▶ When a node is joining the cluster (becoming a Managed node), IBM.CSMAgentRM (through the IBM.ManagementServer class) on the node contacts IBM.DMSRM on the management server and requests to join and pass the RMC node ID, security keys, and so on. IBM.DMSRM stores this information and also gives the new entry to the RMC daemon using the DAPI.
- ▶ When a node is removed from the cluster (using **rmnode**), IBM.DMSRM informs the RMC daemon.
- ▶ The process works in a very similar way for managed devices.

In this way, CSM makes sure that RMC always has an accurate list of the other RMC daemons in the cluster it can talk to. This topology is used by RMC to provide remote operations to any/all of the nodes from the management server, including:

- ▶ Subscription to events on any/all of the nodes (and forwarding the event back to the management server when an event occurs). This is used by IBM.ERRM on the management server to enable the conditions to watch for events on the nodes.
- ▶ Querying resources that are on any/all of the nodes (for example, you can query the network interfaces on all of the nodes using the **lsrsrc -a IBM.NetworkInterface** command)
- ▶ Modify resources or run actions on any/all of the nodes

RSCT software requirements

When using RSCT to install and manage InfiniBand, *at a minimum* you are required to be at RSCT 2.4.5.2 (APAR IY84920).

This can be checked by either issuing the **instfix -ik IY84920** command, or by verifying that the following filesets have been applied (default filesets installed for management domain):

- ▶ devices.chrp.base.ServiceRM 1.3.0.45
- ▶ rsct.core.rmc 2.4.5.1
- ▶ rsct.core.utils 2.4.5.2
- ▶ rsct.core.auditrm 2.4.5.0
- ▶ rsct.core.fsrn 2.4.5.0
- ▶ rsct.core.gui 2.4.5.1
- ▶ rsct.core.hostrm 2.4.5.1
- ▶ rsct.core.sec 2.4.5.1
- ▶ rsct.core.errm 2.4.5.1
- ▶ rsct.core.sr 2.4.5.0
- ▶ rsct.core.lprm 2.4.5.0
- ▶ rsct.core.sensorrm 2.4.5.0

In addition, for a peer domain, the following RSCT filesets are needed:

- ▶ rsct.basic.rte 2.4.5.2
- ▶ rsct.compat.basic.rte 2.4.5.0
- ▶ rsct.compat.clients.rte 2.4.5.0

For other products, such as HACMP, PSSP, and IBM Tivoli System Automation for Multiplatforms, the following RSCT filesets are needed:

- ▶ rsct.opt.storagerm 2.4.5.2
- ▶ rsct.basic.hacmp 2.4.5.2
- ▶ rsct.basic.sp 2.4.5.0
- ▶ rsct.compat.basic.hacmp 2.4.5.0
- ▶ rsct.compat.basic.sp 2.4.5.0
- ▶ rsct.compat.clients.hacmp 2.4.5.0
- ▶ rsct.compat.clients.sp 2.4.5.0

Note: Further details on planning for RSCT can be found in *RSCT: Administration Guide SA22-7889*, *RSCT: Messages GA22-7891*, *RSCT for AIX 5L: Technical Reference SA22-7890*, and *RSCT for Linux Technical Reference SA22-7893*.

CSM and RSCT InfiniBand support enhancements

CSM currently provides support for gathering adapter information from cluster nodes and for automating the configuration of secondary adapters. CSM users use the **getadapters** command to gather network adapter information from the nodes. This information may be related to the adapter used to install a node or it may be related to other (secondary) adapters.

In the current release, the **getadapters** distributed shell (dsh) method will be enhanced to gather IB information. Once the system has been installed, the IB adapter(s) will be automatically defined along with at least one network interface for each adapter. The **getadapters dsh** method will be able to gather IB information by running the **lsattr** command for the interface on the node(s).

With this enhancement, CSM also supports the automatic configuration of IB secondary adapters during the update of the nodes the **updatenode -c** command.

CSM also provides support for monitoring the status of network interfaces. This is done using the **csmdat** command. This command uses an RSCT resource manager to gather the status information. This resource manager already includes InfiniBand interface information.

For example, to check if the node is reachable and the status of the interfaces, you could run the command shown in Example 4-1.

Example 4-1 Checking adapters' status in CSM

```
msib01: /> csmstat -s Status,Network-Interfaces -n aib04
-----
Hostname          Status   Network-Interfaces
-----
aib04             on      en3-Online ib0-Online ib1-Online
```

RSCT uses the resource manager IBM.ConfigRM to gather network adapter resource information for all defined supported adapters and stores the information in the resource class IBM.NetworkInterface. The `lsrsrc` command can be used on the node to view the contents of the resource class IBM.NetworkInterface.

For example, to view the network resource “ib0” and its attributes, you could run the command shown in Example 4-2.

Example 4-2 Checking IB interface persistent attributes

```
aib04: /> lsrsrc -s 'Name == "ib0"' -l IBM.NetworkInterface
Resource Persistent Attributes for IBM.NetworkInterface
resource 1:
    Name           = "ib0"
    DeviceName     = ""
    IPAddress      = "192.168.8.164"
    SubnetMask     = "255.255.255.0"
    Subnet         = "192.168.8.0"
    CommGroup      = ""
    HeartbeatActive = 0
    Aliases        = {}
    DeviceSubType  = 0
    LogicalID      = 0
    NetworkID      = 0
    NetworkID64    = 0
    PortID         = 0
    HardwareAddress = ""
    DevicePathName = ""
    ActivePeerDomain = ""
    NodeNameList   = {"aib04"}
```

General Parallel File System (GPFS)

IBM General Parallel File System (GPFS) provides file system services to parallel and serial applications running on multiple nodes. GPFS allows parallel applications simultaneous access to the same files, or different files, from any node that has the GPFS file system mounted while managing a high level of control over all file system operations. GPFS is particularly appropriate in an environment where the aggregate peak need for data bandwidth exceeds the capability of a distributed file system server.

GPFS allows users shared file access within a single GPFS cluster and across multiple GPFS clusters. A GPFS cluster consists of:

- ▶ AIX 5L nodes, Linux nodes, or a combination of AIX 5L and Linux nodes. A node may be:
 - An individual operating system image on a single computer within a cluster.
 - A partition (LPAR) running an individual copy of an operating system (AIX or Linux). Some System p5 and pSeries machines allow multiple system partitions, each of which is considered to be a node within the GPFS cluster.
- ▶ Network shared disks (NSDs) created and maintained by the NSD component of GPFS.
 - All disks utilized by GPFS must first be given a globally accessible NSD name.
 - The GPFS NSD component provides a method for cluster-wide disk naming and access.

On Linux machines running GPFS, you may give an NSD name to:

 - Physical disks
 - Logical partitions of a disk
 - Representations of physical disks (such as LUNs)

On AIX machines running GPFS, you may give an NSD name to:

 - Physical disks
 - Virtual shared disks
 - Representations of physical disks (such as LUNs)
- ▶ A shared network for GPFS communications allowing a single network view of the configuration. A single network, a LAN or a switch, is used for GPFS communication, including the NSD communication.

Basic GPFS components

GPFS is a clustered file system defined over a number of nodes. On each node in the cluster, GPFS consists of:

- ▶ Administration commands
Most GPFS administration tasks can be performed from any node running GPFS.
- ▶ A kernel extension (file system device driver)
The GPFS kernel extension provides the interfaces to the operating system vnode and virtual file system (VFS) interfaces for adding a file system.
- ▶ A multithreaded daemon
The GPFS daemon performs all I/O and buffer management for GPFS.
- ▶ For nodes in your cluster operating with the Linux operating system and the GPFS open source portability layer
For Linux nodes running GPFS, you must build custom portability modules based on your particular hardware platform and Linux distribution to enable communication between the Linux kernel and the GPFS kernel modules.

GPFS and network communication

Within the GPFS cluster, you may specify different networks for GPFS daemon communication and for GPFS administration command usage. The default communications protocol for communication between nodes in a GPFS nodeset is TCP/IP. This is the only currently supported communication protocol in a GPFS cluster environment.

The interconnect for GPFS daemon-to-daemon and administration command communication depends upon the types of nodes in your cluster. The latest supported interconnects and environments can be found at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.pdf

GPFS and InfiniBand on AIX 5L

According to the previously mentioned Web site, currently (at the time of this writing and for the current GPFS version) InfiniBand is a supported interconnect for an environment of AIX 5L V5.3, GPFS V3.1, and IP over InfiniBand (IPoIB).

Software requirements

The minimum level of GPFS is 3.1.0.4 (with APAR IY86878 applied).

To check the GPFS level, you can either use the `instfix -ik IY86878` command, or by verifying that the following filesets have been applied (using the `lspp -L gpfs.*` command):

- ▶ gpfs.base 3.1.0.4
- ▶ gpfs.msg.en_US 3.1.0.3
- ▶ gpfs.docs.data 3.1.0.1

Note: Further information about GPFS can be found in *GPFS V3.1 Advanced Administration Guide*, SC23-5182-00, *GPFS V3.1 Administration and Programming Reference*, SA23-2221-00, and *GPFS V3.1 Concepts, Planning, and Installation Guide*, GA76-0413-00.

4.2 InfiniBand on System p5 running AIX

This section briefly reviews the following:

- ▶ Implementation of InfiniBand architecture (IBA) on System p5
- ▶ IP over InfiniBand (IPoIB) implementation
- ▶ AIX InfiniBand filesets and components

4.2.1 Implementation of InfiniBand architecture (IBA) on System p5

This section briefly discusses the specific IBA implementation on the IBM System p5 servers. We review key IBA concepts and terms and mention how each is currently implemented.

LPARs (processor nodes)

One or more processors and their associated memory and IO devices. The LPARs interface to the IBA fabric through one or more Host Channel Adapters (HCA) and each HCA has two IBA ports.

The HCA is used to refer to the adapter hardware or the device driver.

The current implementation for IBM pseries P5 only allows one HCA adapter per LPAR and the two available ports cannot connect to the same InfiniBand switch. In addition, only 32 LPARs can be connected to any given IB switch.

Port

This is an interface that connects an IBA device to an IBA link that is bi-directional. The ports are the endpoints to which the data is sent.

Link

This is a high-speed bi-directional connection between two ports on two IBA devices.

IB subnet

This is a set of ports and associated links with a common Subnet Prefix and managed by a common Subnet Manager (SM).

The SM resides on the IB switch. At startup time, the SM discovers and configures all the devices on the subnet and assigns each port with a Local ID and it will periodically check for changes in the subnet's topology.

Verb layer

This is the IBA specification that describes what the API must support without rigorously defining the API and is defined for application and kernel users of IB. It has two main responsibilities:

- ▶ Identifying, initializing, and controlling the HCA device.
 - HCA verbs: Open, Query, Modify and Close
- ▶ Performing the management of hardware resources, such as queue pairs (QPs), completion queues (CQs), and memory registration resources for performing translation and protection management.
 - QP verbs: Create, Modify, Query, Destroy and Get Special QP
 - CQ verbs: Create, Query, Resize, Destroy and Poll
 - Address Handle (AH) verbs: Create, Modify, Query and Destroy
 - Memory Region (MR) verbs: Register, Query, Reregister and Deregister
 - Multicast (MC) verbs: Create, Modify, Query and Destroy
 - Protection Domain (PD) verbs: Allocate and Deallocate

Queue pair (QP)

A queue pair is a message transport engine implemented by the HCA (hardware) that is bi-directional. It is used to dedicate adapter resources for the user/application that allows the (kernel) device driver to be bypassed for data send and receive. It consists of a *Send Queue (SQ)* and a *Receive Queue (RQ)* that are used to pass buffers (messages) in Work Queue Elements (WQEs) to the HCA.

Posting WQEs containing buffers with data to the SQ begins the transmit process and posting WQEs containing buffers to the RQ provides the buffers for the HCA to pass data to the user application later.

QPs can have seven different states:

RST	Reset
INIT	Initialized
RTR	Ready to Receive
RTS	Ready to Send
SQD	Send Queue Drain
SQErr	Send Queue Error
ERR	Error

These QP states are the responsibility of the user application and are managed by using the modify QP verb.

There are two special QPs that are used for management:

- ▶ QP0 is used to manage subnet management packets and is managed by subnet management agent (SMA). The SMA handles all packets received on QP0 and responds to the actions specified in the subnet management packet (SMP).
- ▶ QP1 is general-services interface (GSI) is used by the InfiniBand Connection Manager (ICM) and handles general management packets.

Currently, the HCA on the pseries P5 can implement up to 16.384 (16 K) QPs and each is capable of sending and receiving messages from one or more QPs in remote Host Channel Adapters.

Queue pair number (QPN)

Queue pair numbers identify the QP that the HCA assigns to the IF. The QPN will be received after the creation of the QP. This QPN will be saved by the IF to be used for ARP responses (QP number and GID are values that need to be delivered in the ARP response).

Event queue (EQ)

Indicates any event on the port.

Completion queue (CQ)

This queue is used to pass information to the user application from the HCA.

IBA communication stack

Figure 4-3 illustrates how all the various IBA concepts fit together from a hardware perspective.

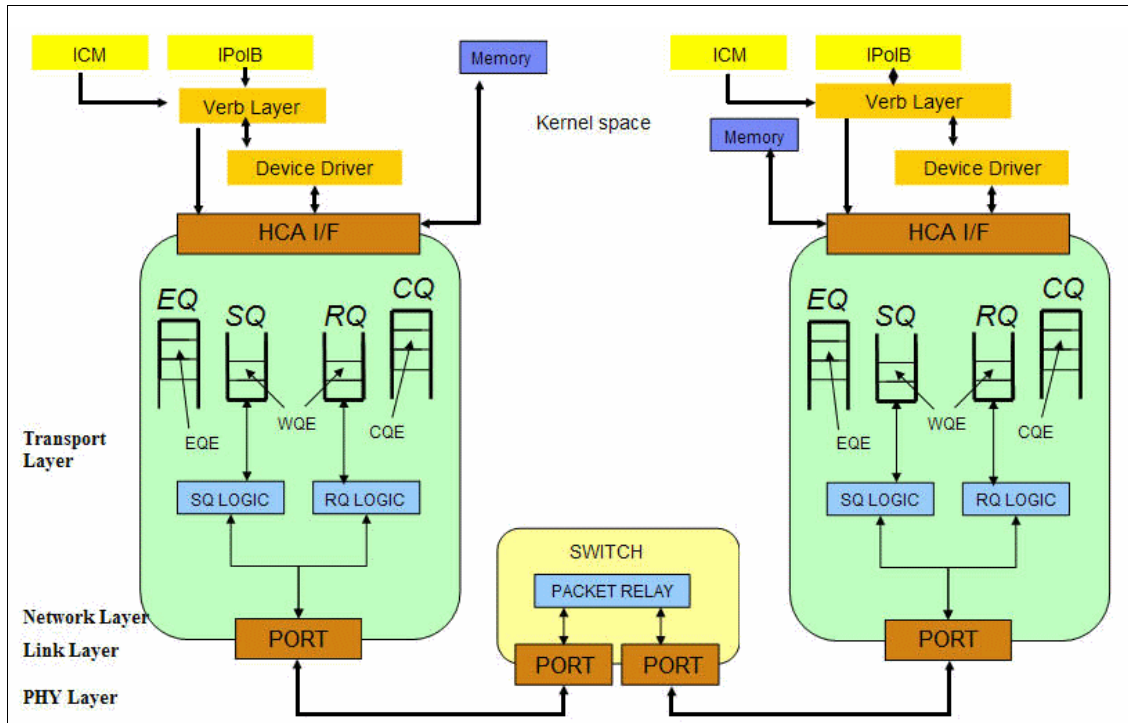


Figure 4-3 IBA communication stack

InfiniBand Connection Manager (ICM)

A driver that performs connection management functions for establishing connections to the different end nodes within the IB subnet.

4.2.2 IP over InfiniBand (IPoIB) implementation

As mentioned in Chapter 3, “InfiniBand hardware overview and implementation” on page 29, currently (at the time of this writing) the only IB communication protocol currently supported in AIX is IPoIB. In the future there are plans to support Message Passing Interface (MPI), Sockets Direct Protocol (SDP), and Network File Server (NFS) using the Kernel Direct Access Programming Library (kDAPL).

The IP over InfiniBand (IPoIB) implementation allows transporting TCP/IP packets over InfiniBand infrastructure.

IPoIB interface

The transport is accomplished by encapsulating IP packets over IB packets using a kernel interface layer (if_ib) between IP and the verb layer.

The current interface has the following limitations:

- ▶ Packets cannot cross IB subnets.
- ▶ The IPoIB interface does not support IPv6.
- ▶ Only UD (unreliable datagram) packets are supported.

Link layer Information

An InfiniBand packet over the UD mode needs certain link layer information determined before an IP packet may be transmitted across the IPoIB link.

The following information is needed:

LID (local identifier) The LID is always needed. A packet always includes the local route header (LRH) that is targeted at the remote node's LID.

GID (global identifier)

The GID is not needed when exchanging information within an IB subnet although it may be included in any packet. It is used to map to the LID by the subnet manager (SM).

QPN (queue pair number)

Every unicast UD communication is always directed to a particular queue pair (QP) at the peer.

Q_Key

A Q_Key is associated with each unreliable datagram QPN. The received packets must contain a Q_Key that matches the QP's Q_Key to be accepted.

P_Key

A successful communication between two IB nodes using UD mode can occur only if the two nodes have compatible P_Keys.

How does IPoIB work

Basically, we need to translate an IP address into a GID and destination QP. Then we need to translate the GID into an LID (local LID of the remote destination). We will then have enough information to encapsulate the IP packet into an IB packet and transmit across the IPoIB link.

Figure 4-4 describes the IPoIB encapsulation:

1. The if_ib interface tries to resolve destination host details (destination QP/GID) by looking into the local IB ARP table.
2. The IB ARP reply returns the destination QP/GID.
3. The ICM does a “Find path” to look up the destination LID.
4. The Path Reply returns the destination LID.
5. We have enough information to transmit the packet across the IPoIB link.

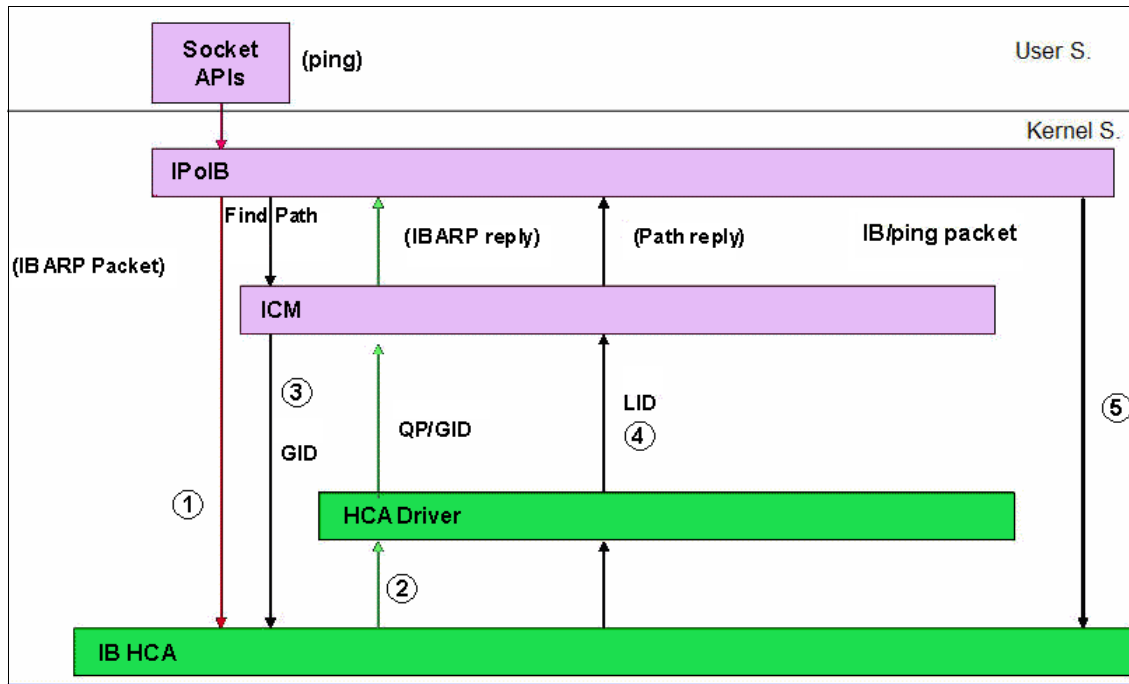


Figure 4-4 How does IPoIB work

4.2.3 AIX InfiniBand filesets and components

Filesets

In AIX 5L V5.3 TL05, there are two filesets that need to be installed to install and configure the GX InfiniBand adapter and related devices. The filesets needed are:

- ▶ devices.chrp.IBM.lhca.rte
- ▶ devices.common.IBM.ib.rte

The `devices.chrp.IBM.lhca.rte` fileset contains the files (configuration, libraries, and commands) to support the GX IB adapter. It contains the kernel extension `gxibdd` and is used to create and manage the IB Host Channel Adapter Device (HCAD) `iba0`.

The `devices.common.IBM.ib.rte` fileset contains the files (configuration, libraries, and commands) to support the InfiniBand Connection Manager (ICM) and the IPoIB kernel interface. This fileset contains the kernel extension `icmdd` for the ICM and `if_ib` for IPoIB.

- ▶ ICM is defined by the “`icm`” device (a shown by the `lsdev` command).
- ▶ IPoIB interfaces are assigned to specific ports and are defined by “`ib0`” and “`ib1`”.

To check the existence of an HCA adapter, we use the `lscfg` command shown in Example 4-3.

Example 4-3 lscfg -v output

```
aib04@/>lscfg -v | grep iba0
  iba0 U787F.001.DPM18DC-P1-C21 InfiniBand Host Channel Adapter
```

To view the availability states of the various devices, we use the `lsdev` command shown in Example 4-4.

Example 4-4 Checking IB adapter status

```
aib04@/>lsdev -C | grep iba
iba0      Defined      IP over Infiniband Network Interface
ib1       Defined      IP over Infiniband Network Interface
iba0      Available    Infiniband Host Channel Adapter
icm       Available    Infiniband Communication Manager
```

In Example 4-4, we can see that the `icm` and the IB adapter are available and that the interfaces `ib0` and `ib1` are defined, but not yet available.

Configuration process

Here we discuss the configuration process.

IB HCA is configured

After a node with an IB HCA is installed or when the IB HCA adapter is first configured (using the `cfgmgr -v` command), the `lsdev -v | grep iba` output will look similar to Example 4-5.

Example 4-5 IB adapter configured

```
aib01@/>lsdev -C | grep iba
ib0      Defined          IP over Infiniband Network Interface
iba0     Available         Infiniband Host Channel Adapter
```

Just like an Ethernet adapter, the HCA adapter is configured and the first interface “ib0” is in a defined state. In addition, the QP0 is created and ready to handle management packets.

ICM is configured

The next step is to configure the ICM device. This can either be done with `smitty icm` or with the command `mkdev -c management -s infiniband -t icm`. After either configuration method is run, the `lsdev -v | grep iba` command shows an output similar to Example 4-6.

Example 4-6 Creating the icm device

```
aib01@/> mkdev -c management -s infiniband -t icm
icm Available
aib01@/>lsdev -C | grep iba
ib0      Defined          IP over Infiniband Network Interface
iba0     Available         Infiniband Host Channel Adapter
icm     Available     Infiniband Communication Manager
```

The icm device is now available. This is the InfiniBand Communication Manager that handles the general service packets from the General Services Interface (GSI) or the special Queue Pair (QP) QP1 (the QP1 has not been yet created in this step).

IB interface(s) is configured

The final step is to configure the IB interfaces (that is, ib0, ib1, and so on). This can either be done using SMIT (`smitty inet`) or with the following command:

```
mkiba -a <ip address> -i ib0 -p 1 -P -1 -A iba0 -S “up” -m “<netmask>”
```

After either configuration method is run, the `lsdev -v | grep iba` command shows an output similar to Example 4-7.

Example 4-7 Configuring ibX interfaces

```
aib01@/>mkiba -a 192.168.8.161 -i ib0 -p 1 -P -1 -A iba0 -S "up" -m 255.255.255.0
ib0 changed
```

```
aib01@/>mkiba -a 192.168.9.161 -i ib1 -p 2 -P -1 -A iba0 -S "up" -m 255.255.255.0
ib1 changed
```

```
aib01@/>lsdev -C | grep iba
```

ib0	Available	IP over Infiniband Network Interface
ib1	Available	IP over Infiniband Network Interface
iba0	Available	Infiniband Host Channel Adapter
icm	Available	Infiniband Communication Manager

When `ib0` is configured, the kernel interface (`if_ib`) will issue an open to the HCA adapter (device) that is handled by the ICM. At this point the Queue Pair (QP) `QP1` is created in both ports of the adapter and general services packets can be handled by the ICM.

Additional details on how to configure the IB devices and the commands used to check the status of the IB adapter and ports (`ibstat`) on AIX are covered in 4.5, “Installation and configuration of AIX nodes” on page 104.

4.3 Test cluster layout and description

This section describes our test environment and the steps we have taken to configure it.

4.3.1 Planning for installation

We have made some assumptions for the planning and installation of our pSeries CSM cluster with an InfiniBand network. We assumed that:

- ▶ The Hardware Management Console (HMC) is installed and operational, and any managed systems are already defined to it.
- ▶ The management and cluster networks are in place and functional.
- ▶ The InfiniBand network is in place and the HCAs are assigned to the LPARs.
- ▶ All the hardware and software requirements outlined in the *Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344 have been met or have been planned for.

Note: Further detail on planning for CSM clusters can be found in the *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344.

4.3.2 Our environment

We created an AIX cluster with a AIX CSM Management server (MS)/ Install server (ISA) and four AIX compute nodes (LPARs). The installation and configuration of the AIX CSM MS is discussed in 4.4, “Installation and configuration of the AIX CSM Management server” on page 97 and the installation and configuration of the AIX compute nodes is discussed in 4.5, “Installation and configuration of AIX nodes” on page 104.

The following tables present the planning data (worksheet) for service network and public network for the sample AIX cluster:

- Table 4-1 presents the HMC network configuration.

Table 4-1 Planning worksheet for HMC

Network type	Service network	Public network
Host name	hmcib01p	hmcib01p
Ethernet adapter	eth0	eth1
IP address	10.10.10.28	192.168.100.28
Netmask	255.255.255.0	255.255.255.0

- Table 4-2 presents the network configuration for the CSM management Server.

Table 4-2 Planning Worksheet for Management server

Model: pSeries 630		
OS: AIX 5LV 5.3 TL05		
Serve as DHCP and Installation server? Yes		
Network type	Service network	Public network
Host name	msib01p	msib01
Ethernet adapter	en0	en1
IP address	10.10.10.29	192.168.100.29
Netmask	255.255.255.0	255.255.255.0

- Table 4-3 presents the AIX LPAR network configuration.

Table 4-3 Planning Worksheet for AIX LPARs

Model: <u>System p52A</u>								
OS: <u>AIX 5L V5.3 ML05</u>								
Service network domain: <u>10.10.10</u>								
Service network port: <u>HMC port-1</u>								
Public network domain: <u>192.168.100</u>								
Public network Ethernet adapter: <u>eth3</u>								
IPoIB network domains: <u>Port 1: 192.168.8 Netmask: 255.255.255.0</u>								
<u>Port 2: 192.168.9 Netmask: 255.255.255.0</u>								
Node #	Service network Service Processor		Public network en0		InfiniBand network			
					Port 1		Port 2	
	Host name	IP	Host name	IP	Host name	IP	Host name	IP
1	aib01p	161	aib01	161	aib01sw1	161	aib01sw2	161
2	aib02p	162	aib02	162	aib02sw1	162	aib02sw2	162
3	aib03p	163	aib03	163	aib03sw1	163	aib03sw2	163
4	aib04p	163	aib04	163	aib04sw1	163	aib04sw2	163

Figure 4-5 on page 97 presents a diagram of our test environment. The LPARs are in fact System p5 52A in full system partition.

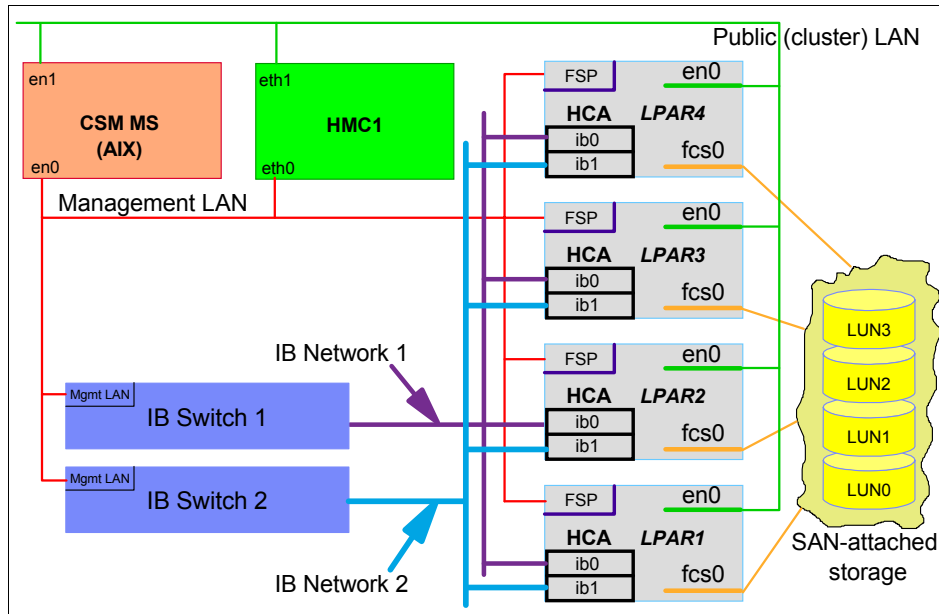


Figure 4-5 ITSO test configuration

4.4 Installation and configuration of the AIX CSM Management server

In this section, we describe how we have set up our System p p5 Cluster Systems Management (CSM) AIX management server in our lab environment. This cluster was used to demonstrate cluster system management with an emphasis on installation and configuration of InfiniBand adapters and network.

4.4.1 Installation of AIX

We performed a standard install of AIX using the AIX 5L V5.3 CDs that already incorporated Technology Level 04. However, to apply the required APAR IY84006, we also installed Technology Level 05. Our management server is a stand-alone IBM eServer pSeries 630 (64-bit IBM POWER4™ processor) with four CPUs and 16 GB of RAM.

After the install, we then performed basic AIX customization to complete the installation of the management server. This included:

- ▶ Set up a network configuration (TCP/IP) suitable for our lab environment. This included a host name and TCP/IP addresses. We have set up two networks: one public (cluster) network and one management (private) network (see Figure 4-5 on page 97).
- ▶ Set the root password.
- ▶ Set up the `/etc/hosts` file from the network matrixes in 4.3.2, “Our environment” on page 95.

4.4.2 Installing the AIX 5L management server

After the successful installation of AIX 5L V5.3 at TL05, we then follow the steps outlined in Chapter 5, “Installing a CSM for AIX 5L management server”, in *BM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344. This section will briefly review those steps we used while installing our AIX MS.

Management server planning

During our planning phase, we decided to take advantage of the CSM administration model and store all the information about our nodes on the management server. This enables us to manipulate the node information in one place and then either update or rebuild the node quickly if needed.

Node information is classified into several categories:

- ▶ Basic information
 - Host name, operating system (OS) version, hardware control connectivity, and so on. This type of information is stored in the `IBM.ManagedNode RSCT` resource class and can be viewed by issuing the `lsrsrc IBM.ManagedNode` command.
- ▶ How the node should network boot is stored in:
 - `/csminstall`: What drivers and kernel should be used
 - `/etc/dhcpd.conf` and `/tftp` (IBM System x: PXE boot instructions)
- ▶ What should be installed on the node is stored in `/csminstall`:
 - What RPMs or NIM resources
 - Customization scripts
 - Secondary adapter information (this is in a file pointed to by the `AdapterStanzaFile` node attribute)

- ▶ Software updates or additional apps to be installed (Linux) are stored in:
 - /csminstall/Linux/<distro>/<version>/<arch>/<service-level>/install¹
 - or
 - /csminstall/Linux/<distro>/<version>/<arch>/<service-level>/updates
- ▶ Configuration of the node

Name resolution, services, automount file systems, conf files, and so on are stored in /cfmroot.
- ▶ Security keys/information are stored in ~root/.ssh and /var/ct/cfg.
- ▶ Node monitoring

Most conditions and responses are stored on the MS.

Since all our nodes are AIX nodes, we did not implement any of the Linux only steps. For details on those steps, see *CSM for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344.

Overview of CSM MS configuration steps

- ▶ We have added the required directories to the PATH and MANPATH variable into the ~/.profile for root user on CSM MS:


```
export PATH=$PATH:/opt/csm/bin:.
export MANPATH=$MANPATH:/opt/csm/man:.
```
- ▶ Next, we created a volume group called csmvg and the /csminstall file system with the following command:


```
crfs -v jfs2 -g csmvg -m /csminstall -a size=1024M -a bf=true
```
- ▶ We installed and verified the required open source software on the CSM MS. Most of the open source packages were installed from the AIX Toolbox for Linux Applications CD, and a few were gathered off of the various Web sites mentioned in Chapter 5, “Installing a CSM for AIX 5L management server”, in *BM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344. We used the `rpm -qa` command to verify that the required RPMs were installed:
 - openssl-0.9.7g-1
 - conserver-8.1.7-2
 - expect-5.34-8
 - tcl-8.3.3-8

¹ <distro> stands for distribution (Red Hat EL, SUSE SLES), <version> is the distribution version, for example, SUSE SLES 9, <arch> stands for the CPU architecture type (Intel® 32b, PowerPC® 64b etc.), and <service-level> represents the actual service pack of the distribution.

- tk-8.3.3-8
- openCIMOM-0.8-1
- ▶ We have downloaded and installed the required CSM and RSCT APARs (IY84922 and IY84920) from the IBM Electronic Fix Distribution service at:

<http://www-03.ibm.com/servers/eserver/support/unixservers/aixfixes.html>

This brought the CSM filesets to V1.5.1.2 and the RSCT filesets to V2.4.5.2.
- ▶ We next accepted the CSM license and copied the CSM files into the /csminstall subdirectories with the following command:

`csminstall -L -c`
- ▶ We decided to set up Cluster Ready Hardware Server (CRHS) and followed the procedures in Appendix B, “Cluster Ready Hardware Server” on page 273. We have also stored the hardware control point user IDs and passwords, as shown in the last step of Appendix B, “Cluster Ready Hardware Server” on page 273.
- ▶ Finally, we verified that the management server has been installed correctly and is ready for use by issuing the `ibm.csm.ms probe` command:

`probemgr -p ibm.csm.ms -l 0`

Note: AIX 5L V5.3 Technology Level 05 updates CSM to Version 1.5.1.1 and RSCT to Version 2.4.5.0. For more information about installing and maintaining AIX, refer to *AIX 5L Version 5.3 Installation and Migration*, SC23-4389.

4.4.3 NIM configuration

This section is not meant to be a detailed discussion of how to install and set up Network Installation Manager (NIM); rather, it gives an overview of the processes we followed in setting up our NIM server. See *AIX 5L Version 5.3 Installation and Migration*, SC23-4389, and the book *NIM from A to Z in AIX 5L*, SG24-7296 for more in-depth detail about NIM master and client installation.

There are two ways to install NIM onto the management server. We can set up the NIM environment manually or we can choose to issue the `nim_master_setup` command (which is, in fact, a wrapper provided by CSM to configure NIM master).

We first had to install the NIM master and Shared Product Object Tree (SPOT) filesets, as the CSM management server will become a NIM master and SPOT server for our cluster. We installed the NIM master file sets from the base media and applied maintenance level (`smitty update_a11`). Application of maintenance

levels is very important to bring the NIM master to the same maintenance level as the operating system. Failure to do this may cause failure of the NIM master. The file sets to install and update are:

- ▶ bos.sysmgt.nim.master
- ▶ bos.sysmgt.nim.spot

For our cluster, we decided to use the `nim_master_setup` command for setting up NIM because it is a very simple but effective way to install a NIM master. It automates the various tasks that are required in setting up a NIM server and is very good for those who may have little experience in setting up a NIM master. Depending on your experience with NIM, you may chose to set it according to your preferences.

Since we decided to use the `nim_master_setup` directory (`/export/nim`), we issued the simple `nim_master_setup -B` command (see Example 4-8). We used the `-B`, option which indicates that we do not want a `mksysb` resource created.

Example 4-8 Output of the `nim_master_setup -B`

```
msib01:/tmp>nim_master_setup -B

##### NIM master setup #####
#
# During script execution, lpp_source and spot resource creation times
# may vary. To view the install log at any time during nim_master_setup,
# run the command: tail -f /var/adm/ras/nim.setup in a separate screen.
#
#####
Device location is /dev/cd0
Resources will be defined on volume group rootvg.
Resources will exist in filesystem /csminstall.
Checking for backup software...already installed.
Checking /tmp space requirement...done
Installing NIM master fileset...already installed.
Defining NIM master...0513-071 The nimesis Subsystem has been added.
0513-071 The nimd Subsystem has been added.
0513-059 The nimesis Subsystem has been started. Subsystem PID is 24604.
0042-001 nim: processing error encountered on "master":
    0042-023 m_chnet: "default " is not a valid NIM routing stanza

Located volume group rootvg.
Creating /tftpboot filesystem...done
Creating bosinst_data resource 5300-05bid_ow...done

Please insert AIX 5.3 product media in device /dev/cd0
```

```
If the location for AIX 5.3 product media differs from
device /dev/cd0, supply the absolute path BEFORE pressing the ENTER key.
=> /backupfs/aix53_lppsource
Checking / space requirement...done
Creating lpp_source resource 530lpp_res...done
Checking / space requirement...done
Checking /tftpboot space requirement...done
Creating spot resource 530spot_res...done
Creating resource group basic_res_grp...done
The following resources now exist:
boot                resources      boot
nim_script          resources      nim_script
5300-05bid_ow       resources      bosinst_data
530lpp_res          resources      lpp_source
530spot_res         resources      spot
basic_res_grp       groups        res_group
NIM master setup is complete - enjoy!
```

Note: If you are using or plan to use CSM's High Availability Management Server (HA MS) in your cluster, see Chapter 4, "High Availability Management Server (HA MS)" in *CSM for AIX 5L and Linux: Administration Guide*, SA23-1343, which covers the NIM setup process.

If you decide to install NIM in the same file system (/csminstall) that was set up for the CSM MS, you can use the command shown in Example 4-9.

Example 4-9 Using the CSM wrapper to set up a NIM master

```
msib01:/> nim_master_setup -a file_system=/csminstall -a
volume_group=csmvg -B
```

This command automatically performs basic NIM tasks. It installs NIM file sets, configures NIM, creates basic resources, and creates a resource group with the resources that are created. We used the -B option, which indicates that we do not want a mksysb resource created.

4.4.4 Updating NIM

After the setup of NIM master has completed, it is still necessary to update AIX, CSM, and RSCT in the lpp_source and the spot resources, from the level that has been available on the CD to the current maintenance level.

For this, we copy the updates gathered from update CDs and downloaded them from the internet to our lpp_source directory /csminstall/lpp_source/530lpp_res. Then we perform a NIM check operation on the lpp_source resource to merge the new files into NIM:

```
msib01:/tmp>nim -o check 530lpp_res
```

Then we update the spot and rebuild the network boot images from this lpp_source by performing a check operation on the spot:

```
msib01:/tmp>nim -o check 530spot_res
```

Now the spot and lpp_source resources are up to date and ready for installing AIX nodes.

4.4.5 Verify InfiniBand filesets

Before proceeding to 4.5, “Installation and configuration of AIX nodes” on page 104, verify that the required InfiniBand filesets are in the NIM lpp_source. You can use the steps presented in Example 4-10.

Example 4-10 Checking NIM lpp_source and spot resources for IB filesets

```
msib01@/>lsnim
master          machines        master
boot           resources       boot
nim_script      resources       nim_script
master_net      networks        ent
5300-05bid_ow   resources       bosinst_data
530lpp_res      resources       lpp_source
530spot_res     resources       spot
basic_res_grp   groups          res_group
aib01           machines        standalone
aib02           machines        standalone
aib03           machines        standalone
aib04           machines        standalone

msib01@/>lsnim -l 530lpp_res
530lpp_res:
  class          = resources
  type           = lpp_source
```

```

arch          = power
Rstate       = ready for use
prev_state   = verification is being performed
location     = /csminstall/lpp_source/530lpp_res
simages      = yes
alloc_count  = 0
server       = master

```

```

msib01@/> nim -o showres '530lpp_res' | grep -e
"devices.chrp.IBM.lhca.rte" -e "devices.common.IBM.ib.rte"
  devices.chrp.IBM.lhca.rte  5.3.0.0          I  N  usr,root
  devices.chrp.IBM.lhca.rte  5.3.0.40         I  N  usr,root
  devices.chrp.IBM.lhca.rte  5.3.0.50         S  N  usr
  devices.common.IBM.ib.rte  5.3.0.0          I  N  usr,root
  devices.common.IBM.ib.rte  5.3.0.40         I  N  usr,root
  devices.common.IBM.ib.rte  5.3.0.50         S  N  usr,root

```

In the previous Example 4-10 on page 103, we use the `lsnim` and `nim` commands to check for the required infiniband filesets in the NIM `lpp_source` '530lpp_res'.

If the filesets are missing, the filesets must be added in the `lpp_source` directory and installed into the spot.

4.5 Installation and configuration of AIX nodes

This section will show the steps necessary to install AIX on the cluster nodes. For a complete description, see the *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344 and the *AIX 5L Version 5.3 Installation and Migration*, SC23-4389. Configuration of InfiniBand adapters and the verification of the setup will be further explained.

4.5.1 Pre-installation tasks

At this point, CSM is set up and NIM has its basic resources configured. However, CSM and NIM do not know about the nodes yet. Before we can install nodes with AIX, the nodes must be defined to CSM.

In order to define the nodes into our cluster, we collect information about all LPARs from the Hardware Management Console (HMC) using the `lshwinfo` command:

```
msib01:/tmp>lshwinfo -p hmc -c hmcib01p -o /tmp/ibnodes
```

This command collects the data of LPARs known to our HMC hmcib01p and stores the data in the file /tmp/ibnodes, as shown in Example 4-11.

Example 4-11 Node information collected from HMC

```
msib01:/tmp>cat /tmp/ibnodes
#Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParID::HWType
::HWModel::HWSerialNum::DeviceType::UUID
no_hostname::hmc::hmcib01p::ib01::2::9131::52A::10391FG:::
no_hostname::hmc::hmcib01p::ib02::2::9131::52A::10391EG:::
no_hostname::hmc::hmcib01p::ib03::2::9131::52A::10391DG:::
no_hostname::hmc::hmcib01p::ib04::2::9131::52A::103920G:::
```

Note: The HMC is dealing with hardware. From a hardware perspective, AIX is installed on a logical partition (LPAR). From a CSM perspective, it does not matter whether AIX is running on an LPAR or on a physical node. Therefore, in CSM terms, everything is called a node.

Note: The names ib01, ib02, ib03, and ib04 are the names of the LPARs defined on the HMC. We edit this file and add the AIX host names that our LPARs will have.

Now we can define our four nodes to CSM with the **definnode** command using the /tmp/ibnodes file as input:

```
msib01:/tmp>definnode -M /tmp/ibnodes InstallOSName=AIX
Defining CSM Nodes:
4 nodes have been defined successfully.
```

Check the defined nodes using the **lsnode** command:

```
msib01:/tmp>lsnode
aib01
aib02
aib03
aib04
```

Detailed information for a node is shown with the `-l` parameter of `lsnode`. See Example 4-12.

Example 4-12 output of the `lsnode -l` command

```
msib01:/tmp>lsnode -l aib01
  Hostname = aib01
  AdapterStanzaFile =
  AllowManageRequest = 0 (no)
  CSMVersion =
  ChangedAttributes = {}
  ConfigChanged = 0 (no)
  ConsoleMethod = hmc
  ConsolePortNum =
  ConsoleSerialDevice =
  ConsoleSerialSpeed = 9600
  ConsoleServerName = hmcib01p
  ConsoleServerNumber =
  FWSvcProc =
  FWSysBIOS =
  HWControlNodeId = ib01
  HWControlPoint = hmcib01p
  HWModel = 52A
  HWSerialNum = 10391FG
  HWType = 9131
  InstallAdapterDuplex =
  InstallAdapterGateway =
  InstallAdapterHostname =
  InstallAdapterMacaddr =
  InstallAdapterName =
  InstallAdapterNetmask =
  InstallAdapterSpeed =
  InstallAdapterType =
  InstallCSMVersion =
  InstallDisk =
  InstallDiskType =
  InstallDistributionName =
  InstallDistributionVersion = 5.3.0
  InstallKernelVersion =
  InstallMethod =
  InstallOSName = AIX
  InstallPkgArchitecture =
  InstallServer =
  InstallServerAKBNode =
  InstallServiceLevel =
```

```
InstallStatus = PreManaged
InstallTemplate =
LICManagedSystemLevel =
LICPowerSubsystemLevel =
LParID = 2
LastCFMUpdateTime =
ManagementServer = 192.168.100.29
Mode = PreManaged
NFSServer =
Name = aib01
NodeNameList = {msib01}
PhysicalLocation =
PowerMethod = hmc
PowerStatus = 1 (on)
Properties =
Status = 127 (unknown)
UUID =
UpdatenodeFailed = 0 (false)
UserComment =
```

Note: The Mode attribute of the node is PreManaged and there is no network information yet. After the node is successfully added to the cluster, this attribute will change to Managed.

If there are multiple nodes to be installed, it is helpful to define CSM *nodegroups* that contain the nodes to be installed. All nodes in a nodegroup can be installed simultaneously. CSM nodegroups are defined using the **nodegrp** command:

```
msib01:/tmp>nodegrp -a aib01,aib02,aib03,aib04 aibnodes
```

This defines a nodegroup containing the four nodes of our test cluster. A nodegroup can also be defined using the **smitty csm_mkgroup** fast path, as shown in Example 4-13.

Example 4-13 SMIT csm_mkgroup

Define CSM Node Groups

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

	[Entry Fields]	
New Node Group Name	[aibnodes]	
List of Nodes	[aib01,aib02,aib03,aib> +	
OR		
Select String	[]	
OR		
File containing list of groups to define	[]	
Group names represent generic node names	no	+
Display Verbose Messages?	no	+

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

All CSM commands allow you to specify either a node or a list of nodes through the **-n** option or a nodegroup or a list of nodegroups via the **-N** option.

The installation process needs to power on and off the nodes. Therefore the hardware control should be checked. Use the **rpower** command to see if hardware control is configured and operational, as shown in Example 4-14.

Example 4-14 Checking if HW control is operational

```
msib01:/tmp>rpower -N aibnodes -l query
aib04 on   LCDs are blank
aib03 on   LCDs are blank
aib02 on   LCDs are blank
aib01 on   LCDs are blank
```

This shows that all four LPARS are powered on and that the LCDs of the nodes are blank, that is, no serious error condition is present.

Alternately, you can use the SMIT fast path `csminst_rp`, as shown in Example 4-15.

Example 4-15 SMIT csminst_rp

Power Control

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

	[Entry Fields]	
* Action	[query]	+
CEC Startup Mode	[]	+
Node List	[]	+
Node Groups	[aibnodes]	+
Device List	[]	+
Device Groups	[]	+
Use All Nodes?	no	+
Use All Nodes and Devices?	no	+
Display LCD Values?	no	+
Wait for command to complete?	no	+
Display Verbose Messages?	no	+
Sort output by hostname?	no	+

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

4.5.2 Get network adapter information

CSM cluster node installation is done through an Ethernet network. Therefore CSM must know about Ethernet adapters in the cluster nodes. The **getadapters** command can automatically collect this information and store it in a stanza file for further processing. Example 4-16 shows the output of the **getadapters** command.

Example 4-16 Output of the getadapters command

```
msib01:/tmp>getadapters -a -D -t ent -s auto -d auto -z /tmp/myadapters
Can not use dsh - No nodes in Managed or MinManaged mode.
Acquiring adapter information from Open Firmware for node aib04.
Acquiring adapter information from Open Firmware for node aib01.
Acquiring adapter information from Open Firmware for node aib03.
Acquiring adapter information from Open Firmware for node aib02.

#Node::adapter_type::interface_name::MAC_address::location::media_speed::adapter_dupl
ex::install_server::install_gateway::ping_status::machine_type::netaddr::subnet_mask

aib04::ent::::00145E961A23::U787F.001.DPM18DC-P1-T6::auto::auto::::0.0.0.0::ok::insta
ll::::

aib01::ent::::00145E963595::U787F.001.DPM18BL-P1-T6::auto::auto::::0.0.0.0::ok::insta
ll::::

aib03::ent::::00145E96342B::U787F.001.DPM1894-P1-T6::auto::auto::::0.0.0.0::ok::insta
ll::::

aib02::ent::::00145E96346B::U787F.001.DPM157P-P1-T6::auto::auto::::0.0.0.0::ok::insta
ll::::

#---Stanza Summary-----
#   Date: Mon Oct  9 17:25:42 EDT 2006
#   Stanzas Added: 4
#---End Of Summary-----
```

Alternately, you can use SMIT with the **esm_getadapters** fast path, as shown in Example 4-17.

Example 4-17 SMIT esm_getadapters

Get Adapter Information

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

[TOP]	[Entry Fields]	
Adapter Type to Acquire	[ent]	+
Discover the first adapter that can be used to ping the management server?	no	+
Network Speed	[auto]	+
Network Duplex	[auto]	+
Server Hostname or IP address	[]	
Gateway IP address	[]	
Discover all the adapters that can be used to ping the management server?	yes	+
Gather information for every adapter?	no	+
Limit execution to one node at a time per HMC? (HMC-based nodes only)	no	+
Method to use to obtain MAC address	[]	+
Adapter Information Filename	[]	
OR		
Perform Adapter Collection on All Nodes?	no	+
OR		
Node List	[aib01]	+
OR		
Node Groups	[]	+
OR		
A file containing a list of node names	[]	
* Save the install adapter information in the CSM database?	no	+
Name of stanza file to create or update	[/tmp/mystanzafile]	
Output filename	[]	
Save the name of the secondary adapter stanza file.	[]	

Display Verbose Messages?	no	+	
[BOTTOM]			
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

The **getadapters** command execution may take a while since it has to boot the nodes to the open firmware prompt. A detailed log file of the **getadapters** process can be found in the `/var/log/csm/getadapters/` directory.

We verify the stanza file has the correct entry for network installation. Edit this file if necessary.

Note: The **getadapters** command only collects information about adapters that can be used for the network installation process, so it will not show virtual adapters or InfiniBand adapters.

Once verified, we can transfer the adapter stanza file to the CSM database using the following command:

```
msib01:/tmp>getadapters -w -f /tmp/myadapters
```

This will write the stanzafile to the CSM database. Alternately, you can use the SMIT `csm_getadapters` fast path.

4.5.3 Further configuration

At this point any customization that goes beyond basic AIX installation should be set up.

Configurations that are done by setting up configuration files without executing special setup commands can be done very easily using the Configuration File Manager (CFM) of CSM. An example for this is the `/etc/hosts` file. In addition to the IP addresses and host names of our cluster nodes and our management server, we input the IP addresses and names of the IPoIB interfaces in our cluster that will be configured only after the installation of AIX is complete.

Configuration steps that require the execution of configuration commands should be done in a shell script. This script can be executed as part of the actual installation process described below. An example would be turning Simultaneous Multi Threading (SMT) off.

For an extended discussion on configuration file manager (CFM), see the *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344. For setting up configuration scripts that are executed by NIM, see the manual *AIX 5L Version 5.3 Installation and Migration*, SC23-4389.

4.5.4 Preparing NIM for nodes (clients) installation

To verify that the NIM server is set up correctly, use the `lsnim` command (shown in Example 4-18). This command must be run on the NIM installation server. In our case, this is the CSM management server (see 4.4.3, “NIM configuration” on page 100).

Example 4-18 Checking available NIM installation resources

```
msib01:/tmp>lsnim
master          machines      master
boot           resources    boot
nim_script     resources    nim_script
master_net     networks     ent
5300-04bid_ow  resources    bosinst_data
530lpp_res     resources    lpp_source
530spot_res    resources    spot
aib01          machines    standalone
aib03          machines    standalone
aib02          machines    standalone
aib04          machines    standalone
basic_res_grp  groups      res_group
osprereboot    resources    script
```

Now that the nodes and installation adapters are defined to CSM, and the basic NIM configuration is complete, the cluster nodes to be installed have to be defined to NIM. This is done by using the `csm2nimnodes` command on the management server:

```
msib01:/tmp>csm2nimnodes -N aibnodes
```

This command does not send the output to the screen. A detailed log can be found in `/var/csm/log/csm2nimnodes.log`.

Next, the CSM customization scripts have to be defined as NIM resources. This is done using the `csmsetupnim` command:

```
msib01:/tmp>csmsetupnim -N aibnodes
```

Detailed output for this command is logged in `/var/csm/log/csmsetupnim.log`.

In the last step to install the nodes, NIM must be conditioned to install the nodes. This is done by using the **nim** command on the NIM server:

```
msib01:/tmp>nim -o bos_inst -a source=rte -a boot_client=no -a
group=basic_res_grp aib02
```

This will set up NIM to perform an AIX installation on node aib02. To verify the correct resource allocation, use the **lsnim** command shown in Example 4-19.

Example 4-19 Checking NIM resource allocation for node installation

```
msib01:/tmp>lsnim -l aib02
aib02:
  class           = machines
  type            = standalone
  connect         = shell
  platform        = chrp
  netboot_kernel  = mp
  if1             = master_net aib02 00145E96346B ent
  net_settings1   = 100 full
  cable_type1     = N/A
  Cstate          = BOS installation has been enabled
  prev_state      = ready for a NIM operation
  Mstate          = currently running
  boot            = boot
  bosinst_data    = 5300-04bid_ow
  lpp_source      = 530lpp_res
  nim_script      = nim_script
  script          = osprereboot
  spot            = 530spot_res
  cpuid           = 00031FA4D700
  control         = master
```

The Cstate of this node shows that BOS installation has been enabled.

Finally, the installation is started with the **netboot** command:

```
msib01:/tmp>netboot -n aib02
```

This command returns to a shell prompt after a while, when the installation has started, but long before the installation process has finished. Progress of the installation can be observed using a *read-only* console (the **rconsole** command) launched on the CSM MS:

```
msib01:/tmp>rconsole -r -t -n aib02
```

4.5.5 Verification of the AIX installation

In order to verify that the nodes have been successfully added to the cluster, we use the **lsnode** command and check for the Mode attribute. Selected attributes can be specified to reduce the verbosity of the command output (see Example 4-20).

Example 4-20 Checking node manageability status

```
msib01:/tmp>lsnode -N aibnodes -a Mode
aib01:  Managed
aib02:  Managed
aib03:  Managed
aib04:  Managed
```

The long listing of the **lsnode** command output for an installed node is shown in Example 4-21. Compare this to the output of the same **lsnode** command when the node was in *premanaged* mode (see Example 4-12 on page 106).

Example 4-21 Output of lsnode -l for an installed node

```
msib01:/>lsnode -l aib01
Hostname = aib01
AdapterStanzaFile = /tmp/aib01.ibstanzafile
AllowManageRequest = 0 (no)
CSMVersion = 1.5.1.0
ChangedAttributes = {LastCFMUpdateTime}
ConfigChanged = 0 (no)
ConsoleMethod = hmc
ConsolePortNum =
ConsoleSerialDevice =
ConsoleSerialSpeed = 9600
ConsoleServerName = hmcib01p
ConsoleServerNumber =
FWSvcProc =
FWSysBIOS =
HWControlNodeId = ib01
HWControlPoint = hmcib01p
HWModel = 52A
HWSerialNum = 10391FG
HWType = 9131
InstallAdapterDuplex = auto
InstallAdapterGateway =
InstallAdapterHostname = aib01
InstallAdapterMacaddr = 00145E963595
InstallAdapterName =
```

```
InstallAdapterNetmask = 255.255.255.0
InstallAdapterSpeed = auto
InstallAdapterType = ent
InstallCSMVersion =
InstallDisk =
InstallDiskType =
InstallDistributionName =
InstallDistributionVersion = 5.3.0
InstallKernelVersion =
InstallMethod = nim
InstallOSName = AIX
InstallPkgArchitecture =
InstallServer =
InstallServerAKBNode =
InstallServiceLevel =
InstallStatus = Managed
InstallTemplate =
LICManagedSystemLevel =
LICPowerSubsystemLevel =
LParID = 2
LastCFMUpdateTime = 1161203200
ManagementServer = 192.168.100.29
Mode = Managed
NFSServer =
Name = aib01
NodeNameList = {msib01}
PhysicalLocation =
PowerMethod = hmc
PowerStatus = 1 (on)
Properties =
Status = 1 (alive)
UUID =
UpdatenodeFailed = 0 (false)
UserComment =
```

We can get a status report on the newly installed nodes using the **csmdat** command, as shown in Example 4-22. Here we can see that the system (CEC) power is on, the LPAR is started, and that the Ethernet network interface en3 is online.

Example 4-22 csmstat output showing all nodes installed and operational

```
msib01:/>csmstat
-----
Hostname           HWControlPoint   Status   PowerStatus   Network-Interfaces
-----
aib01              hmcib01p         on       on            en3-Online
aib02              hmcib01p         on       on            en3-Online
aib03              hmcib01p         on       on            en3-Online
aib04              hmcib01p         on       on            en3-Online
```

At this point we are able to log into the installed client node from the management server through **telnet**, **rsh**, or on a *read/write rconsole* and verify the correct installation of the operating system and perform additional configuration tasks.

Note: CSM does not set up initial root passwords. Initial passwords can be distributed with CSM configuration file manager (CFM).

4.5.6 Configuring InfiniBand adapters on AIX nodes

InfiniBand adapters can be configured by explicitly using AIX commands or it can be done by CSM using a customization script.

AIX commands

Configuring InfiniBand adapters with AIX commands takes two steps: defining the InfiniBand Communication Manager (ICM) and configuring the IP over InfiniBand (IPoIB) network devices.

- Configure the ICM with the **mkdev** command:

```
aib01:/tmp>mkdev -c management -s infiniband -t icm
```

Alternately you can use the SMIT fast path `mk_icm`, as shown in Example 4-23.

Example 4-23 SMITTY `mk_icm`

Add an Infiniband Communication Manager

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

	[Entry Fields]	
Infiniband Communication Manager Device Name	icm	
Minimum Request Retries	[1]	+#
Maximum Request Retries	[7]	+#
Minimum Response Time (msec)	[100]	+#
Maximum Response Time (msec)	[4300]	+#
Maximum Number of HCA's	[256]	+#
Maximum Number of Users	[65000]	+#
Maximum Number of Work Requests	[65000]	+#
Maximum Number of Service ID's	[1000]	+#
Maximum Number of Connections	[65000]	+#
Maximum Number of Records Per Request	[64]	+#
Maximum Queued Exception Notifications Per User	[1000]	+#
Number of MAD buffers per HCA	[64]	+#

- ▶ Next we can create the IP devices. Since we have an InfiniBand adapter with two ports, we can define an IP device on each port using the `mkiba` command:

```
aib01:/tmp>mkiba -a 192.168.8.161 -i ib0 -p 1 -A iba0 -S up -P -1 -m
255.255.255.0
aib01:/tmp>mkiba -a 192.168.9.161 -i ib1 -p 2 -A iba0 -S up -P -1 -m
255.255.255.0
```

These commands define IPoIB devices `ib0` and `ib1`. They are created on port 1 and port 2, respectively. Both use adapter `iba0`. Both devices use the default partition key `-1`.

Note: In AIX 5L V5.3 ML 5, there is no IPv6 support for IP over InfiniBand.

CSM

As previously mentioned, the `getadapters` support for gathering InfiniBand adapter information is only provided through the use of the `dsh` collection method. In addition, the configuration of InfiniBand adapters through CSM on AIX systems is only supported by using the `updatenode` command with the `-c` option. IB adapter configuration is not supported as part of a NIM node AIX installation.

In order to configure InfiniBand adapters on AIX nodes, perform the following steps:

- ▶ We run the **getadapters** command in the dsh mode to get the adapter information from the running node and create the stanza file. For example, to get all the InfiniBand adapter information from the cluster node aib01 and create a stanza file called /tmp/myIBstanzafile, we issue the command shown in Example 4-24.

Example 4-24 Using the getadapters command

```
msib01:/tmp>getadapters -m dsh -t iba -z /tmp/myIBstanzafile -n aib01
Acquiring adapter information using dsh.
```

```
#Node::adapter_type::interface_name::ib_adapter::ib_port::mtu::netaddr:
:subnet_mask::p_key::srq_size
```

```
aib01::iba::ib0:::1::2044:::-1::1
```

```
#---Stanza Summary-----
#   Date: Tue Oct 10 11:30:52 EDT 2006
#   Stanzas Added: 1
#---End Of Summary-----
```

- ▶ The contents of the definition stanza looks similar to the one shown in Example 4-25. The adapter type (iba) is specified.

Example 4-25 IB adapter stanza created by the getadapters command

```
msib01:/tmp>cat /tmp/myIBstanzafile
###CSM_ADAPTERS_STANZA_FILE###--do not remove this line
#---Stanza Summary-----
#   Date: Tue Oct 10 11:30:52 EDT 2006
#   Stanzas Added: 1
#---End Of Summary-----
```

```
aib01:
machine_type=secondary
adapter_type=iba
interface_name=ib0
netaddr=
subnet_mask=
ib_adapter=
ib_port=
mtu=
p_key=
```

srq_size=

If the IB device has been (unsuccessfully) defined before, some of the empty values in Example 4-25 on page 119 may have defined values (the fields are not empty).

Note: The attribute of srq_size in the stanza file must always be empty or the following commands will fail.

- ▶ Next, edit the stanza file. In our setup, we have one InfiniBand adapter (iba0) per node. Therefore, the **getadapters** command returns one stanza per node. Each adapter has two ports. We want to define one InfiniBand device on each of the ports. Therefore, we need two stanzas per node. We duplicate the stanza and set the necessary attributes for machine_type, adapter_type, interface_name, netaddr, subnet_mask, ib_adapter, and ib_port. After we have edited the stanza file, it looks like the one shown in Example 4-26.

Example 4-26 Edited InfiniBand adapter stanza file

```
###CSM_ADAPTERS_STANZA_FILE###--do not remove this line
aib01:
machine_type=secondary
adapter_type=iba
interface_name=ib0
netaddr=192.168.8.161
subnet_mask=255.255.255.0
ib_adapter=iba0
ib_port=1
mtu=
p_key=
srq_size =

aib01:
machine_type=secondary
adapter_type=iba
interface_name=ib1
netaddr=192.168.9.161
subnet_mask=255.255.255.0
ib_adapter=iba0
ib_port=2
mtu=
p_key=
srq_size =
```

- ▶ We have saved the contents of the stanza file in the node definition. Then we run the **getadapters** command with the **-W** option:

```
msib01:/tmp>getadapters -W -f /tmp/myIBstanzafile
```

CSM saves the contents of the AdapterStanzaFile in the node definitions.
CSM also checks the stanza file for errors.

- ▶ Finally, we configure the adapters. We run the **updatenode** command with the **-c** option, as shown in Example 4-27.

Example 4-27 Configuring the InfiniBand adapter on node aib01

```
msib01:/tmp>updatenode -c -n aib01
```

Now running updatenode.client on the nodes.

```
aib01: Setting Management Server to 192.168.100.29.
```

```
aib01: Node Install - Successful.
```

```
aib01: Output log is being written to "/var/log/csm/install.log".
```

Now running CFM to push /cfmroot files the nodes.

```
aib01: No CFM files were transferred to this machine.
```

4.5.7 Verification of the InfiniBand configuration

The successful configuration of the InfiniBand adapter can be tested most easily with the **ping** command, as shown in Example 4-28.

Note: We have set up an /etc/hosts file with IP names, as shown in Table 4-3 on page 96.

Example 4-28 Checking the IP connectivity on IB adapters

```
aib04:/>ping aib01sw1
```

```
PING aib01sw1 (192.168.8.161): 56 data bytes
```

```
64 bytes from 192.168.8.161: icmp_seq=0 ttl=255 time=1 ms
```

```
64 bytes from 192.168.8.161: icmp_seq=1 ttl=255 time=0 ms
```

```
^C
```

```
--- aib01sw1 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0/0/1 ms
```

All nodes can be verified with the **csmdat** command, as shown in Example 4-29.

Example 4-29 Output of the csmdat command with InfiniBand up and running

```
msib01:/tmp>csmdat
```

Hostname	HWControlPoint	Status	PowerStatus	Network-Interfaces		
aib01	hmcib01p	on	on	en3-Online	ib0-Online	ib1-Online
aib02	hmcib01p	on	on	en3-Online	ib0-Online	ib1-Online
aib03	hmcib01p	on	on	en3-Online	ib0-Online	ib1-Online
aib04	hmcib01p	on	on	en3-Online	ib0-Online	ib1-Online

Compare the output shown in Example 4-22 on page 117 (no IB information) to the output shown in Example 4-29 (IB configured). The IP over InfiniBand devices are displayed like any other IP device.

Basic configuration data for IP network device can be obtained using the **ifconfig** command shown in Example 4-30.

Example 4-30 Checking network adapter information

```
aib04:/>ifconfig ib0
ib0:
flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT
>
    inet 192.168.8.164 netmask 0xffffffff broadcast 192.168.8.255
    tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1

aib04:/>ifconfig ib1
ib1:
flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT
>
    inet 192.168.9.164 netmask 0xffffffff broadcast 192.168.9.255
    tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

For comparison, Example 4-31 shows the same command output for an standard Ethernet device.

Example 4-31 Ethernet interface parameters

```
aib04: />ifconfig en3
en3:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GRO
UPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
    inet 192.168.100.164 netmask 0xffffffff broadcast
192.168.100.255
    tcp_sendspace 131072 tcp_recvspace 65536
```

Another typical IP command is **arp**. This shows information about the address resolution protocol (ARP). Since the network adapter used is an InfiniBand adapter, the hardware address has a different format. Therefore, we have to explicitly specify the adapter type **ib**, as shown in Example 4-32.

In this example, we can easily see that our InfiniBand adapters are connected to two InfiniBand switches and which adapter is connected to which switch. All arp entries with the same destination GID (DGID) belong to the same switch. Also, sender queue pair numbers (SQP), receiver queue pair numbers (RQP), sender local id (SLID), and destination local ID (DLID) can be seen.

Example 4-32 InfiniBand specific arp table entries.

```
aib04:/tmp> arp -t ib -a
aib01sw2 (192.168.9.161) at slid:0x0013 sqp:0x0021 dlid:0x0016 rqp:0x004b
DGID:fe:80:00:00:00:00:00:00:02:55:00:50:00:27:7d
aib01sw1 (192.168.8.161) at slid:0x000a sqp:0x0020 dlid:0x0007 rqp:0x004a
DGID:fe:80:00:00:00:00:00:00:02:00:02:55:00:50:00:27:3d
aib02sw2 (192.168.9.162) at slid:0x0013 sqp:0x0021 dlid:0x0014 rqp:0x001f
DGID:fe:80:00:00:00:00:00:00:02:55:00:50:00:1c:7d
aib02sw1 (192.168.8.162) at slid:0x000a sqp:0x0020 dlid:0x0008 rqp:0x001e
DGID:fe:80:00:00:00:00:00:00:02:00:02:55:00:50:00:1c:3d
aib03sw2 (192.168.9.163) at slid:0x0013 sqp:0x0021 dlid:0x0015 rqp:0x0043
DGID:fe:80:00:00:00:00:00:00:02:55:00:50:00:08:7d
aib03sw1 (192.168.8.163) at slid:0x000a sqp:0x0020 dlid:0x0009 rqp:0x0042
DGID:fe:80:00:00:00:00:00:00:02:00:02:55:00:50:00:08:3d
```

Total number of entries: 6

For comparison, Example 4-33 shows the output of the **arp** command for an Ethernet network interface.

Example 4-33 ARP information for an Ethernet adapter

```
aib04:/tmp>arp -a
  hmcib01 (192.168.100.28) at 0:d:60:b:31:a2 [ethernet] stored in bucket 125
  msib01 (192.168.100.29) at 0:2:55:cf:37:1b [ethernet] stored in bucket 126
  bglfen1 (192.168.100.41) at 0:2:55:d3:e0:c6 [ethernet] stored in bucket 138
  ? (192.168.100.48) at 0:d:60:b:3f:44 [ethernet] stored in bucket 145
```

Similar to Ethernet IP devices, information about the hardware addresses can also be retrieved (on AIX only) using the **netstat -i** command, as shown in Example 4-34.

Example 4-34 netstat -i command on IB adapters

```
aib04:/>netstat -I ib0
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
ib0 2044 link#3 0.0.0.1c.fe.80.0.0.0.0.0.2.0.2.55.0.50.0.22.3d 534236 0
1076194 0 0
ib0 2044 192.168.8 aib04sw1 534236 0 1076194 0 0
```

For comparison, Example 4-35 shows the **netstat -i** command output for an Ethernet network interface.

Example 4-35 netstat -i for an Ethernet interface

```
aib04:/>netstat -I en3
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
en3 1500 link#2 0.14.5e.96.1a.23 93193 0 46261 3 0
en3 1500 192.168.100 aib04 93193 0 46261 3 0
```

The most detailed information about the configuration and the current status of the InfiniBand subsystem can be obtained with the **ibstat** command. This command returns detailed data on the InfiniBand node, its ports, and its network interfaces, as shown in Example 4-36 on page 125.

Example 4-36 Detailed output of the ibstat command

```
aib04:/tmp>ibstat -v iba0
```

```
=====
  INFINIBAND DEVICE INFORMATION (iba0)
=====
```

```
Infiniband Debug Disabled
```

```
-----
  IB NODE INFORMATION (iba0)
-----
```

```
Number of Ports:                2
Globally Unique ID (GUID):      00.02.55.00.50.00.22.00
Maximum Number of Queue Pairs:  16367
Maximum Outstanding Work Requests: 32768
Maximum Scatter Gather per WQE:  252
Maximum Number of Completion Queues: 16380
Maximum Multicast Groups:       256
Maximum Memory Regions:         61382
Maximum Memory Windows:         61382
```

```
-----
  IB PORT 1 INFORMATION (iba0)
-----
```

```
Global ID Prefix:                fe.80.00.00.00.00.00.02
Local ID (LID):                  000a
Port State:                      Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
  P_Key[0]:                      ffff
Current Number of GUID's:        1
Globally Unique ID List:
  GUID[0]:                       00.02.55.00.50.00.22.3d
```

```
-----
  IB PORT 2 INFORMATION (iba0)
-----
```

```
Global ID Prefix:                fe.80.00.00.00.00.00.00
Local ID (LID):                  0010
Port State:                      Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
```

```

P_Key[0]:                ffff
Current Number of GUID's: 1
Globally Unique ID List:
  GUID[0]:                00.02.55.00.50.00.22.7d

```

IB INTERFACE ARP TABLE

```

aib01sw2 (192.168.9.161) at slid:0x0010 sqp:0x0029 dlid:0x000b rqp:0x0022
DGID:fe:80:00:00:00:00:00:00:02:55:00:50:00:27:7d
aib01sw1 (192.168.8.161) at slid:0x000a sqp:0x0028 dlid:0x0007 rqp:0x0023
DGID:fe:80:00:00:00:00:00:02:00:02:55:00:50:00:27:3d

```

Total number of entries: 2

IB INTERFACE (ib0) INFORMATION

```

ib0: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>
      inet 192.168.8.164 netmask 0xffffffff broadcast 192.168.8.255
      tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1

```

device		N/A	True
ib_adapter	iba0	Infiniband Host Channel Adapter	True
ib_port	1	Infiniband Host Channel Adapter Port	True
mtu	2044	Maximum IP Packet Size for This Device	True
netaddr	192.168.8.164	Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
p_key	-1	Partition Key	True
rfc1323	1	Enable/Disable TCP RFC 1323 Window Scaling	True
srq_size	4000	Transmit or Receive Hardware Queue Size	True
state	up	Current Interface Status	True
tcp_recvspace	131072	Set Socket Buffer Space for Receiving	True
tcp_sendspace	131072	Set Socket Buffer Space for Sending	True

IB INTERFACE (ib1) INFORMATION

```

ib1: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>
      inet 192.168.9.164 netmask 0xffffffff broadcast 192.168.9.255
      tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1

```

device		N/A	True
ib_adapter	iba0	Infiniband Host Channel Adapter	True
ib_port	2	Infiniband Host Channel Adapter Port	True

mtu	2044	Maximum IP Packet Size for This Device	True
netaddr	192.168.9.164	Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
p_key	-1	Partition Key	True
rfc1323	1	Enable/Disable TCP RFC 1323 Window Scaling	True
srq_size	4000	Transmit or Receive Hardware Queue Size	True
state	up	Current Interface Status	True
tcp_recvspace	131072	Set Socket Buffer Space for Receiving	True
tcp_sendspace	131072	Set Socket Buffer Space for Sending	True

Most of the IP specific information of the **ibstat** output is well known from Ethernet networks and is identical to the output provided by **arp**, **ifconfig**, and **lsattr** commands. We can limit the output of **ibstat** to InfiniBand specific attributes with the **-np** option, as shown in Example 4-37.

*Example 4-37 InfiniBand specific output of **ibstat -np***

```
aib04:/tmp>ibstat -pn

=====
INFINIBAND DEVICE INFORMATION (iba0)
=====

-----
IB NODE INFORMATION (iba0)
-----
Number of Ports:                2
Globally Unique ID (GUID):     00.02.55.00.50.00.22.00
Maximum Number of Queue Pairs: 16367
Maximum Outstanding Work Requests: 32768
Maximum Scatter Gather per WQE: 252
Maximum Number of Completion Queues: 16380
Maximum Multicast Groups:      256
Maximum Memory Regions:        61382
Maximum Memory Windows:        61382

-----
IB PORT 1 INFORMATION (iba0)
-----
Global ID Prefix:               fe.80.00.00.00.00.00.02
Local ID (LID):                 000a
Port State:                     Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
```

```

P_Key[0]:                ffff
Current Number of GUID's: 1
Globally Unique ID List:
  GUID[0]:                00.02.55.00.50.00.22.3d

```

IB PORT 2 INFORMATION (iba0)

```

Global ID Prefix:        fe.80.00.00.00.00.00.00
Local ID (LID):          0013
Port State:              Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
  P_Key[0]:              ffff
Current Number of GUID's: 1
Globally Unique ID List:
  GUID[0]:                00.02.55.00.50.00.22.7d
aib04:/tmp>

```

Finally, the InfiniBand adapter and device information can be verified. As discussed in 4.2.2, “IP over InfiniBand (IPoIB) implementation” on page 89, there should be four devices per node in our setup: the InfiniBand Host Channel Adapter `iba0`, the IB communication manager device `icm0` and two IPoIB network interfaces `ib0` and `ib1`. All of them should be in the Available state, as shown in Example 4-38.

Example 4-38 IB devices configured

```

aib04:/tmp>lsdev | grep ib
ib0      Available      IP over Infiniband Network Interface
ib1      Available      IP over Infiniband Network Interface
iba0     Available      Infiniband Host Channel Adapter
icm      Available      Infiniband Communication Manager

```

The device attributes of the IPoIB network interfaces `ib0` and `ib1` can be checked with the `lsattr` command. Since the current version of AIX does not support the same amount of IP features on InfiniBand as it does on Ethernet, this device has less attributes than the `en3` device, as shown in Example 4-39 on page 129.

Example 4-39 Attributes of ib and en devices

```
aib04: />lsattr -El ib0
```

device		N/A	True
ib_adapter	iba0	Infiniband Host Channel Adapter	True
ib_port	1	Infiniband Host Channel Adapter Port	True
mtu	2044	Maximum IP Packet Size for This Device	True
netaddr	192.168.8.164	Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
p_key	-1	Partition Key	True
rfc1323	1	Enable/Disable TCP RFC 1323 Window Scaling	True
srq_size	4000	Transmit or Receive Hardware Queue Size	True
state	up	Current Interface Status	True
tcp_recvspace	131072	Set Socket Buffer Space for Receiving	True
tcp_sendspace	131072	Set Socket Buffer Space for Sending	True

```
aib04: />lsattr -El en3
```

alias4		IPv4 Alias including Subnet Mask	True
alias6		IPv6 Alias including Prefix Length	True
arp	on	Address Resolution Protocol (ARP)	True
authority		Authorized Users	True
broadcast		Broadcast Address	True
mtu	1500	Maximum IP Packet Size for This Device	True
netaddr	192.168.100.164	Internet Address	True
netaddr6		IPv6 Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
prefixlen		Prefix Length for IPv6 Internet Address	True
remmtu	576	Maximum IP Packet Size for REMOTE Networks	True
rfc1323		Enable/Disable TCP RFC 1323 Window Scaling	True
security	none	Security Level	True
state	up	Current Interface Status	True
tcp_mssdflt		Set TCP Maximum Segment Size	True
tcp_nodelay		Enable/Disable TCP_NODELAY Option	True
tcp_recvspace		Set Socket Buffer Space for Receiving	True
tcp_sendspace		Set Socket Buffer Space for Sending	True

Example 4-40 shows the icm device attributes.

Example 4-40 Device attributes of the AIX icm

```
aib04: /> lsattr -El icm
MaxNumConn      65000 Maximum Number of Connections          True
MaxNumEqe       1000  Maximum Queued Exception Notifications Per User True
MaxNumHca       256   Maximum Number of HCA's                 True
MaxNumSid       1000  Maximum Number of Service ID's         True
MaxNumUser      65000 Maximum Number of Users                 True
MaxNumWkrq     65000 Maximum Number of Work Requests         True
MaxRecords      64    Maximum Number of Records Per Request   True
MaxResponseTime 4300  Maximum Response Time (msec)           True
MaxRetries      7     Maximum Request Retries                 True
MinResponseTime 100   Minimum Response Time (msec)           True
MinRetries      1     Minimum Request Retries                 True
NumMadBufs     64    Number of MAD buffers per HCA          True
aib04: />
```

To verify that InfiniBand adapters transfer data, we can use the **topas** command. On three nodes, we started a process that copies to the fourth node (aib02) using this command:

```
dd if=/dev/zero bs=256k | rsh aib02 "cat - >/dev/null"
```

This generates large blocks of binary zeros, transfers them to another node through **rsh**, and dumps them to **/dev/null**. Example 4-41 shows a snapshot of the **topas** output while three of these commands were running.

Example 4-41 Output of the topas command showing data transfer over InfiniBand

```
Topas Monitor for host:  aib02          EVENTS/QUEUES  FILE/TTY
Thu Oct  5 17:50:44 2006  Interval:  2    Cswitch 170.1K3  Readch   0.2G
                               Syscall 235.3K2  Writech  0.2G
Kernel  75.3  |#####| Reads   117.4K0  Rawin    0
User     4.8  |##| Writes  117.4K5  Ttyout   324
Wait     0.0  | | Forks    0  Igets    0
Idle    19.9  |#####| Execs    0  Namei    0
                               Runqueue  3.5  Dirblk   0
Network KBPS  I-Pack  O-Pack  KB-In  KB-Out  Waitqueue  0.0
ib0     260.2K 131.0K  62.9K  260.2K  0.0
en3      0.4   1.0    0.5    0.0    0.3
ib1      0.0   0.0    0.0    0.0    0.0
lo0      0.0   0.0    0.0    0.0    0.0
PAGING
Faults   0  Real,MB  3712
Steals   0  % Comp   22.3
PgspIn   0  % Noncomp 9.0
Disk     Busy%  KBPS    TPS  KB-Read  KB-Writ  PgspOut  0  % Client  9.0
hdisk1   0.0   0.0    0.0   0.0    0.0  PageIn   0
```

```

cd0      0.0      0.0      0.0      0.0      0.0      PageOut      0      PAGING SPACE
dac0     0.0      0.0      0.0      0.0      0.0      Sios          0      Size,MB      512
hdisk0   0.0      0.0      0.0      0.0      0.0          % Used      1.2
dac1     0.0      0.0      0.0      0.0      0.0      NFS (calls/sec) % Free      98.7
hdisk5   0.0      0.0      0.0      0.0      0.0      ServerV2     0
hdisk2   0.0      0.0      0.0      0.0      0.0      ClientV2     0      Press:
hdisk3   0.0      0.0      0.0      0.0      0.0      ServerV3     0      "h" for help
hdisk4   0.0      0.0      0.0      0.0      0.0      ClientV3     0      "q" to quit

```

```

Name          PID  CPU%  PgSp  Owner
CqKp          221390  38.1   0.4  root
cat           430186  10.1   0.1  root
cat           372832   9.7   0.1  root
cat           434250   9.7   0.1  root
syncd        123104   0.1   0.5  root

```

4.6 GPFS installation and configuration

In this section, we describe the installation and configuration of the General Parallel File System (GPFS). We use InfiniBand as a transport for inter-node communication.

4.6.1 Communication considerations for GPFS

GPFS is a parallel file system. Each node that has a GPFS file system mounted must be able to communicate with all storage devices that are part of this file system. This can be achieved with all nodes connected to shared disks in a Storage Area Network (SAN). This results in high disk access bandwidth on all nodes. However, for large clusters, this requires a large SAN infrastructure, which may be expensive and difficult to implement and maintain.

If SAN cannot be used to connect all nodes in the cluster to the storage, GPFS can access disks over the network, using the Network Shared Disks (NSD). NSDs are directly attached (through SAN) to at least one cluster node, the primary NSD server. To increase availability, it is possible to also define a secondary NSD server for each disk in the GPFS file system. The nodes that do not have direct access to the disk (SAN) will access the disks through the NSD servers over an IP network. This network often is the performance limiting factor.

In our test setup, we will use IP over InfiniBand (IPoIB) to increase the bandwidth used for accessing NSDs compared to Ethernet networks.

GPFS requires that each node is able to execute remote commands on all other nodes as root user, and without user interaction of any kind (password, passphrase, or key acceptance). By default, **rsh** is used for remote command execution and **rcp** for copying configuration files. Another option is to use secure shell/copy (**ssh/scp**). For details, see the manual *GPFS V3.1: Concepts, Planning, and Installation Guide*, GA76-0413.

Note: In our environment, as we have assimilated our setup to a High Performance Computing (HPC) cluster, we have decided to use the standard remove command/copy programs (**rsh/rcp**) available on AIX. If you plan to use secure shell/copy (**ssh/scp**), you need to install additional software from the AIX Toolbox for Linux CDs.

If the cluster nodes are connected by more than one network interface, by default remote commands are executed through the network that is used for the NSD data transfer. In our test cluster, each node has three communication interfaces configured with IP protocol: one Ethernet interface and two IPoIB networks.

Since we are using one of our IPoIB networks for GPFS data and metadata traffic, the IP addresses for this subnet must be listed in the `~/rhosts` file. For using separate networks, see the *GPFS V3.1: Concepts, Planning, and Installation Guide*, GA76-0413.

4.6.2 GPFS installation

This section describes the planning and installation of GPFS in our environment.

Attention: In this section, we assume that remote command / copy between all GPFS nodes has been set up properly (no user interaction).

Hardware setup

As there are manuals and documents available, we do not discuss in detail hardware requirements for GPFS. In short, we need cluster nodes connected together through a fast IP network, and disks (LUNs) that are connected to one or more of the cluster nodes. For a detailed description of the hardware requirements, see the *GPFS V3.1: Concepts, Planning, and Installation Guide*, GA76-0413.

In our setup, we have two FC attached LUNs hosted in a DS 4500 Total Storage subsystem. The storage is attached to two cluster nodes acting as primary and secondary NSD servers.

Software setup

The GPFS software is packaged in three filesets. They are installed on our systems as shown in Example 4-42.

Example 4-42 GPFS installed filesets

```
aib04:/tmp>lspp -l gpfs.*
  Fileset                Level  State      Description
-----
Path: /usr/lib/objrepos
  gpfs.base              3.1.0.5  APPLIED    GPFS File Manager
  gpfs.msg.en_US         3.1.0.4  APPLIED    GPFS Server Messages - U.S. English

Path: /etc/objrepos
  gpfs.base              3.1.0.5  APPLIED    GPFS File Manager

Path: /usr/share/lib/objrepos
  gpfs.docs.data         3.1.0.1  APPLIED    GPFS Server Manpages and Documentation
```

In preparation of the GPFS installation, it is convenient to create a file that contains all node names that will be part of the cluster, one per line.

Attention: Since we use IPoIB transport for our NSDs, this file contains the IPoIB names of our GPFS cluster nodes.

In our setup, we have created a file named `/tmp/gpfsnodes`, as shown in Example 4-43.

Example 4-43 Node descriptor file for our cluster

```
aib01:/tmp>cat /tmp/gpfsnodes
aib01sw1:quorum
aib02sw1:quorum
aib03sw1
aib04sw1
```

Two of the nodes are configured as quorum nodes. For a detailed discussion of the quorum concepts in GPFS, see the *GPFS V3.1 Concepts, Planning, and Installation Guide*, GA76-0413.

GPFS configuration

The first step in creating the GPFS cluster is using the `mmcrcluster` command shown in Example 4-44.

Example 4-44 Creating the GPFS cluster

```
aib01:/tmp> mmcrcluster -C IBcluster -N /tmp/gpfsnodes -paib01sw1 -s
aib02sw2
Thu Oct 5 13:45:35 EDT 2006: mmcrcluster: Processing node aib01sw1
Thu Oct 5 13:45:36 EDT 2006: mmcrcluster: Processing node aib02sw1
Thu Oct 5 13:45:37 EDT 2006: mmcrcluster: Processing node aib03sw1
Thu Oct 5 13:45:38 EDT 2006: mmcrcluster: Processing node aib04sw1
mmcrcluster: Command successfully completed
mmcrcluster: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

This is executed on one of the cluster nodes. It creates the GPFS cluster named `IBcluster` with node `aib01sw1` as the primary server and node `aib02sw2` as the secondary server on all cluster nodes. To confirm the cluster definition, we use the `mm1scluster` command, as shown in Example 4-45.

Example 4-45 Output of the `mm1scluster` command

```
aib01:/tmp>mm1scluster
```

```
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      IBcluster.aib01sw1
GPFS cluster id:       13882355340112381119
GPFS UID domain:      IBcluster.aib01sw1
Remote shell command:  /usr/bin/rsh
Remote file copy command: /usr/bin/rcp
```

```
GPFS cluster configuration servers:
```

```
-----
```

```
Primary server:  aib01sw1
Secondary server: aib02sw1
```

Node	Daemon node name	IP address	Admin node name
Designation			
1	aib01sw1	192.168.8.161	aib01sw1 quorum
2	aib02sw1	192.168.8.162	aib02sw1
3	aib03sw1	192.168.8.163	aib03sw1
4	aib04sw1	192.168.8.164	aib04sw1

The GPFS daemons are started with the **mmstartup -a** command, as shown in Example 4-46. In our case, GPFS will also be started automatically after a system reboot.

Example 4-46 Starting GPFS daemon

```
aib01:/tmp>mmstartup -a
Thu Oct 5 13:46:19 EDT 2006: mmstartup: Starting GPFS ...
The GPFS subsystem is already active.
```

Next, we define the NSDs. As previously mentioned, we have two FC attached LUNs (hdisk5 to the AIX OS), as shown in Example 4-47.

Example 4-47 Disk devices attached to our storage servers

```
aib04:/tmp>lscfg | grep 1742
* dac0          U787F.001.DPM18DC-P1-C3-T1-W200300A0B812106F 1742-900 (900) Disk Array
Controller
* dac1          U787F.001.DPM18DC-P1-C3-T1-W200200A0B812106F 1742-900 (900) Disk Array
Controller
+ hdisk4        U787F.001.DPM18DC-P1-C3-T1-W200300A0B812106F-L0 1742-900 (900) Disk Array
Device
+ hdisk5        U787F.001.DPM18DC-P1-C3-T1-W200300A0B812106F-L1000000000000 1742-900 (900)
Disk Array Device
```

Even though it is possible supply disk parameters to the **mmcrnsd** command on the command line, this is not desirable, especially in configurations with large numbers of disks. We create a disk descriptor file for our NSDs.

The disks to become NSDs are listed in this file, one per line; for each disk, we specify information about the disk name, its primary and secondary NSD servers, and information about the (GPFS) disk usage. For a detailed description of the format of the disk descriptor file, see the *GPFS V3.1 Concepts, Planning, and Installation Guide*, GA76-0413. We have created a file named `/tmp/nsddef`, as shown in Example 4-48.

Example 4-48 Disk descriptor file for our NSDs

```
aib01:/tmp>cat /tmp/nsddef
hdisk4:aib01:aib02:dataAndMetadata:
hdisk5:aib01:aib02:dataAndMetadata:
```

Next, we create the NSD using the `mmcrnsd` command, as shown in Example 4-49.

Example 4-49 Creating NSDs

```
aib01:/tmp>mmcrnsd -F /tmp/nsddef
mmcrnsd: Processing disk hdisk4
mmcrnsd: Processing disk hdisk5
mmcrnsd: Propagating the cluster configuration data to all
        affected nodes. This is an asynchronous process.
```

To verify the NSD definition, we use the `mm1snsd` command, as shown in Example 4-50.

Example 4-50 Listing NSDs

```
aib01:/tmp>mm1snsd
```

File system	Disk name	Primary node	Backup node
(free disk)	gpfs1nsd	aib01sw1	
(free disk)	gpfs2nsd	aib01sw1	

Next, we create the GPFS file system using the `mmcrfs` command. The `mmcrnsd` command has updated the `/tmp/nsddef` file with information about the newly created NSDs (Example 4-51) so we can reuse it for creating the GPFS file system.

Example 4-51 Disk descriptor file prepared for adding disks to a file system

```
aib01:/tmp>cat /tmp/nsddef
# hdisk4:aib01sw1::dataAndMetadata
gpfs1nsd::dataAndMetadata:4001::
# hdisk5:aib01sw1::dataAndMetadata
gpfs2nsd::dataAndMetadata:4001::
```

We create the file system using the `mmcrfs` command shown in Example 4-52 on page 137.

Example 4-52 Creating the GPFS file system

```
aib01:/tmp>mmcrfs /gpfs gpfs1 -F /tmp/nsddef -B 512K -R 2 -M 2
```

```
The following disks of gpfs1 will be formatted on node aib01:
  gpfs1nsd: size 71687000 KB
  gpfs2nsd: size 71687000 KB
Formatting file system ...
Disks up to size 153 GB can be added to storage pool 'system'.
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Completed creation of file system /dev/gpfs1.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

For a complete description of this command see the manual *GPFS V3.1 Administration and Programming Reference*, SA23-2221. To verify the file system definition, we have used the `mm1sfs` command, as shown in Example 4-53.

Example 4-53 Output of the mm1sfs command showing the GPFS config details

```
aib01 # mm1sfs gpfs1
flag value      description
-----
-s roundRobin   Stripe method
-f 16384        Minimum fragment size in bytes
-i 512          Inode size in bytes
-I 16384        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 cluster     Block allocation type
-D posix       File locking semantics in effect
-k posix       ACL semantics in effect
-a 1048576     Estimated average file size
-n 32          Estimated number of nodes that will mount file system
-B 524288     Block size
-Q none        Quotas enforced
  none        Default quotas enabled
-F 53248      Maximum number of inodes
-V 9.03       File system version. Highest supported version: 9.03
-u yes        Support for large LUNs?
```

```

-z no          Is DMAPI enabled?
-E yes        Exact mtime mount option
-S no        Suppress atime mount option
-K whenpossible Strict replica allocation option
-P system     Disk storage pools in file system
-d gpfs1nsd;gpfs2nsd Disks in file system
-A yes        Automatic mount option
-o none       Additional mount options
-T /gpfs      Default mount point

```

Finally, we can mount the file system and verify its size, as shown in Example 4-54.

Example 4-54 Mounting the newly create dGPFD file system on all nodes

```
aib01:/tmp>mmmount /gpfs -a
```

```
aib01:/tmp>df -g /gpfs
```

Filesystem	GB blocks	Free	%Used	Iused	%Iused	Mounted on
/dev/gpfs1	546.93	395.58	28%	4040	3%	/gpfs

4.6.3 Monitoring GPFS over InfiniBand

Now we will monitor our GPFS and verify that the InfiniBand adapters are being used for data transport between the nodes.

Important: This setup was designed to demonstrate GPFS configuration for InfiniBand using IPoIB. It was never meant to obtain any performance data or do optimizations. The limiting factor in our setup is number of NSDs and NSD servers that affect the GPFS data striping performance (we only have two disks and two storage nodes).

To check the data traffic (flow), we have used the **topas** command. For this, we created a load by writing to the /gpfs file system on each node. In Example 4-55 on page 139, we can see the outgoing and incoming data through the ib0 network interface and being read/written to the hdisks. Due to caching parameters, network and disk data rates do not match.

Example 4-55 Topas showing the data flow over InfiniBand

```

Topas Monitor for host:   aib02
Thu Oct  5 17:31:42 2006 Interval:  2

Kernel  9.3  |###|
User    0.8  |#  |
Wait    0.0  |   |
Idle    90.0 |#####|

EVENTS/QUEUES  FILE/TTY
Cswitch  2436  Readch  0.0G
Syscall  22703 Writech  0.0G
Reads    174   Rawin   0
Writes   174   Ttyout  351
Forks    0     Igets   0
Execs    0     Namei   0
Runqueue 0.0   Dirblk  0
Waitqueue 0.0

Network  KBPS  I-Pack  O-Pack  KB-In  KB-Out
ib0     19.0K  9853.0  3434.5  19.0K  0.0
en3     0.4    1.0    0.5    0.0    0.4
ib1     0.0    0.0    0.0    0.0    0.0
lo0     0.0    0.0    0.0    0.0    0.0

PAGING  MEMORY
Faults  33   Real,MB  3712
Steals  0   % Comp   22.2
PgspIn  0   % Noncomp 9.0
PgspOut 0   % Client  9.0
PageIn  0
PageOut 0  PAGING SPACE
Sios    0   Size,MB  512
        % Used  1.2
NFS (calls/sec) % Free  98.7
ServerV2 0
ClientV2 0  Press:
ServerV3 0  "h" for help
ClientV3 0  "q" to quit

Disk  Busy%  KBPS  TPS  KB-Read  KB-Writ
hdisk5 100.0  32.5K  63.5  2.5  32.5K
hdisk4 100.0  14.5K  29.0  0.0  14.5K
dac0   0.0   47.0K  92.5  2.5  47.0K
hdisk0 0.0   0.0   0.0   0.0   0.0
dac1   0.0   0.0   0.0   0.0   0.0
hdisk1 0.0   0.0   0.0   0.0   0.0
hdisk2 0.0   0.0   0.0   0.0   0.0
hdisk3 0.0   0.0   0.0   0.0   0.0
cd0    0.0   0.0   0.0   0.0   0.0

Name      PID  CPU%  PgSp  Owner
mmfsd64   311478  4.1  4.8  root
CqKp      221390  3.4  0.4  root
dd        413734  2.0  0.1  root
dd        352434  0.5  0.1  root
topas     372802  0.1  1.1  root
getty     266372  0.0  0.4  root
gil       69666  0.0  0.9  root

```




IBM System p cluster with InfiniBand and SUSE SLES 9

This chapter provides information about how to configure the InfiniBand adapters on IBM System p5 running SLES 9. We also provide information about how to set up an environment of an IB cluster with SLES 9. The following topics are discussed:

- ▶ InfiniBand considerations for SLES 9
- ▶ Introduction of IB cluster running SLES 9 SW components
- ▶ Installation and configuration of an IB cluster with SLES 9

5.1 InfiniBand considerations for SLES 9

This section describes the basic elements and considerations for an InfiniBand implementation of IBM System p with SLES 9. It includes following topics:

- ▶ Supported hardware
- ▶ Software components and versions
- ▶ InfiniBand implementation on SLES 9

5.1.1 Supported hardware

For a complete list of hardware components and versions supported, refer to 3.5, “Supported System p servers” on page 40. The basic configuration capable of running IB on System p with SLES 9 is a partition in one server supporting the IBM GX adapters. The software requirements are described later in 5.1.2, “Software components and versions” on page 143. Depending on the application needs, the partition should obtain the appropriate CPU capability, RAM size, and disk space.

Note: An IB HCA must be assigned to the SLES 9 partition.

On the HMC, open the Properties dialog (right-click the partition's profile) and make sure a GUID index is assigned to the HCA with a valid value (in range 1-62), as shown in Figure 5-1.

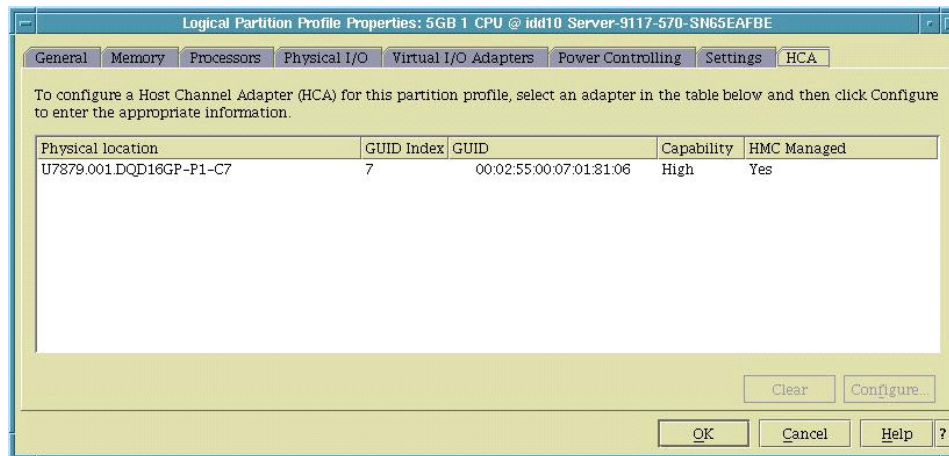


Figure 5-1 GUID Index

5.1.2 Software components and versions

Here we discuss the software components and their versions.

Open Fabrics Enterprise Distribution (OFED)

OFED is a Linux based software package that has been designed to exploit the InfiniBand infrastructure, as specified by the IBTA.

OFED has been developed by the OpenFabrics Alliance organization with the active participation of many HCA vendors, industry members, and national research laboratories. For a complete description of the OFED software stack, check the following Web page:

http://www.openfabrics.org/docs/Sep06_OpenFabrics_Stack.ppt

For more details, you can also check the OpenIB Wiki pages at:

<http://openfabrics.org/tiki>

The OFED stack consists of the following components:

- ▶ Core layer
This exposes APIs/IB verbs required to access InfiniBand specific resources and functionality, for example, memory regions, queue pairs, MAD service, and so on. It implements the infrastructure for management of device drivers as well as HCA resources. In addition, the core layer is a vehicle for user and kernel communication.
- ▶ Driver layer
This consists of device drivers provided by the HCA provider. A device driver implements the kernel services as defined by IB verbs to allow consumers to access and manipulate the HCA's resources. In the case of eHCA, it is `hcad_mod` and `ib_ehca` respectively.
- ▶ Upper level protocol layer
This composes kernel modules that rely on the core layer and provide another abstraction, for example, IP over InfiniBand or Socket Direct Protocol. This approach helps consumers to utilize InfiniBand based on the abstraction they already know.
- ▶ User space API layer
This includes a set of libraries allowing user-level clients to access kernel services. Typically, a device driver provider also delivers an implementation of the IB user space API as a library. In the case of eHCA, it is `libehca`.

► Application layer

This includes all tools and applications respectively that rely on IB user space API. Examples are the diagnostic tools such as `ibstat`, `ibping`, `perfquery`, and so on, located under the directory `src/userspace/management/` or various implementations of the Message Passing Interface (MPI).

Software for supporting InfiniBand HCA

The following software components are required in order to exploit InfiniBand technology on a SLES 9 Operating System Image (OSI):

1. SUSE SLES 9, Service Pack 3
2. `libsfs` for user space support only

This library provides a generic interface for querying system device information exposed through `sysfs`. Its default location is in `/usr/local/lib` and `/usr/local/lib64` respectively.

3. IBM HCA device driver (eHCA)

This component allows OFED to access and exploit an underlying IBM HCA.

Note: The device driver is a part of OFED!

Cluster software components and versions

In addition to the device driver and the library, to exploit the IB infrastructure, you should also install IBM cluster software to build a SLES 9 cluster. The cluster software packages that can be used for exploiting high speed InfiniBand technology are CSM, RSCT, GPFS, PE, and TWS LL. The current packages for these products are listed in Table 5-1.

Table 5-1 Cluster software packages and available versions for SLES 9

Software package / Available version	RPMs and versions
Cluster Systems Management (CSM) 1.6.0.1	<code>csm.client-1.6.0.1-34</code> <code>csm.core-1.6.0.1-34</code> <code>csm.dsh-1.6.0.1-34</code> <code>csm.diagnostics-1.6.0.1-34</code> <code>csm.gui.dcem-1.6.0.1-34</code> <code>csm.server-1.6.0.1-34</code> <code>csm.deploy-1.6.0.1-34</code>

Software package / Available version	RPMs and versions
Reliable Scalable Clustering Technology (RSCT) 2.4.6.1	rrsct.64bit-2.4.6.1-0 rsct.basic-2.4.6.1-06290 rsct.core-2.4.6.1-06290 rsct.core.cimrm-2.4.6.1-06290 rsct.core.utils-2.4.6.1-06290 rsct.sdk-2.4.6.1-06290
General Parallel File System (GPFS) 3.1.0.6	gpfs.base-3.1.0-6 gpfs.docs-3.1.0-6 gpfs.gpl-3.1.0-6 gpfs.msg.en_US-3.1.0-6
Parallel Environment (PE) 4.3.0.1	ppe_ppc_64bit_sles900-4.3.0.1-s001a ppe_ppc_base_32bit_sles900-4.3.0.1-s001a lapi_ppc_32bit_US_sles900-2.4.3.1-s001a lapi_ppc_32bit_base_IP_sles900-2.4.3.1-s001a lapi_ppc_64bit_IP_sles900-2.4.3.1-s001a lapi_ppc_64bit_US_sles900-2.4.3.1-s001a
IBM Tivoli Workload Scheduler (TWS) LoadLeveler 3.4.0.1	LoadL-full-SLES9-PPC64-3.4.0.1-0 LoadL-full-lib-SLES9-PPC-3.4.0.1-0 LoadL-full-license-SLES9-PPC64-3.4.0.1-0

5.1.3 InfiniBand implementation on SLES 9

Note: Throughout this chapter, we have changed the host name of our systems to reflect Linux installations. For example, host “ib02” becomes “lib02”.

Installing OFED and eHCA on SLES 9 SP3

1. Download the OpenIB support package (in our case, the file is named openib-0058-gen2-sles9sp3.ppc.tar.gz) from the following Web page:
<http://developer.novell.com/devres/storage/drivers/index.html#IBM>

At the time of this writing, we obtained a table containing the downloadable IB drivers for SLE9, as shown Figure 5-2.

SLES 9				
Driver	Version	Service Pack	Architecture	Adapters
adp94xx	1.0.7-11	SP3	i386	Adapter List
	1.0.7-11	SP3	x86-64	
	1.0.7-3	SP2	i386	
	1.0.7-3	SP2	x86-64	
	1.0.5-14	SP2	i386	
	1.0.5-14	SP2	x86-64	
	0.0.5-7	SP1	x86-64	
openib	0058	SP3	ppc	Adapter List
	0057	SP3	ppc	

Figure 5-2 eHCA download page for SLES 9

Click **0058** to download the openib-0058-gen2-sles9sp3.ppc.tar.gz file. If you do not have a Web browser available on the partition, use the following command in the shell interface to download the file:

```
lib02:~ # wget \
http://developer.novell.com/devres/storage/drivers/openib-0058-gen2-
sles9sp3.ppc.tar.gz
```

We have stored the file on the local node in the ~/work directory.

2. Unpack and verify the downloaded file.

In the directory that contains the downloaded file, we have created a new directory called sles9_openib (~/work/sles9_openib) and unpacked the openib-0058-gen2-sles9sp3.ppc.tar.gz, as shown in Example 5-1.

Example 5-1 Unpacking the openib package

```
lib02:~ # mkdir sels9_openib
lib02:~ # cd sles9_openib
lib02:~ # tar xzf ../openib-0058-gen2-sles9sp3.ppc.tar.gz
```

We have obtained the following RPM packages:

- openib_kernel_ehca2-0058-1.ppc64.rpm

This RPM package contains the IB compiled kernel modules shown in Table 5-2 on page 147.

Table 5-2 IB kernel modules

Module	Module usage
hcad_mod.ko	eHCA InfiniBand device driver module
ib_at.ko	OFED's kernel module, address translation
ib_cm.ko	OFED's kernel module, connection manager
ib_core.ko	OFED's kernel module, core layer used by all device drivers
b_ipoib.ko	OFED's kernel module, IP over InfiniBand
ib_mad.ko	OFED's kernel module, management datagram protocol
ib_ping.ko	OFED's kernel module
ib_sa.ko	OFED's kernel module, subnet agent
ib_uat.ko	OFED's kernel module, user space support for AT
ib_uverbs.ko	OFED's kernel module, user space support for InfiniBand
ibmebus.ko	eHCA ebus driver module

- openib_kernel_ehca2-0058-1.src.rpm
This RPM contains source code for the kernel modules shown in Table 5-2.
- openib_userspace_32-1.0RC2-1.ppc.rpm
This RPM provides 32-bit user space support as a 32-bit library (libibverbs.so) together with utilities, manuals, and header files required for developing and running user space applications.
- openib_userspace_32-1.0RC2-1.src.rpm
This RPM contains source code for the user space libraries and tools.
- openib_userspace_64-1.0RC2-1.ppc64.rpm
This RPM contains 64-bit libraries as well as utilities, manuals, and header files required for running and developing user space applications.
- openib_userspace_64-1.0RC2-1.src.rpm
This RPM contains source code for the 64-bit user space libraries and tools.
- openib_userspace_ehca-0058-1.ppc.rpm
This RPM contains the 32-bit eHCA user space library.

- openib_userspace_ehca-0058-1.src.rpm
This RPM contains source code for the 32-bit eHCA user space library.
 - openib_userspace_ehca_64-0058-1.ppc64.rpm
This RPM contains the 64-bit eHCA user space library.
 - openib_userspace_ehca_64-0058-1.src.rpm
This RPM contains source code for the 64-bit eHCA user space library.
3. Installing the packages containing the binary modules and libraries.
Installation of kernel modules and libraries requires root access.
- Install the kernel modules, that is, OFED and the eHCA device driver:
lib02:~ # rpm -ivh openib_kernel_ehca2-0058-1.ppc64.rpm
 - Install OFED user space libraries and tools for 32-bit support:
lib02:~ # rpm -ivh openib_userspace_32-1.0RC2-1.ppc.rpm
 - Install the eHCA user space library for 32-bit support:
lib02:~ # rpm -ivh openib_userspace_ehca-0058-1.ppc.rpm
- Alternately, for 64-bit support, you have to install the following RPMs:
- Install the OFED user space libraries and tools for 64-bit support:
lib02:~ # rpm -ivh openib_userspace_64-1.0RC2-1.ppc64.rpm
 - Install the eHCA user space library for 64-bit support:
lib02:~ # rpm -ivh openib_userspace_ehca_64-0058-1.ppc64.rpm

4. Verifying installation

Check if the following files are present, and in the correct location, as shown in Example 5-2 (the `rpm -ql` command is used to retrieve the list):

Example 5-2 Kernel files for InfiniBand

```
/etc/init.d/boot.ibm_ib
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/hcad_mod.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_at.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_cm.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_core.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_ipoib.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_mad.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_ping.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_sa.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_uat.ko
/lib/modules/2.6.5-7.244-pseries64/updates/infiniband/ib_uverbs.ko
```

The file `boot.ibm.ib` is used to load all kernel modules at boot time. Once the modules have been loaded, the `lsmod` command should show the following IB-related modules (among other modules):

```
ibmebus, ib_core, hcad_mod, ib_mad, ib_sa and ib_ipoib
```

To allow user space applications to run, the `ib_uverbs` module must be loaded. If it is not loaded, you can use the following command to load it manually:

```
lib02:~ # modprobe ib_uverbs
```

Note: In Linux, individual modules can be loaded using the `modprobe` command. However, the recommended way is to invoke the script that loads the modules at system boot time.

You can call the `/etc/init.d/boot.ibm_ib` script to start and stop OFED and load and unload eHCA device driver:

– Start OFED and the eHCA device driver:

```
lib02:~ # /etc/init.d/boot.ibm_ib start
```

– Stop OFED and the eHCA device driver:

```
lib02:~ # /etc/init.d/boot.ibm_ib stop
```

– Get a list and status of available HCAs:

```
lib02:~ # /etc/init.d/boot.ibm_ib status
```

Note: This command uses the user space utilities `ibv_devices` and `ibv_devinfo` to display the status information of available HCAs. This requires the kernel module `ib_uverbs` to be loaded as previously mentioned.

Note: A HCA port must be connected to a switch in order to be recognized as available.

5. IP over InfiniBand configuration (including TCP/IP).

Each available IB adapter port can be configured with an IP address using the following commands:

```
lib02:~ # ifconfig ib0 192.168.178.120
```

```
lib02:~ # ifconfig ib1 192.168.178.121
```

Note: The previous `ifconfig` command assumes that the network (subnet) mask is 24-bit (class C, 255.255.255.0). If your subnet mask is different, you must explicitly specify it.

As with Ethernet devices, the InfiniBand ports are numbered (by default) as `ib0`, `ib1`, and so on. In our case, we have configured both `ib0` and `ib1` in the same IP subnet.

6. Functional tests.

Note: For the following tests, two HCAs are required!

- IP ping test (assuming our IP address is 192.168.178.120):

```
lib02:~ # ping -f -l 20 192.168.178.121
```

which performs a flood ping with 20 packets.
- Copying a file using secure copy:

```
lib02:~ # scp -R /usr/src/linux 192.168.178.121:/tmp
```
- In addition to the standard TCP/IP tools, OFED also provides some useful user space applications. For details and examples, refer to 8.1, “Monitoring tools for AIX 5L and SLES 9” on page 250.

Building an eHCA device driver and user space library

Note: If you are installing IBM certified cluster hardware and software, it is not necessary or recommended to build the IB device driver or user space library from the source code.

However, if you intend to compile the drivers and libraries yourself (eHCA and OpenIB stack), you will need to install the Linux kernel source code (SLES9 SP3).

To compile and build the eHCA device driver from source code, you need first to install the `openib_kernel_ehca2-0058-1.src.rpm` RPM package (see “Installing OFED and eHCA on SLES 9 SP3” on page 145). To install the source code package, we have used the commands shown in Example 5-3 on page 151.

Example 5-3 Install source code of eHCA device driver

```
lib02:~ # rpm -ihv openib_kernel_ehca2-0058-1.src.rpm
lib02:~ # ls /usr/src/packages/SOURCES
ehca2_EHCA2_0058.tgz
linux-2.6.15-rc5_header.tar.gz
svn_trunk_6454_kernel.tgz
```

The source packages are stored in directory `/usr/src/packages/SOURCES`:

- ▶ `ehca2_EHCA2_0058.tgz`
Source code for eHCA kernel and user space drivers
- ▶ `linux-2.6.15-rc5_header.tar.gz`
Header files for kernel 2.6.15-rc5 required for backport compiling of eHCA
- ▶ `svn_trunk_6454_kernel.tgz`
OpenIB stack source code revision 6454.

To build and install an eHCA device driver, OpenIB stack, and user space libraries, execute the following steps:

1. Unpack the source code under `/usr/local/src/ehca2`, as shown in Example 5-4.

Example 5-4 Unpack source code

```
lib02:~ # mkdir -p /usr/local/src/ehca2
lib02:~ # cd /usr/local/src/ehca2
lib02:~ # tar zxf /usr/src/packages/SOURCES/ehca2_EHCA2_0058.tgz
lib02:~ # tar zxf \
/usr/src/packages/SOURCES/linux-2.6.15-rc5_header.tar.gz
lib02:~ # mkdir trunk_6454
lib02:~ # cd trunk_6454
lib02:~ # tar zxf /usr/src/packages/SOURCES/svn_trunk_6454_kernel.tgz
lib02:~ # cd ..
```

Executing the commands in Example 5-4 creates the following directories:

- `ehca2/`
- `linux-2.6.15-rc5/`
- `trunk_6454/`

2. Edit the Makefile located in the ehca2 directory, and update it with correct paths.

Using the editor of your choice, search for the lines containing the following variables, and correct the right hand side, as shown in Example 5-5

Example 5-5 Edit ehca2/Makefile

LINUX_PATH2615=/usr/src/linux-2.6.15-rc5	# wrong
LINUX_PATH2615=/usr/local/src/ehca2/linux-2.6.15-rc5	# correct
GEN2_PATH = /home/source/trunk_6454	# wrong
GEN2_PATH = /usr/local/src/ehca2/trunk_6454	# correct

3. Patch OpenIB stack for kernel 2.6.5-7.244, that is, SLES9 SP3, as shown in Example 5-6.

Note: This step needs to be done only once.

Example 5-6 Patch OpenIB stack

```
lib02:~ # cd /usr/local/src/ehca2
lib02:~ # make patch
...
cd /usr/local/src/ehca2/trunk_6454;patch -p1 <
/usr/local/src/ehca2/ehca2/ehca/linux265/svn_6454_patch
patching file src/linux-kernel/infiniband/core/addr.c
patching file src/linux-kernel/infiniband/core/sysfs.c
patching file src/linux-kernel/infiniband/core/ucm.c
patching file src/linux-kernel/infiniband/core/user_mad.c
patching file src/linux-kernel/infiniband/core/uverbs.h
patching file src/linux-kernel/infiniband/core/uverbs_mem.c
patching file src/linux-kernel/infiniband/ulp/ipoib/ipoib.h
cd /usr/local/src/ehca2/trunk_6454/src/linux-kernel; patch -p1 <
/usr/local/src/ehca2/ehca2/patches/24226_ipoib.patch
patching file infiniband/ulp/ipoib/ipoib_ib.c
patching file infiniband/ulp/ipoib/ipoib_main.c
```

4. Compile and install the eHCA, OpenIB modules, and user space libraries, as shown in Example 5-7 on page 153.

Example 5-7 Compile and install eHCA, OpenIB, and user space libraries

```
make
```

```
...
```

If you ever happen to want to link against installed libraries in a given directory, LIBDIR, you must either use libtool, and specify the full pathname of the library, or use the `-LLIBDIR` flag during linking and do at least one of the following:

- add LIBDIR to the ``LD_LIBRARY_PATH'` environment variable during execution
- add LIBDIR to the ``LD_RUN_PATH'` environment variable during linking
- use the ``-Wl,--rpath -Wl,LIBDIR'` linker flag
- have your system administrator add LIBDIR to ``/etc/ld.so.conf'`

See any operating system documentation about shared libraries for more information, such as the `ld(1)` and `ld.so(8)` manual pages.

```
-----  
make[2]: Leaving directory  
~/usr/local/src/ehca2/trunk_6454/src/userspace/libehca'  
make[1]: Leaving directory  
~/usr/local/src/ehca2/trunk_6454/src/userspace/libehca'
```

5. Verify the installation.

All kernel modules are installed in the directory `/lib/modules/<kernel revision>/extra`, as shown in Example 5-8.

Example 5-8 Kernel modules

```
lib02:~ # ls -l /lib/modules/`uname -r`/extra  
hcad_mod.ko  
ib_at.ko  
ib_cm.ko  
ib_core.ko  
ib_ipoib.ko  
ib_mad.ko  
ib_multicast.ko  
ib_ping.ko  
ib_sa.ko  
ib_uat.ko  
ib_ucm.ko  
ib_verbs.ko  
ibmebus.ko  
rdma_cm.ko  
rdma_ucm.ko
```

All OpenIB user space libraries are stored under the directory `/usr/local/lib` (32-bit) and `/usr/local/lib64` (64-bit). The eHCA user space library is located in `/usr/local/lib/infiniband` and `/usr/local/lib64/infiniband`. User space tools (`ibv_devinfo` etc.) can be found in `/usr/local/bin` (64-bit) and `/usr/local/bin32/bin` (32-bit).

If you need to apply a patch or any source code change under the `ehca2` directory, you need to issue the commands shown in Example 5-9 before compiling the new code.

Example 5-9 Preparing for compiling a new version of code

```
lib02:~ # pwd
current directory is /usr/local/src/ehca2
lib02:~ # cd ehca
lib02:~ # make prepare
# this will copy new code into OpenIB stack code tree appropriately
```

Note: In case you plan to install OFED and the eHCA device driver on another Linux kernel version, refer to Appendix D, “Installing OFED and eHCA on Linux Kernel 2.6.16, 2.6.17, and 2.6.18” on page 285.

5.2 Introduction of cluster software components for SLES 9

This section describes the IBM Cluster solution for System p with InfiniBand running SLES 9. We focus on describing cluster software packages that accompany IBM hardware elements to form the overall cluster solution. The software components include IBM Cluster System Management (CSM, which is based on Reliable Scalable Clustering Technology (RSCT)), General Parallel File System (GPFS), and HPC software (Low level Application Programming Interface (LAPI), Parallel Environment (PE) and LoadLeveler). This section covers the following topics:

- ▶ HPC cluster overview
- ▶ System management software for SLES9 clustering
- ▶ Software packages for High Performance Computing

5.2.1 HPC cluster overview

In recent years, High Performance Computing (HPC) has become popular in many industries. HPC has the capabilities to deliver the performance needed for the computing jobs used for mathematical simulation of complex natural phenomena. With the profit of clustering technology and the growing acceptance of open source software, supercomputers can now be created for a fraction of the cost of traditional high performance machines.

In the early times of cluster-based computing, the typical supercomputer was a solution based on vector processor(s), usually much more expensive due to specialized hardware and software. With SLES 9 and other open source clustering software components, and technological advances in commodity hardware, the situation now is quite different.

You can now build powerful clusters with a relatively small budget and keep adding extra nodes based on computing needs. Almost every industry needs fast processing power. With the increasing availability of cheaper and faster computers, more and more companies are interested in exploiting the technological benefits. There is no upper boundary to the needs of computer processing power, even with the rapid increase in power; the demand is considerably more than what is currently available.

InfiniBand adapters and switches deliver industry-leading performance in a cluster interconnect, allowing organizations to gain maximum performance advantage and return from their investment in clustered systems. Therefore IBM is focused on promoting, enabling, and delivering the software components needed to support an InfiniBand adapters and switches for System p running SLES9 environments.

5.2.2 System management software for SLES9 clustering

IBM cluster system management software stack for SLES9 is a set of software tools that provides the ability to make the cluster system management easier. CSM provides the ability of single point of control for the entire cluster, for installing, monitoring, and maintaining your cluster.

GPFS provides a high performance shared file system, which provides applications running on multiple nodes with parallel and concurrent access to the shared storage.

RSCT is the infrastructure used by a variety of IBM products to provide clusters with improved system availability, scalability, and ease of use.

We describe the software packages focusing on the enhancement for using InfiniBand in following sections. For more information about these software packages, visit the IBM Cluster Information Center at:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp>

Cluster System Management (CSM)

When using InfiniBand in an AIX cluster, CSM helps with configuring InfiniBand switches and HCAs, and monitoring the status of HCAs in nodes. Please refer to Chapter 4, “InfiniBand on AIX 5L” on page 73 for more information.

For a SLES9 cluster, CSM can help with the following tasks:

- ▶ Configuring InfiniBand switches when using CRHS in the cluster
HMC(s) can manage and monitor the switches. Refer to Appendix B, “Cluster Ready Hardware Server” on page 273 for more information.
- ▶ Installation of InfiniBand device driver packages on nodes
The secondary adapter configuration is not supported in the current CSM Version 1.5.1; however, the installation of InfiniBand device driver packages can be simplified by using the CSM software management services (SMS) feature. The required RPM packages may be placed in the appropriate directory on the CSM management server and then the **smsupdatenode** command can be used to install software on the nodes.
- ▶ Monitoring the status of HCAs
IB adapters can be monitored on cluster nodes using the **csmdat** command.

For additional information about CSM, check the following URL:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.pe.doc/pebooks.html>

Reliable Scalable Cluster Technology (RSCT)

RSCT is a set of software components that provide a comprehensive clustering environment for AIX and Linux (SLES 9, but not limited to). RSCT provides the infrastructure services for a cluster, and a common abstraction for the resources of the individual system or the cluster of nodes.

You can use RSCT resource monitoring and control (RMC) system for global access to subsystems and resources throughout the cluster, thus providing a single monitoring/management infrastructure for clusters.

InfiniBand switches and HCAs are managed and monitored as network resources by RSCT directly or through upper layer software, such as CSM and LoadLeveler.

- ▶ When using CRHS for a cluster, RSCT is needed to create the peer domain for the HMCs. InfiniBand switches are defined as hardware elements by using the RMC command `mkrhws`. The configuration then is added to the peer domain database, so that InfiniBand switches are managed and monitored by the HMC. Refer to Appendix B, “Cluster Ready Hardware Server” on page 273 for information about how to define InfiniBand switches in a CRHS environment.
- ▶ RSCT is also used to create a peer domain for nodes with HCA(s). The configuration information of HCA(s) is saved in the database of the peer domain. IBM TWS LoadLeveler then uses RSCT RMC APIs to support dynamic adapter configuration for InfiniBand, and monitors HCA ports status by using TWS LoadLeveler APIs, and commands such as `llstatus` and `llsummary`.
- ▶ RSCT monitors the InfiniBand adapters (and not only) through topology services. Topology services send heartbeat packets between adapters in the nodes belonging to the same peer domain.
- ▶ By using RSCT group services, an upper layer software (LAPI, for example) can improve communication availability. It is possible to quickly determine when an adapter’s ability to communicate has ceased, and can then “fail over” all communication to the remaining links.

Group services are also able to determine when an adapter’s ability to communicate has resumed, and can then restore communication to the initial links.

For more information about RSCT, visit the following URL:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.rsct.doc/rsctbooks.html>

General Parallel File System (GPFS)

GPFS is a high performance shared-disk file system that can provide data access from nodes in a cluster environment. GPFS relies on a network to transfer large amounts of data to/from nodes not directly connected to SAN storage through FC adapter(s). This is specially critical for HPC applications that often require low latency for communication, not just high bandwidth.

GPFS V3.1 on SLES9 supports InfiniBand by using IPoIB protocol to get higher bandwidth. This is not different than IP over Ethernet when installing and customizing GPFS environment. Refer to the following URL for more information about GPFS:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks.html>

5.2.3 Software packages for High Performance Computing

Message Passing Interface (MPI) is one of the most used methods for parallel computing. IBM Parallel Environment (PE) provides optimized compatible communication functions, library, and a runtime environment for MPI applications. To perform data communication and for optimal performance, PE interfaces with low-level application programming interface (LAPI).

LAPI is a message passing API that provides a one-sided communication model. LAPI interfaces with a lower level protocol, running in the user space (User Space protocol), which offers a low latency and high bandwidth communication path to user applications, running over a high performance switched networking infrastructure (InfiniBand in SLES9). Alternatively, LAPI also interfaces with the IP layer. Programers can use LAPI with or without the parallel operating environment (POE). POE is a component of the IBM Parallel Environment (PE) licensed program.

IBM Tivoli Workload Scheduler (TWS) LoadLeveler provides various ways of scheduling and managing applications for best performance and most efficient use of resources. LoadLeveler manages both serial and parallel jobs over a cluster of machines or servers.

In the following sections, we describe LAPI, PE, and LoadLeveler, focusing on the enhancements for using InfiniBand. For more information about these HPC software products, visit the IBM Cluster Information Center at:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp>

Low-level application programming interface (LAPI)

Note: A SLES 9 implementation of LAPI is now shipped with the Parallel Environment (PE) for Linux. This is a licensed program different from the AIX implementation, where LAPI is a part of AIX5L.

When using InfiniBand, a SLES9 implementation of LAPI includes almost all of the same features and functions as the RSCT LAPI supporting HPS on AIX5L. These features and functions (InfiniBand related) include:

- ▶ Support the user space (US) protocol.

LAPI provides optimal communication performance on the InfiniBand switch, which works in conjunction with SLES9 on a specific family of IBM System p servers. LAPI provides the following basic functions for optimal performance:

- Monitoring adapter status.

Adapter status monitoring depends on the PNSD for SLES9, and the group services component of RSCT. Figure 5-3 illustrates the relationship among LAPI, PNSD, and RSCT group services.

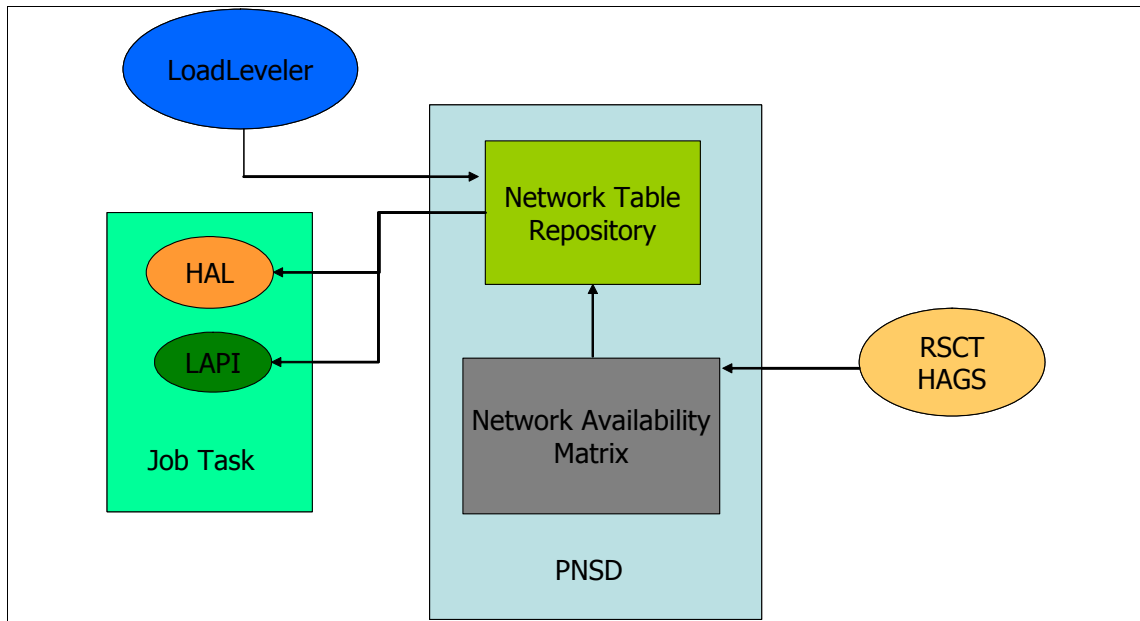


Figure 5-3 The relationships with PNSD¹ and HPC stack

RSCT group services component (GS) updates adapter status in the PNSDs of the nodes within a given peer domain.

PNSD serves as a repository for network availability matrix (NAM) and network table (NTBL) information. For each task in a job, the NTBL captures location and resource usage information. Window resources on the adapter are reserved by making NTBL API calls. These calls are supported with a separate NTBL API library that interacts with the PNSD.

¹ Protocol Network Services Daemon

Window resources can be queried and network tables can be loaded and unloaded using other API calls. The acquiring of window resources and the loading and unloading of network tables, operations that require root privileges, are typically performed by a resource manager such as TWS LoadLeveler. The network table is also queried by the running of task jobs at startup to determine the overall task geometry. The NAM in LAPI for SLES9 serves as a repository for connection status information that is reported by RSCT group services and also as a notification agent for triggering LAPI's failover and recovery function.

RSCT sends network status to the PNSD, which then sends this information to LAPI. All transactions are logged in the PNSD log file `/tmp/serverlog`. Older data is stored in the `/tmp/serverlog.old` file. You need to log in as root to access these files. Using the information in the PNSD log file, you can:

- Check the adapter status as reported by RSCT.
- Determine which job IDs are running.
- Determine which resources are allocated.

For example, job timeouts, or TWS LoadLeveler errors could signal that an adapter is down. To verify this, you can check the PNSD log file to see what RSCT has reported on the adapter state.

– Use of multiple adapters.

In systems with multiple adapters on each node, if jobs are launched with settings to request use of multiple adapters within the job tasks, when an adapter fails, LAPI switches communication over to an available adapter. If the original adapter status is restored, LAPI resumes the use of this adapter for communication for the remainder of the job run.

Unlike LAPI on AIX, the SLES9 implementation of LAPI does not include or support the following features and functions:

- ▶ The shared memory kernel extension
- ▶ Bulk message transfer
- ▶ Checkpoint and restart operations
- ▶ Data striping
- ▶ Job preemption
- ▶ User-initiated RDMA transfer

In most instances, users will not use LAPI directly. MPI is a more popular parallel method, hence the code is “parallelized” using PE library, and the jobs are managed using TWS LoadLeveler, which interfaces with LAPI. For more information about LAPI, refer to the following URL:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.help.rsct.doc/rsct_books/rsct_lapi_prog_guide/b151pg03.html

IBM Parallel Environment (PE)

For optimal performance, PE supports the User Space (US) protocol for providing communication path(s) when using System p with InfiniBand running SLES9. However, PE also lets you run parallel applications that use the IP interface of LAPI.

The user space interface allows applications to take full advantage of the high speed InfiniBand, and you should use it whenever communication is a critical issue. With LoadLeveler, the user space interface can be used by more than one process per node at a given time.

To fully utilize the InfiniBand, you can set and adjust the PE environment variables shown in Table 5-3.

Table 5-3 PE environment variables for related to InfiniBand

Environment variable/ command-line flags	Set	Possible values for InfiniBand	Default value
MP_DEVTYPE -devtype	Specifies the device type class. Note that you can only specify a single device type per parallel job; device types may not be mixed.	One of the following strings: ib InfiniBand	None
MP_EUIDEVICE -euidevice	The adapter set to use for message passing. Valid values are an adapter device name or network type, configured by LoadLeveler. You can also specify sn_all or sn_single for the InfiniBand.	One of the following strings: sn_all sn_single Or: <i>adapter device name or network type string</i> (as configured in LoadLeveler)	The adapter set used as the external network address

Environment variable/ command-line flags	Set	Possible values for InfiniBand	Default value
MP_EUILIB -euilib	The communication subsystem implementation to use for communication either with the IP communication subsystem or the User Space communication subsystem.	These values are case-sensitive: ip us	ip
MP_INFOLEVEL -infolevel -ilevel	The level of message reporting.	One of the following integers: 0 Error. 1 Warning and error. 2 Informational, warning, and error. 3 Informational, warning, and error. Also reports high-level diagnostic messages for use by the IBM Support Center. 4, 5, 6 Informational, warning, and error. Also reports high- and low-level diagnostic messages for use by the IBM Support Center.	1
MP_INSTANCES -instances	The number of instances of User Space windows or IP addresses to be assigned. This value is expressed as an integer, or the string max. If the values specified exceeds the maximum allowed number of instances, as determined by LoadLeveler, that number is substituted.	A positive integer, or the string max.	1

For more information about PE, check the following Web page:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.pe.doc/pebooks.html>

IBM Tivoli Workload Scheduler LoadLeveler V3.3.2

The TWS LoadLeveler product has been extended to include support for InfiniBand adapters in System p with SLES 9 clusters. InfiniBand adapters on any other platforms are not yet supported. This support does not place new constraints on the submit nodes or where the central manager (CM) runs.

Checkpoint/restart and preemption functions are not supported on TWS LoadLeveler for SLES9 when running in an InfiniBand cluster environment. For more information about addition of support for InfiniBand adapters in TWS LoadLeveler, see the following URL:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.loadl.doc/doc_updates/113_3.2update.html

The following features are supported on TWS LoadLeveler with InfiniBand:

- ▶ Provide for a virtual InfiniBand adapter to act as a repository for the adapter resource.

A new adapter class is defined for an InfiniBand adapter. Adapter resources will be equally divided among ports and statically allocated to ports. The InfiniBand adapter determines whether an adapter port is in use by a job that wants it not shared. The adapter port is considered in use exclusively if any adapter port on the same adapter is in use exclusively.
- ▶ Provide for managing ports of InfiniBand adapters.

An InfiniBand adapter port class is defined for a port of an InfiniBand adapter. The InfiniBand adapter port class will inherit all members and member functions. LoadLeveler will create an InfiniBand adapter port at start-up when a port of an InfiniBand adapter is configured.
- ▶ Support for dynamic adapter configuration.

LoadLeveler uses the RMC API to support dynamic adapter configuration. If an OSI (machine stanza) in the admin file does not contain any adapters, then LoadLeveler will perform the dynamic adapter configuration to determine which adapters are present at startup through the RMC API. For Startd and Starter, if an IB adapter ports exists, then the dynamic adapter function will create an LIDynamicMachine to contain and maintain RMC adapter data. InfiniBand port objects will need a unique ID constructed from an IB unique ID plus port number.
- ▶ Provide information to POE (Parallel Operating Environment) for running a job

When POE is notified that the job is ready to start, it queries each task instance of the job for the configured adapter port, and starts using it. POE retrieves the device name, window number, interface address, port number, and switch table tag from the adapter usage object.

When using IBM TWS LoadLeveler with InfiniBand, some external interfaces (LL configuration file, APIs, and commands) are used or modified. These are:

- ▶ New Adapter Stanza defined in administration file LoadL_admin.

A new keyword, `port_number`, is added for an InfiniBand adapter port. The `port_number` specifies the port number of the InfiniBand adapter port. This value is set by the `llxtRPD` command and should *NOT* be changed manually. The `adapter_name`, `type`, `network_type`, `interface_address`, `interface_name`, `multiple_address`, `logical_id`, `adapter_type`, `device_driver_name`, and `network_id` of the IB adapter port are specified for the adapter. The adapter stanza for InfiniBand support only contains the adapter port information, as shown in Example 5-10.

Example 5-10 Adapter stanza for InfiniBand in LoadL_admin file

```
lib01sw1: type = adapter
          adapter_name = ib0
          network_type = InfiniBand
          interface_address = 192.168.8.161
          interface_netmask = 255.255.255.0
          interface_name = lib01sw1
          logical_id = 24
          adapter_type = InfiniBand
          device_driver_name = ehca0
          network_id = 18338657682652659714
          port_number = 1
```

- ▶ The `llstatus` command is updated to provide the information about InfiniBand ports.

The `-a` and `-l` options will display the port number for each InfiniBand adapter. The `llstatus` command will not show port information for non-IB adapters. The information format is:

```
name (network_type, interface_name, interface_address,
      multilink_address, nodeid,available_windows/total_windows,
      available_rcxt/total_rcxt rCxt Blks,connectivity_array, state,
      port_number)
```

For example:

```
ib0 (InfiniBand,lib01sw1,192.168.8.161,,1,59/64,0/0 rCxt
Blks,1,READY,1)
```

- ▶ Keywords *network* and *max_protocol_instances* in LoadLeveler Job Command File (JCF).

You need to specify the *network* keyword when you want a task of a parallel job step to request an InfiniBand adapter that is defined in the LoadLeveler administration file. The format of network keyword is:

```
network.protocol = type, usage, mode, comm_level,  
instances=<number|max>, rcxtblocks=number
```

For example:

```
# @ network.mpi = sn_all,shared,us,,instances=1
```

The keyword *max_protocol_instances* is used to specify the maximum number of instances in the network statement above.

- ▶ LoadLeveler API `ll_get_data` is updated to retrieve the port number and lmc on an IB adapter port

The LoadLeveler API specification enum is updated with new data members, `LL_AdapterUsagePortNumber`, `LL_AdapterUsageLmc`, `LL_AdapterPortNumber`, and `LL_AdapterLmc`. New data members can be used with the `ll_get_data` call to retrieve the port number and lmc on an InfiniBand adapter port.

5.3 Installation and configuration

In this section, we will go through the steps of setting up a SLES9 cluster with InfiniBand interconnect. For most software components, you can find detailed information in the installation guide(s). However, the steps listed in this section may not be identical with the sequence shown in the installation guides. We cover the following topics:

- ▶ Planning
- ▶ Sample SLES 9 Cluster layout and description
- ▶ Installation steps for setting up a SLES 9 cluster

5.3.1 Planning

When planning a System p cluster with InfiniBand and SLES 9, you have to bring together several management tools and various cluster software. Major components to consider are:

- ▶ Frames (racks)
- ▶ Servers
- ▶ I/O devices
- ▶ InfiniBand network devices

- ▶ Service network, including:
 - HMCs
 - Ethernet devices
 - CSM Server (for multiple HMC environments)
 - Linux distribution server (for servers with no removable media)
- ▶ System management applications, including:
 - HMC
 - CSM
 - RSCT
 - IBM Network Manager
- ▶ Other cluster software packs, including:
 - GPFS
 - LAPI and PE
 - LoadLeveler

Required firmware and device drivers levels

Refer to 3.5, “Supported System p servers” on page 40 for detailed information about the hardware requirements needed to support an InfiniBand cluster solution. For the latest information, see the Facts and Features Web site at:

<http://www.ibm.com/servers/eserver/clusters/hardware/factsfeatures.html>

Required cluster software versions

Refer to 5.1.2, “Software components and versions” on page 143 for information about software packages and required levels.

Planning the InfiniBand switch configuration

InfiniBand switches require a custom configuration to function in an IBM System p cluster. The configuration settings that require planning are shown in the following sections.

IP addressing on the service Ethernet network

You must determine if the IB management port (connected to the service Ethernet) will use static IP addressing or DHCP. If you have only one HMC and you are not using CRHS mode in the cluster, the InfiniBand Switch should be a DHCP client.

Note: You must assign a static IP address for switches when you are using CSM and Cluster Ready Hardware Server. This is mandatory when you have multiple HMCs in the cluster.

GID-prefix

Each subnet in the InfiniBand network must be assigned a GID-prefix, which will be used to identify the subnet for addressing purposes, and within IBM Network Manager. The GID-prefix is an arbitrary assignment with a format of 16 hexadecimal digits: xx:xx:xx:xx:xx:xx:xx:xx. (for example, FE:80:00:00:00:00:00:01).

LMC value

If you are using a switch in an IBM HPC cluster, you will need to set the LMC value to 2.

Switch name

You should assign a name for each switch. This is an arbitrary assignment, but we recommend that you use some sort of convention, so that you can easily identify each switch.

Considerations for 12x adapters cabling

If you need to connect 12x HCAs to a 4x switch, you will require special IBM octopus cables. These cables allow to connect three 4x switch ports to one 12x HCA port.

Planning an IBM GX HCA configuration

An IBM GX Host Channel Adapter (HCA) requires certain configuration settings to work in an IBM system p InfiniBand cluster. These are shown in the following sections.

GUID index

Each physical InfiniBand HCA contains a set of 64 globally unique IDs (GUIDs) than can be assigned to partition profiles. These are used to address Logical HCA (LHCA) resources on a physical HCA (IB adapter). You can assign multiple GUIDs to each profile, but you can assign only one GUID from each HCA to each partition profile. Each GUID can be used by only one logical partition at a time.

You can create multiple partition profiles with the same GUID, but only one of those partition profiles can be activated at a time. The GUID index is used to choose one of the 64 GUIDs available for an HCA. It can be any number from one (1) through 64. Quite often you will assign a GUID index based on which LPAR and profile you will be configuring.

Capability

The Capability setting is used to indicate the level of sharing to be done. This can be one of the following:

1. Low
2. Medium
3. High
4. Dedicated

GID-prefix for each HCA port

The GID-prefix for a port is not something that you explicitly set, but it is important to understand the IB subnet to which a port attaches. This is determined by the switch to which the HCA port is connected. The GID-prefix is actually configured for the switch.

Refer to 2.10, “Adapter sharing” on page 26 for more information about the configuration settings for IBM GX HCA. Additional information can be found at:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp?topic=/iphau/referenceinfiniband.htm>

Management subsystem planning

An HMC is required to manage the LPARs and to configure the GX bus HCAs in the servers, as well as to run the IBM Network Manager (management software for the IB fabric). The maximum number of servers (CECs²) that can be managed by an HMC is 32. When you go beyond 32 servers, additional HMCs are required.

If you have a single HMC in the cluster, this is usually configured to be the DHCP server for the service network. However, if Cluster Ready Hardware Server (CRHS) and CSM are being used in the cluster, the CSM Management Server should be configured as the DHCP server for the service network, and CRHS must be configured to recognize the servers, BPAs, HMCs, and InfiniBand switches in the cluster.

If you require more than one HMC to manage your cluster servers and switches, you will need to use CSM on a Management Server, and you must configure CRHS. You also need to assign a static IP address for each HMCs. This is mandatory when you have multiple HMCs in the cluster.

² Central Electronic Complex - identified by a Hypervisor connected to the HMC

Note: We recommend that you have two service networks connected to different Ethernet switches/VLANs (IP subnets too) to support redundancy for IBM servers, BPCs, and HMCs.

However, the InfiniBand switches only support a single service network (even though some InfiniBand switch models have multiple Ethernet connections).

For more information about management subsystem planning, refer to the following Web page:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.csm.doc/clusterbooks.html>

5.3.2 Sample SLES 9 Cluster layout and description

We use a sample SLES 9 cluster as described in 5.3.3, “Installation steps for setting up a SLES 9 cluster” on page 170 to demonstrate the processes of setting up a SLES 9 cluster. Refer to 4.3, “Test cluster layout and description” on page 94 for the system architecture we have used. In this sample environment, the only difference between the AIX cluster and the SLES 9 cluster is that we use an OpenPower720 as the CSM management server. Table 5-4, Table 5-5 on page 170, and Table 5-6 on page 170 contain the planning worksheets for service network and public network for our test cluster running SLES 9.

Table 5-4 HMC planning worksheet

Network type	Service network	Public network
Host name	hmcib01p	hmcib01p
Ethernet adapter	eth0	eth1
IP address	10.10.10.28	192.168.100.28
Netmask	255.255.255.0	255.255.255.0

Table 5-5 CSM Management server planning worksheet

Model: OpenPower 720		
OS: SLES9 SP3		
Serve as DHCP and Installation server? Yes		
Network type	Service network	Public network
Host name	msib02p	msib02
Ethernet adapter	eth0	eth1
IP address	10.10.10.40	192.168.100.30
Netmask	255.255.255.0	255.255.255.0

Table 5-6 Computing nodes planning worksheet

Model: <u>System p55A</u>								
OS: <u>SLES9 SP3</u>								
Service network domain: <u>10.10.10</u>								
Service network port: <u>HMC port-1</u>								
Public network domain: <u>192.168.100</u>								
Public network Ethernet adapter: <u>eth3</u>								
IPoB network domains: <u>Port 1: 192.168.8 Netmask: 255.255.255.0</u>								
<u>Port 2: 192.168.9 Netmask: 255.255.255.0</u>								
Node #	Service Network Service Processor		Public Network Eth0		InfiniBand Network Port 1		Port 2	
	Host name	IP	Host name	IP	Host name	IP	Host name	IP
1	lib01p	161	lib01	161	lib01sw1	161	lib01sw2	161
2	lib02p	162	lib02	162	lib02sw1	162	lib02sw2	162
3	lib03p	163	lib03	163	lib03sw1	163	lib03sw2	163
4	lib04p	164	lib04	164	lib04sw1	164	lib04sw2	164

5.3.3 Installation steps for setting up a SLES 9 cluster

This section describes the steps we have used to set up a SLES 9 cluster with InfiniBand in our test lab. Our cluster consists of four (4) compute nodes, as described in 4.3, “Test cluster layout and description” on page 94. Nevertheless, the same steps also apply for installing a larger cluster system, with up to 96 compute nodes.

Before you install the cluster software stack

Before you begin with the installation steps in this section, ensure that you have completed the following:

- ▶ You should have a clear understanding of the system architecture you want to set up. Start with a cluster diagram and a network planning worksheet for your system. Refer to 4.3, “Test cluster layout and description” on page 94 and 5.3.2, “Sample SLES 9 Cluster layout and description” on page 169.
- ▶ All hardware units are on site and ready to be used. This includes:
 - Installation of HMC. In the test environment, we use only one HMC. This can be set up as a stand-alone system or CRHS with CSM. To make the installation steps apply for multiple HMCs systems, we have used the CRHS with CSM option. Hence, it necessary to set up a CRHS environment on the HMC. A series of “how to” steps for this type of configuration follows; for detailed information, see also:
<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.csm.doc/clusterbooks.html>
 - Set the static IP addresses for the Ethernet adapters on the HMC for both the service network and the public network.
 - Enable SLP and RSCT on the interface configured on the service network.
 - Preparation of the CSM Management Server.
 - Installation and configuration of SLES9 SP3 on the Management Server.
 - Set static IP addresses for service network and public network.
 - Set up the DHCP server on the CSM Management server for the service network.

Note: In the test scenario, the MS acts as SLES 9 installation server at the same time. It will be also configured as a DHCP server for public network during the process of installing CSM.

- The steps for setting up the InfiniBand switches are:
 - Assign a static IP addresses for each InfiniBand switch.
 - Assign the GID prefix for the InfiniBand switch to identify the subnet of which the switch is a member.
 - Set the LMC value to 2, if your cluster system is designed for running HPC applications.

Refer to 3.11, “IBM Network Manager (IBM NM)” on page 59 for details.

- Prepare the compute nodes, Ethernet switches, and cables.
 - Connect Ethernet cabling to the service network and public network (compute node CECs, HMCs, IB switches, and CSM MS).
 - Connect the compute nodes' InfiniBand cables with InfiniBand switches.
 - Connect the compute nodes' power cables. Make sure that the service processor on each node gets an IP address from the DHCP server.

Steps for setting up a SLES 9 cluster environment

- ▶ “Step 1. Compute node installation using CSM” on page 172
- ▶ “Step 2. Configuration of InfiniBand adapters on nodes” on page 179
- ▶ “Step 3. GPFS installation and configuration” on page 183
- ▶ “Step 4. Installing compilers” on page 188
- ▶ “Step 5. PE installation and testing” on page 188
- ▶ “Step 6. Creating an RSCT Peer Domain for compute nodes” on page 191
- ▶ “Step 7. LoadLeveler installation and configuration” on page 192

Step 1. Compute node installation using CSM

This step describes how to configure CSM and the node installation process. Not all steps are necessary to complete the installation of nodes; some are just to make the procedure easier. For a complete description, refer to the manual *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344.

Create a /data partition (Optional)

We recommend creating a separate partition for a file system mounted under /data, on the CSM management server, to store the software packages, configuration files and scripts, and so on. The size of this partition largely depends on the amount of data you plan to store. In our test environment, we decided to store the ISO image file of SLES9 installation CDs (including three images for SLES 9 SP3). Thus, we created an 8 GB partition for the /data file system. For our environment, the directory structure is shown in Table 5-7.

Table 5-7 Directory structure of /data NFS file system

Directory	Contents
/data/RPMs	Non-IBM software
/data/Sles9_CD	ISO images for SLES9 and service packs
/data/0058_driver	Device driver for IBM InfiniBand adapter
/data/scripts	Some home grown scripts

Directory	Contents
/data/tmp	Temporary files

Downloading non-IBM software

Download the required non-IBM software, and store the packages in the /data/RPMs directory (see the CSM installation instructions). These packages are:

- ▶ autoupdate 5.4.1 (or higher)
- ▶ openCIMOM 0.8
- ▶ perl-RPM2

Register the IP labels

Attention: IP name resolution (direct and reverse) is critical to cluster operation, from management operations to running applications. It is *MANDATORY* that all nodes in the cluster resolve the IP address/names *IDENTICALLY*.

We also recommend setting the name service switch (NSS) resolution order in /etc/nsswitch.conf to be identical on all nodes in the cluster.

Add the IP addresses and labels for your cluster nodes to the /etc/hosts file on the management server. They are used by CSM for resolving the IP address for each node during the process of installing nodes. We have configured our /etc/hosts as shown in Example 5-11.

Example 5-11 The /etc/hosts file on the management server

```
# Management Server
10.10.10.40    msib02.itso.ibm.com    msib02p #service network
192.168.100.30 msib02.itso.ibm.com    msib02  #public network

# HMC
10.10.10.28    hmcib01p
192.168.100.28 hmcib01

# Computing Nodes
192.168.100.161 lib01.itso.ibm.com    lib01
192.168.100.162 lib02.itso.ibm.com    lib02
192.168.100.163 lib03.itso.ibm.com    lib03
192.168.100.164 lib04.itso.ibm.com    lib04
```

Create the /csminstall partition

We also recommend creating a separate partition for the /csminstall file system, on the management server, to store the CSM packages. The size of this partition is also 8 GB.

Install CSM packages and accept the license agreement

Mount the CSM CD and install the csm.core package:

```
msib02:~ # mount /dev/cdrom /media/cdrom
msib02:~ # rpm -ivh /media/cdrom/csm.core-*
```

Before running the **installms** command to install the CSM software, we also uploaded the SLES9 ISO images into the directory /data/Sles9_CD/GA, and SLES 9 SP3 ISO images into the directory /data/Sles9_CD/SP3, as shown in Example 5-12.

Example 5-12 ISO image files for SLES9 and SP3

```
msib02:~ # ls /data/Sles9_CD/GA
.          SLES-9-ppc-RC5-CD3.iso
..         SLES-9-PPC-RC5-CD4.iso
SLES-9-ppc-RC5-CD1.iso SLES-9-ppc-RC5-CD5.iso
SLES-9-ppc-RC5-CD2.iso SLES-9-ppc-RC5-CD6.iso

msib02:~ # ls /data/Sles9_CD/SP3
. .. 1.iso 2.iso 3.iso
```

Run the following command to install the CSM software:

```
msib02:~ # installms -p \
/media/cdrom:/data/RPMs:/data/Sles9_CD/GA:/data/Sles9_CD/SP3
```

The command **installms** gets non-IBM packages from /data/RPMs, and installs required packages directly from ISO images. In this way, you do not have to change SLES9 CDs manually.

To accept the CSM license, run the following command, and follow the directions to accept the CSM license at the prompt:

```
msib02:~ # csmconfig -L /media/cdrom/csm1um.full
```

For HMC-attached System p hardware, issue the **systemid** command to store hardware control point user ID(s) and password(s):

```
msib02:~ # systemid hmcib01 hscroot
```


To verify that the management server has been installed correctly and is ready for use, we strongly recommended you run the `ibm.csm.ms` probe and make sure the probe is run successfully. To run the probe, issue the following command:

```
msib02:~ # probemgr -p ibm.csm.ms -l 0
```

Set up the CRHS environment

Note: Setting up the CRHS environment is mandatory if you have more than one HMC in your cluster.

Refer to Appendix B, “Cluster Ready Hardware Server” on page 273 for the steps needed to set up the CRHS environment. We strongly recommend you use the CRHS feature for your cluster using InfiniBand. You should enable IBM Network Manager (NM) on HMC after the CRHS environment is ready so that you can manage and monitor InfiniBand switches.

Define managed (compute) nodes in the cluster

First, we use the `lshwinfo` command to gather managed nodes information, as shown in Example 5-13.

Example 5-13 Hardware information of managed nodes

```
msib02:~ # lshwinfo -p hmc -c hmcib01 -o /data/tmp/hwinfo.txt
msib02:~ # cat /data/tmp/hwinfo.txt
Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParID::HWType:
:HWModel::HWSerialNum::DeviceType::UUID
no_hostname::hmc::hmcib01::ib01::2::9131::52A::10391FG:::
no_hostname::hmc::hmcib01::ib02::2::9131::52A::10391EG:::
no_hostname::hmc::hmcib01::ib03::2::9131::52A::10391DG:::
no_hostname::hmc::hmcib01::ib04::2::9131::52A::103920G:::
```

Modify the file `/data/tmp/hwinfo.txt`, and replace the “no_hostname” field with the IP labels of your managed nodes, as shown in Example 5-14.

Example 5-14 Hardware information of managed nodes after being modified

```
msib02:~ # cat /data/tmp/hwinfo.txt
Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParID::HWType:
:HWModel::HWSerialNum::DeviceType::UUID
lib01.itso.ibm.com::hmc::hmcib01::ib01::2::9131::52A::10391FG:::
lib02.itso.ibm.com::hmc::hmcib01::ib02::2::9131::52A::10391EG:::
lib03.itso.ibm.com::hmc::hmcib01::ib03::2::9131::52A::10391DG:::
lib04.itso.ibm.com::hmc::hmcib01::ib04::2::9131::52A::103920G:::
```

To define the managed nodes, run the following command:

```
msib02:~ # definenode -M /data/tmp/hwinfo.txt
```

We recommend creating a nodegroup for the managed nodes by using the **nodegrp** command:

```
msib02:~ # nodegrp -a lib01,lib02,lib03,lib04 libnodes
```

Check the nodes in the node group (see Example 5-15).

Example 5-15 Nodegroup libnodes

```
msib02:~ # nodegrp libnodes
lib01.itso.ibm.com
lib02.itso.ibm.com
lib03.itso.ibm.com
lib04.itso.ibm.com
```

At this time nodes are in “PreManaged” mode, and you can use the **lnode -a Mode** command to verify them. Next, you can check the power status of the managed nodes by running the **rpower** command, as shown in Example 5-16.

Example 5-16 Example of command rpower

```
msib02:~ # rpower -a cec_query
lib03.itso.ibm.com on
lib04.itso.ibm.com on
lib02.itso.ibm.com on
lib01.itso.ibm.com on
```

```
msib02:~ # rpower -a query
lib03.itso.ibm.com off
lib04.itso.ibm.com off
lib02.itso.ibm.com off
lib01.itso.ibm.com off
```

Setting up a SLES9 distribution server

We use the management server as the SLES9 distribution server in our test cluster. If you intend to set up separate SLES9 distribution servers, you should define the servers as cluster members (nodes) first, and have the SLES9 operating system installed and available on the servers. You should also modify the attributes of managed nodes accordingly using the **chnode** command. For more information about how to set up separate distribution servers, check the following URL:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.csm.doc/clusterbooks.html>

To copy SLES9 distribution(s) and service level(s) from CDs to the management server's /csminstall directory, run the following commands:

```
msib02:~ # copycds -A InstallDistributionName=SLES
InstallDistributionVersion=9 InstallPkgArchitecture=ppc64
InstallServiceLevel=GA InstallOSName=Linux -p /data/Sles9_CD/GA
msib02:~ # copycds -A InstallDistributionName=SLES
InstallDistributionVersion=9 InstallPkgArchitecture=ppc64
InstallServiceLevel=SP3 InstallOSName=Linux -p /data/Sles9_CD/SP3
```

By using the -p option, the **copycds** command will copy the files from each ISO image under the appropriate directories, as though copying from a CD, and mount one ISO image at a time (**mount -o loop=/dev/loop0 ...**).

Installation of managed nodes

Before you proceed with the installation, you should configure AutoYaST and define the nodes to be installed in the CSM database. CSM provides the **csmsetupyast** command to complete this task.

Run the following command to configure AutoYaST and define the nodes to be installed in the CSM database. Be sure to use the -x option for the command line to avoid copying the SLES9 disks again, since this procedure has been completed by the **copycds** command.

```
msib02:~ # csmsetupyast -x -a
```

This command creates the AutoYaST template files for the nodes in the cluster. These template files are in XML format, and are saved into the /csminstall/csm/1.5.1/autoyast.SLES9 directory. You can alter the configuration templates for your nodes, for example, to create a user during the installation process. For more information about the AutoYaST configuration file, check the following Web page:

<http://yast2.suse.com/autoinstall/>

The **csmsetupyast** command also defines the nodes to be installed in the CSM database, and sets up the DHCP server on the SLES9 distribution servers at the same time. Check the /etc/dhcpd.conf to verify the specific information for each node.

CSM provides a probe (ibm.csm.install-server) to verify that the SLES9 distribution server has been installed and configured correctly. We strongly recommend running this probe before installing nodes:

```
msib02:~ # probemgr -p ibm.csm.install-server -l 0
```

CSM provides a probe `ibm.csm.node-install` to verify that the managed nodes are ready to be installed. We strongly recommend running this probe before installing the nodes:

```
msib02:~ # probemgr -p ibm.csm.node-install -a
```

Once these two probes are completed successfully, you can begin with the installation of nodes. Run the following command to install the nodes:

```
msib02:~ # installnodes -a
```

You can open several consoles to monitor the installation process for part or all nodes. Using the following command, for example, we monitor the statuses of node `lib01` and `lib03`:

```
msib02:~ # rconsole -t -r -n lib01,lib03
```

Note: If you want to open more than one console at a time, make sure you are running the previous command (`rconsole`) in a GUI (X Window System) environment. If there is no X Window System environment available, you can monitor only one node in a terminal window.

You can also use the `monitorinstall` command to monitor the status of all nodes during the install process, as shown in Example 5-17 (the `watch` command is used to execute the `monitorinstall` command every two seconds).

Example 5-17 Monitoring installation

```
msib02:~ # watch monitorinstall
Every 2.0s: monitorinstall          Fri Oct 13 13:01:56 2006
```

Node	Mode	Status
lib01.itso.ibm.com	Managed	Rebooting and Installing
lib02.itso.ibm.com	Managed	Rebooting and Installing
lib03.itso.ibm.com	Managed	Installed
lib04.itso.ibm.com	Managed	Installed

After the installation for all nodes has completed, you may run the `csmdat` command or use a `dsh` command to verify that all the nodes are available, as shown in Example 5-18 on page 179.

Example 5-18 Verify if all nodes are available

```
msib02:~ # dsh -a date
lib04.itso.ibm.com: Fri Oct 13 14:11:34 EDT 2006
lib02.itso.ibm.com: Fri Oct 13 14:11:33 EDT 2006
lib03.itso.ibm.com: Fri Oct 13 14:11:32 EDT 2006
lib01.itso.ibm.com: Fri Oct 13 14:11:37 EDT 2006
```

```
msib02:~ # csmstat
```

```
-----
Hostname           HWControlPoint   Status   PowerStatus Network-Interfaces
-----
lib01.itso.ibm.c~ hmcib01          on       on          eth3-Online
lib02.itso.ibm.c~ hmcib01          on       on          eth3-Online
lib03.itso.ibm.c~ hmcib01          on       on          eth3-Online
lib04.itso.ibm.c~ hmcib01          on       on          eth3-Online
-----
```

Note: You may need to configure secure shell before using the **dsh** command. For an example, refer to *Configuration and Tuning GPFS for Digital Media Environments*, SG24-6700.

Step 2. Configuration of InfiniBand adapters on nodes

Note: CSM supports the configuration of secondary network adapters during node installation or update (by using the **update node -c** command). However, on SLES 9 nodes, CSM only supports the configuration of Ethernet adapters for now. You need to configure InfiniBand adapters on nodes manually.

For the installation of the InfiniBand device driver on a SLES9 node, refer to 5.1.2, “Software components and versions” on page 143 for more information. The method presented makes the configuration of cluster nodes as easy as possible.

Before configuring the InfiniBand adapters

It is a best practice to NFS export the /data file system on the CSM management server and mount it on all computing nodes. The required software packages are stored in different directories under /data, as shown in Table 5-7 on page 172.

Installation of the InfiniBand device driver

You can download the (SLES 9) device driver packages for IBM InfiniBand HCAs from the following URL:

<http://developer.novell.com/devres/storage/drivers/index.html#IBM>

Click **0058** to download the file `openib-0058-gen2-sles9sp3.ppc.tar.gz`. For a description of the OpenIB package, refer to 5.1.2, “Software components and versions” on page 143.

Store the package in the `/data/0058_driver` directory and unpack it. In our test environment, we have also created a shell script named `inst11_dd_0058.SuSE`, as shown in Example 5-19:

Example 5-19 Script for installing the IBM IB HCA device driver

```
msib02:/data/0058_driver # cat install_dd_0058.SuSE
#!/usr/bin/bash
rpm -ivh /data/0058_driver/openib_kernel_ehca2-0058-1.ppc64.rpm
rpm -ivh /data/0058_driver/openib_kernel_ehca2-0058-1.src.rpm
rpm -ivh /data/0058_driver/openib_userspace_32-1.0RC2-1.ppc.rpm
rpm -ivh /data/0058_driver/openib_userspace_32-1.0RC2-1.src.rpm
rpm -ivh /data/0058_driver/openib_userspace_64-1.0RC2-1.ppc64.rpm
rpm -ivh /data/0058_driver/openib_userspace_64-1.0RC2-1.src.rpm
rpm -ivh /data/0058_driver/openib_userspace_ehca-0058-1.ppc.rpm
rpm -ivh /data/0058_driver/openib_userspace_ehca-0058-1.src.rpm
rpm -ivh /data/0058_driver/openib_userspace_ehca_64-0058-1.ppc64.rpm
rpm -ivh /data/0058_driver/openib_userspace_ehca_64-0058-1.src.rpm
```

We have saved a copy of this script in the `/data/scripts` directory, and then run the following command on the management server:

```
msib02:~ # dsh -a /data/scripts/install_dd_0058.SuSE
```

Configuring IPoIB for the InfiniBand adapters

To support software that cannot utilize the user space protocol directly, InfiniBand provides IP over InfiniBand (IPoIB) protocol support. In our test cluster we use IPoIB to configure GPFS on SLES9.

To configure IPoIB, you need to add following lines to the file `/etc/sysctl.conf`:

```
#ipoib tuning
net.ipv4.conf.default.arp_ignore = 2
net.ipv4.conf.default.arp_filter = 1
net.ipv4.tcp_wmem = 32768 131072 524288
net.ipv4.tcp_rmem = 32768 131072 524288
net.core.wmem_max = 1048576
net.core.rmem_max = 1048576
```

You can change these parameters on each node, one by one, or you can run the following command on the management server to change the parameters on all nodes at the same time:

```
msib02:~ # dsh -a "cat /data/tmp/sysctl.txt >> /etc/sysctl.conf"
```

The file `/data/tmp/sysctl.txt` contains the previously mentioned parameters. Also, this command assumes that the `/data` directory is NFS mounted on all managed nodes.

Next, create the following files on each node:

```
/etc/sysconfig/network/ifcfg-ib0  
/etc/sysconfig/network/ifcfg-ib1
```

These files are personalized for each node. For example:

```
BOOTPROTO='static'  
BROADCAST='192.168.8.255'  
IPADDR='192.168.8.161'  
MTU=''  
NETMASK='255.255.255.0'  
NETWORK='192.168.8.0'  
REMOTE_IPADDR=''  
STARTMODE='onboot'
```

You need to set the IP address and modify other parameters separately for each port on each node. You can do it on each node one by one, or use the script to configure all nodes at one time.

After you configured IPoIB, you should check the device driver modules configuration file `/etc/modprobe.conf.local`, as shown in Example 5-20:

Example 5-20 Device driver modules configuration file `/etc/modprobe.conf.local`

```
options hcad_mod nr_ports=2 port_act_time=120  
options ib_ipoib send_queue_size=1024 rcv_queue_size=1024
```

If your system has fewer than 32 nodes, set `send_queue_size` and `rcv_queue_size` to 1024. If your system has more than 32 nodes, set `send_queue_size` and `rcv_queue_size` to 2048.

Reboot all nodes after the configuration by using the following command:

```
msib02:~ # rpower -a reboot
```

Verifying the IPoIB configuration

After all nodes have restarted, you may use following commands to verify the status of the InfiniBand adapters:

- ▶ To display the HCA adapters and GUID, use the **ibv_devices** command, as shown in Example 5-21.

Example 5-21 Command for displaying the HCA adapters and GUID

```
lib01:~ # ibv_devices
  device                node GUID
  -----                -
  ehca0                 0002550050002700
```

- ▶ To query the InfiniBand devices, use the **ibv_devinfo** command, as shown in Example 5-22.

Example 5-22 Get GUID and subnet prefix information

```
lib01:~ # ibv_devinfo -d ehca0 -v | grep GUID
GUID[ 0]:                fe80:0000:0000:0002:0002:5500:5000:273d
GUID[ 0]:                fe80:0000:0000:0000:0002:5500:5000:277d
```

The GUID is formed by concatenating the subnet prefix with the GUID. Verify that your subnet prefixes are correct (one or two subnets). Example 5-22 illustrates a configuration with two different subnet prefixes, fe80:0000:0000:0002 and fe80:0000:0000:0000.

- ▶ To query the InfiniBand hardware version, use the **ibv_devinfo** command, as shown in Example 5-23.

Example 5-23 Display the HCA hardware level

```
lib01:~ # ibv_devinfo | grep hw_ver
hw_ver:                  0x1000003
```

Note: If your HCA hardware version is not "0x1000003", contact IBM Support.

- ▶ To check the device driver version, use the **modinfo** command, as shown in Example 5-24.

Example 5-24 Check InfiniBand device driver version

```
lib01:~ # modinfo hcad_mod | grep version
version:                 EHCA2_0058 A693B35D29EC5C569166E1F
```

For more commands and how to monitor the HCA status, refer to Chapter 8, “Monitoring tools for InfiniBand adapter” on page 249.

Modify /etc/hosts

Once IPoIB has been configured, you should modify the /etc/hosts file to add name resolution for the IP addresses associated with the IB adapters and ports. In Example 5-25 we present an excerpt from the /etc/hosts on our IB nodes.

Example 5-25 IP name resolution for IB adapters (/etc/hosts)

```
lib01:~ # cat /etc/hosts
.....
# IPoIB port-1 ib0
192.168.8.161 lib01sw1
192.168.8.162 lib02sw1
192.168.8.163 lib03sw1
192.168.8.164 lib04sw1

# IPoIB port-2 ib1
192.168.9.161 lib01sw2
192.168.9.162 lib02sw2
192.168.9.163 lib03sw2
192.168.9.164 lib04sw2
```

Step 3. GPFS installation and configuration

This section describes how to set up a GPFS cluster using InfiniBand as an interconnect. For the current version of IB stack and GPFS, only IPoIB is supported. Since IP is utilized, GPFS installation and configuration follows the same procedures as for basic GPFS cluster using an Ethernet network. For details and tuning parameters, refer to *GPFS V3.1 Concepts, Planning, and Installation Guide*, GA76-0413.

Attention: For a SLES9 setup, we have decided to use the ssh/scp remote command/copy programs readily available on Linux.

In this section we assume that remote command / copy between all GPFS nodes has been set up properly (no user interaction).

Installing GPFS package

As with the other software, we store the GPFS package in the NFS exported directory /data on the management server, under /data/HPC/GPFS, together with an installation shell script, `gpfs_install-3.1.0-0_sles9.ppc`.

Once the GPFS RPMs and install script are prepared, we execute the following commands from the management server:

```
msib02:~ # dsh -a "/data/HPC/GPFS/gpfs_install-3.1.0-0_sles9.ppc  
--text-only --silent"  
msib02:~ # dsh -a "rpm -ivh /usr/lpp/mmfs/3.1/*.*rpm"
```

The following packages should be installed (check with `rpm -qa|grep gpfs`):

```
gpfs.gpl-3.1.0-0  
gpfs.msg.en_US-3.1.0-0  
gpfs.docs-3.1.0-0  
gpfs.base-3.1.0-0
```

Install the latest GPFS service package (PTF)

To obtain the latest GPFS code, check the following Web site:

<http://www14.software.ibm.com/webapp/set2/sas/f/gpfs/download/home.html>

Download the latest corrective service package for GPFS, and store the packages (RPMs) into the `/data/HPC/GPFS/Fixes` directory on the management server. Then issue following command on the management server:

```
msib02:~ # dsh -a rpm -Uvh /data/HPC/GPFS/Fixes/*.rpm
```

Check and see if updates have been installed on all nodes:

```
msib02:~ # dsh -a "rpm -qa|grep gpfs" | dshbak -c
```

Building the GPL portability layer

Note: You do not have to compile the portability layer on all cluster nodes. If all nodes in the cluster run the same OS level, you can compile the modules on one node and the distribute (copy) the compiled modules to the rest of the nodes.

GPFS V3.1 provides a configuration script to help you build the portability layer. As GPFS manuals recommend, you should use a non-root user to build the portability layer modules from source code. We have crated an user named `ibitso`, and executed the commands (steps) shown in Example 5-26 on page 185.

Example 5-26 Building the GPL modules

```
lib01:~ # chown -R ibitso:users /usr/lpp/mmfs/src
lib01:~ # su - ibitso
ibitso@lib01:~> export SHARKCLONEROOT=/usr/lpp/mmfs/src
ibitso@lib01:~> /usr/lpp/mmfs/src/config/configure
ibitso@lib01:~> cd /usr/lpp/mmfs/src
ibitso@lib01:/usr/lpp/mmfs/src> make Autoconfig
ibitso@lib01:/usr/lpp/mmfs/src> make World
ibitso@lib01:/usr/lpp/mmfs/src> exit
lib01:~ # cd /usr/lpp/mmfs/src
lib01:/usr/lpp/mmfs/src # make InstallImages
```

Note: To make the process of building the Linux portability interface a smoother process, you should make sure that GNU C/C++ and tools for Linux have been installed on the nodes.

Creating the GPFS cluster

Before you define the GPFS cluster, you must check if you can execute remote commands (**ssh**) from any node in the cluster to all nodes in the cluster (including local node), over all interfaces that will be defined in the cluster.

To define the GPFS cluster nodes you have to create a node definition file. The node definition file contains all the nodes in the cluster, identified by the IP label of the adapter that will be used for GPFS communication and the node role, one per line. In our case, we use the ib0 adapter IP label for GPFS communication. The node descriptor file we have used to create our cluster is shown in Example 5-27.

Example 5-27 NodeFile for GPFS cluster

```
lib01:~ # cat /data/tmp/mmnodefile
lib01sw1:quorum:lib01
lib02sw1:quorum:lib02
lib03sw1:quorum:lib03
lib04sw1::lib04
```

We selected lib01, lib02, and lib03 as quorum nodes, use lib01 as the primary cluster data server and lib02 as the secondary cluster data server. Also, we decided to use the nodes' administrative interface for GPFS remote copy/command execution. This is why the first field in the file represents the IP label of the ib0 interface and the last field represents the IP label of the administrative (Ethernet) interface. We create the cluster using the command shown in Example 5-28.

Example 5-28 Create a GPFS cluster

```
lib01:~ # mmcrcluster -N /data/tmp/mmnodefile -p lib01sw1 -s lib02sw1
-r /usr/bin/ssh -R /usr/bin/scp -A
```

```
lib01:~ # mmlscluster
```

```
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      lib01sw1
GPFS cluster id:       13882355340112934961
GPFS UID domain:      lib01sw1
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
```

```
GPFS cluster configuration servers:
```

```
-----
```

```
Primary server:  lib01sw1
Secondary server: lib02sw1
```

Node	Daemon node name	IP address	Admin node name	Designation
1	lib01sw1	192.168.8.161	lib01	quorum
2	lib02sw1	192.168.8.162	lib02	quorum
3	lib03sw1	192.168.8.163	lib03	quorum
4	lib04sw1	192.168.8.164	lib04	

Creating Network Shared Disks (NSDs)

Select the disks (LUNs) you will use for GPFS file system. We create a disk descriptor file containing the disks to be used for file system and their designation, one per line, as shown in Example 5-29.

Example 5-29 DescFile for GPFS

```
lib01:~ # cat /data/tmp/nsdfile
/dev/sde:lib01sw1:lib02sw1:::
/dev/sdf:lib02sw1:lib01sw1:::
```

Note: In Example 5-29 on page 186, /dev/sde is the device name on lib01, and is accessible on lib02, while /dev/sdf is the device name on lib02, and is accessible on lib01.

Use the commands shown in Example 5-30 to create NSDs and to check the NSD configuration.

Example 5-30 Create NSD for GPFS cluster

```
lib01:~ # mmcnsd -F /data/tmp/nsdfile

lib01:~ # cat /data/tmp/nsdfile
# /dev/sde:lib01sw1:lib02sw1:::
gpfs1nsd:::dataAndMetadata:4001::
# /dev/sdf:lib02sw1:lib01sw1:::
gpfs2nsd:::dataAndMetadata:4001::

lib01:~ # mmlnsd
File system   Disk name   Primary node   Backup node
-----
 (free disk)  gpfs1nsd    lib01sw1      lib02sw1
 (free disk)  gpfs2nsd    lib02sw1      lib01sw1
```

Creating the GPFS file system

Attention: Any operation on a GPFS file system requires that the GPFS daemon (mmfs) is up and running.

Before creating a file system, you have to start the GPFS daemon on all nodes by running the following command:

```
lib01:~ # mmstartup -a
```

Check the status of GPFS cluster using the **mmgetstate** command, as shown in Example 5-31.

Example 5-31 GPFS cluster status

```
lib01:~ # mmgetstate -a

Node number  Node name      GPFS state
-----
      1      lib01sw1      active
      2      lib02sw1      active
```

The nodes' status shows "active", so you can create a file system using the NSDs defined in the previous step and the same disk definition file used for NSD creation (the file has been modified by the `mmcrnsd` command and adjusted for the `mmcrfs` command):

```
lib01:~ # mmcrfs /gpfs gpfs1v -F /data/tmp/nsdfile -A yes
```

Attention: The previous command used the default file system parameters. If you need to create a file system with different parameters, check the *GPFS V3.1 Administration and Programming Reference*, SA23-2221.

Mount the GPFS file system using the `mmmount gpfs1v -a` command. The GPFS file system will be mounted automatically at the next system (or GPFS daemon) restart.

Step 4. Installing compilers

Note: In this section, we assume that the /data directory on the management server has been exported through NFS and mounted on all compute nodes.

If you plan to develop applications, you need to install IBM C/C++ and Fortran compilers. IBM compilers are used for the IBM Parallel Environment (PE).

Installation of IBM XL C/C++

Copy the installation files from XL C/C++ media into the /data/HPC/Compiler/C directory on the management server. Log on to the nodes, run the `x1c_install` command from the /data/HPC/Compiler/C directory (NFS mounted from MS), and follow the directions.

Installation of IBM XL Fortran

Copy the installation files from the media into the /data/HPC/Compiler/Fortran directory on the management server. Log on to the nodes, run the `x1f_install` command from /data/HPC/Compiler/Fortran directory, and follow the directions.

Step 5. PE installation and testing

This section describes basic process of PE installation; for more information about the installation, visit the following URL:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.pe.doc/pebooks.html>

Installing the IBM PE using the supplied script

An installation script, `pe_install.sh`, is supplied to ease the install process. You can use this script to install only the PE license RPM, or you can install the PE license RPM along with the PE and LAPI product RPMs.

We have copied the script and all RPMs (PE, LAPI) into the `/data/HPC/PE` directory, and ran the following command on the management server to accept the license and install all the required LAPI and PE packages:

```
msib02:! # dsh -a "/data/HPC/PE/pe_install.sh -dir /data/HPC/PE"
```

Verify the POE installation

To test the installation of POE, use the POE installation verification program (IVP), which can be found in `/opt/ibmhpc/ppe.poe/samples/ivp`. We recommend you use a non-root account (`ibitso`) to run the script `ivp.linux.script`, as shown in Example 5-32.

Example 5-32 verify the POE installation by using IVP

```
ibitso@lib02:~> cd /opt/ibmhpc/ppe.poe/samples/ivp

ibitso@lib02:/opt/ibmhpc/ppe.poe/samples/ivp> ./ivp.linux.script
Verifying the existence of the Binaries
Partition Manager daemon /etc/pmdv4 is executable
POE files seem to be in order
Compiling the ivp sample program
Output files will be stored in directory /tmp/ivp14264
Creating host.list file for this node
Setting the required environment variables
Executing the parallel program with 2 tasks

POE IVP: running as task 0 on node lib02
POE IVP: there are 2 tasks running
POE IVP: running as task 1 on node lib02
POE IVP: all messages sent
POE IVP: task 1 received <POE IVP Message Passing Text>

Parallel program ivp.out return code was 0
```

If the test returns a return code of 0, POE IVP is successful. To test message passing, run the tests in `/opt/ibmhpc/ppe.poe/samples/poetest.bw` and `poetest.cast`. To test threaded message passing, run the tests in `/opt/ibmhpc/ppe.poe/samples/threads`.

Testing POE with IPoIB

POE provides another test program called **poetest.bw** that can be found in the `/opt/ibmhpc/ppe.poe/samples` directory. We use this program to test the IPoIB configuration in this step, and to test the User Space protocol in “Step 7. LoadLeveler installation and configuration” on page 192. We strongly recommend copying the entire contents of the `poetest.bw` directory to a shared file system, such as NFS or GPFS, so that the package can be accessed across all nodes.

Note: Before running any interactive POE jobs, make sure that Remote Shell (**rsh**) is configured for the respective user, without prompting for password. If you use LoadLeveler to submit jobs, it is not necessary to enable **rsh**.

To test POE on IPoIB, you need to edit a host file (`host.list` by default) in the directory where your executable program resides. In this file, specify the IP label or IP address of the IB adapters. See Example 5-33.

Example 5-33 poetest.bw test

```
ibitso@lib01:/data/tmp/poetest.bw> ls
bw bw.f bw.o bw.run host.list makefile.linux README.bw

ibitso@lib01:/data/tmp/poetest.bw> cat host.list
lib01sw1
lib02sw1

ibitso@lib01:/data/tmp/poetest.bw> ./bw.run
hello from task 0
hello from task 1
MEASURED BANDWIDTH = xxx.xxxxxx *10**6 Bytes/sec
```

Note: If you are interested in the performance data or performance tuning for IPoIB, please refer to the following white paper:

http://www-03.ibm.com/systems/p/software/whitepapers/hpc_linux.pdf

IBM LoadLeveler is needed to support the User Space (US) protocol. We discuss how to test POE with the US protocol in “Step 7. LoadLeveler installation and configuration” on page 192.

Step 6. Creating an RSCT Peer Domain for compute nodes

LoadLeveler will use the RSCT RMC API to support dynamic adapter configuration, if an machine stanza in the Admin file does not contain any adapters. You can also collect information about InfiniBand adapters by issuing the LoadLeveler command `llextRPD`, which is based on RSCT Peer Domain (RPD) functions.

This section describes the processes to set up an RPD for nodes with IB HCA installed. For more information, refer to RSCT documentation Web pages:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.rsct.doc/rsctbooks.html>

Before creating a Peer Domain for compute nodes, be sure that you have the packages shown in Example 5-34 installed on all nodes.

Example 5-34 RSCT packages

```
lib01:~ # rpm -qa | grep rsct
rsct.basic-2.4.6.0-06249
rsct.core.utils-2.4.6.0-06249
rsct.core.cimrm-2.4.6.0-06249
rsct.64bit-2.4.6.0-0
rsct.core-2.4.6.0-06249
```

We have created a file that lists the administrative IP labels of the compute nodes, one node per line.

Note: Even though you specify the nodes' IP label on the administrative network, RSCT will discover all adapters (including InfiniBand) in the node and will monitor them.

On one of the compute nodes (lib01 in our example), run the commands shown in Example 5-35 to create a PeerDomain, and check the status.

Example 5-35 Create peer domain for nodes with HCA installed

```
lib01:~ # cat /data/tmp/rpnodes
lib01
lib02
lib03
lib04

lib01:~ # preprnode -f /data/tmp/rpnodes

lib01:~ # mkrpdomain -f /data/tmp/rpnodes ib_peerdomain

lib01:~ # startrpdomain ib_peerdomain

lib01:~ # lsrpdomain
Name           OpState RSCTActiveVersion MixedVersions TSPort  GSPort
ib_peerdomain Online  2.4.6.0           No           12347  12348

lib01:~ # lsrpnode
Name OpState RSCTVersion
lib04 Online  2.4.6.0
lib03 Online  2.4.6.0
lib02 Online  2.4.6.0
lib01 Online  2.4.6.0
```

Step 7. LoadLeveler installation and configuration

Before installing IBM TWS LoadLeveler, you should have a shared file system across all nodes. For example, you can NFS export the /home directory on management server, and mount it on all compute nodes.

For the test system described in 5.3.2, “Sample SLES 9 Cluster layout and description” on page 169, we use the management server also as the LoadLeveler Central Manager (CM), hence the need to install LoadLeveler on all nodes in the cluster, including the CSM MS.

LoadLeveler installation

Assuming you have the LL code on the MS already, and the /data file system NFS exported and mounted on all compute nodes, run the following commands (in sequence) to install packages and accept the license:

```
msib02:~ # dsh -a "rpm -ivh \  
/data/HPC/LL/LoadL-full-license-SLES9-PPC64-3.3.2.5-0.ppc64.rpm"  
msib02:~ # dsh -a "/opt/ibmll/LoadL/sbin/install_ll -y -d \  
/data/HPC/LL"  
msib02:~ # dsh -a "rpm -ivh \  
/data/HPC/LL/LoadL-full-lib-SLES9-PPC-3.3.2.5-0.ppc.rpm"
```

LoadLeveler configuration

Create the loadl group and load the user on the management server:

```
msib02:~ # groupadd -g 201 loadl  
msib02:~ # useradd -d /home/loadl -u 201 -g 201 -m loadl
```

Note: Be sure that group ID 201 and user ID 201 are not used for another group and user. If so, replace 201 with another available number.

Copy the passwd, shadow, and group files in the /etc directory on the management server to all the compute nodes manually. Alternately, if the CFM function has been enabled during the CSM configuration, you can use the **cfmupdatenode** command.

Now, run the following commands on the management server to initialize LoadLeveler:

```
msib02:~ # mkdir /var/loadl  
msib02:~ # chown -R loadl:loadl /var/loadl  
msib02:~ # dsh -a "mkdir /var/loadl"  
dsh -a "chown -R loadl:loadl /var/loadl"  
msib02:~ # su - loadl  
loadl@msib02:~ > cd /opt/ibmll/LoadL/full/bin; ./llinit -local  
/var/loadl -release /opt/ibmll/LoadL/full -cm msib02
```

On all compute nodes, run the following commands:

```
su - loadl  
cd /opt/ibmll/LoadL/full/bin  
./llinit -local /var/loadl -release /opt/ibmll/LoadL/full -cm msib02
```

Note: Check if the /home/loadl/bin directory on the management server is a symbolic link to /opt/ibmll/LoadL/full/bin. If not, create the symlink manually.

Customizing LoadLeveler

Customizing LoadLeveler consists of editing the LoadL_config and LoadL_admin global configuration files, and the local file, LoadL_config.local, to match the needs of your installation. For detailed instructions, refer to the LoadLeveler product manuals.

To enable LL to use the InfiniBand adapters, you can either explicitly store the adapters' description into Adapter Stanza in LoadL_admin file, or you can use the dynamic adapter configuration feature provided by RMC (this assumes you have already configured your LL nodes in an RPD).

By using the command **llextrPD**, you can extract the adapter information data from an RSCT peer domain to set up the administration file, as shown in Example 5-36.

Example 5-36 Output of command llextrPD

```
loadl@lib01:~> llextrPD

#llextrPD: Cluster = "ib_peerdomain" ID = "2VGgMFn9eYpf0SrGovVHwm" on
Tue Oct 17 13:07:11 2006

lib02: type = machine
      adapter_stanzas = lib02.itso.ibm.com lib02sw2 lib02sw1
      alias = lib02.itso.ibm.com lib02sw2 lib02sw1

lib02.itso.ibm.com: type = adapter
      adapter_name = eth3
      network_type = ethernet
      interface_address = 192.168.100.162
      interface_netmask = 255.255.255.0
      interface_name = lib02.itso.ibm.com
      logical_id = 0

lib02sw2: type = adapter
      adapter_name = ib1
      network_type = InfiniBand
      interface_address = 192.168.9.162
      interface_netmask = 255.255.255.0
      interface_name = lib02sw2
      logical_id = 20
      adapter_type = InfiniBand
      device_driver_name = ehca0
      network_id = 18338657682652659712
      port_number = 2
```

```

lib02sw1: type = adapter
    adapter_name = ib0
    network_type = InfiniBand
    interface_address = 192.168.8.162
    interface_netmask = 255.255.255.0
    interface_name = lib02sw1
    logical_id = 8
    adapter_type = InfiniBand
    device_driver_name = ehca0
    network_id = 18338657682652659714
    port_number = 1

lib01: type = machine
    adapter_stanzas = lib01.itso.ibm.com lib01sw2 lib01sw1
    alias = lib01.itso.ibm.com lib01sw2 lib01sw1

lib01.itso.ibm.com: type = adapter
    adapter_name = eth3
    network_type = ethernet
    interface_address = 192.168.100.161
    interface_netmask = 255.255.255.0
    interface_name = lib01.itso.ibm.com
    logical_id = 0

lib01sw2: type = adapter
    adapter_name = ib1
    network_type = InfiniBand
    interface_address = 192.168.9.161
    interface_netmask = 255.255.255.0
    interface_name = lib01sw2
    logical_id = 22
    adapter_type = InfiniBand
    device_driver_name = ehca0
    network_id = 18338657682652659712
    port_number = 2

lib01sw1: type = adapter
    adapter_name = ib0
    network_type = InfiniBand
    interface_address = 192.168.8.161
    interface_netmask = 255.255.255.0
    interface_name = lib01sw1
    logical_id = 7
    adapter_type = InfiniBand
    device_driver_name = ehca0

```

```
network_id = 18338657682652659714
port_number = 1
```

... ..

You will find information about InfiniBand adapters on all nodes in the peer domain. Store this information into the LoadL_admin file, and modify the items according to your requirements.

After the customization of LoadLeveler has been completed, use the **llctl** command to start LoadLeveler, and use the **llstatus** command to check the status, as shown in Example 5-37.

Example 5-37 Startup LL and check status

```
loadl@msib02:~> llctl -g start
```

```
loadl@msib02:~> llstatus -a
```

```
=====
msib02.itso.ibm.com
```

```
=====
lib02.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,64/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib02sw2,192.168.9.162,,2,64/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib02sw1,192.168.8.162,,2,64/64,0/0 rCxt Blks,1,READY,1)
```

```
=====
lib01.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,64/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib01sw2,192.168.9.161,,1,64/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib01sw1,192.168.8.161,,1,64/64,0/0 rCxt Blks,1,READY,1)
```

Refer to 5.2.3, “Software packages for High Performance Computing” on page 158 for the explanation of the output shown in Example 5-37.

Testing LoadLeveler with IBM PE

You may use the sample package poetest.bw to test the LoadLeveler supporting user space protocol with IBM PE.

We store this package in the /data/HPC/TEST/poetest.bw directory. The files required for this test are shown in Example 5-38.

Example 5-38 Sample poetest.bw

```
ibitso@msib02:/data/HPC/TEST/poetest.bw> ls -ltr
total 16
-rw-r--r-- 1 ibitso users 538 2006-10-11 20:45 makefile.linux
-rw-r--r-- 1 ibitso users 3068 2006-10-17 12:55 bw.f
-rw-r--r-- 1 ibitso users 12 2006-10-17 13:09 host.list
-rwxr-xr-x 1 ibitso users 368 2006-10-17 13:09 lltest.cmd
```

The files makefile.linux and bw.f are left as they are in the original poetest.bw package (unchanged). The other two files, host.list and lltest.cmd, should be created according to your local configuration. Example 5-39 and Example 5-40 show the contents of these files in our sample cluster:

Example 5-39 File host.list

```
ibitso@msib02:/data/HPC/TEST/poetest.bw> cat host.list
lib01
lib02
```

In the host.list file, it is not necessary to specify the IB adapters IP labels, such as lib01sw1 and lib02sw1. Since we test User Space protocol, IP information is irrelevant here.

Example 5-40 File lltest.cmd

```
ibitso@msib02:/data/HPC/TEST/poetest.bw> cat lltest.cmd
#!/bin/sh
# @ error = job1.%(Host).%(Cluster).%(Process).err
# @ output = job1.%(Host).%(Cluster).%(Process).out
# @ class = small
# @ job_type = parallel
# @ node = 2
# @ tasks_per_node = 1
# @ network.mpi = sn_all,shared,us,,instances=1
# @ queue

poe ./bw -procs 2
```

Note: To make use of User Space protocol, make sure that you have the keyword network in the LoadLeveler command file.

Now run the following commands to generate an executable file and submit a LoadLeveler job:

```
loadl@msib02:~ > make -f makefile.linux; llsubmit ./lltest.cmd
```

After the job has been submitted, you may trace the job status using the commands shown in Example 5-41 and Example 5-42.

Example 5-41 llstatus

```
ibitso@lib01:/data/HPC/TEST/poetest.bw> llstatus
```

Name	Schedd	InQ	Act	Startd	Run	LdAvg	Idle	Arch	OpSys
lib01.itso.ibm.com	Avail	0	0	Run	1	0.01	0	PPC64	Linux2
lib02.itso.ibm.com	Avail	0	0	Run	1	0.06	9999	PPC64	Linux2
msib02.itso.ibm.com	Avail	0	0	None	0	0.92	0	PPC64	Linux2

Observe that two processes of the job have been distributed over to lib01 and lib02.

Example 5-42 llstatus -a

```
ibitso@lib01:/data/HPC/TEST/poetest.bw> llstatus -a
```

```
=====
msib02.itso.ibm.com
=====
lib02.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,63/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,63/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib02sw2,192.168.9.162,,2,63/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,63/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib02sw1,192.168.8.162,,2,63/64,0/0 rCxt Blks,1,READY,1)
=====
lib01.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,63/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,63/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib01sw2,192.168.9.161,,1,63/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,63/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib01sw1,192.168.8.161,,1,63/64,0/0 rCxt Blks,1,READY,1)
=====
```

You can see that one `us_window` has been used on each node. Now, the `available_windows/total_windows` is 63/64 on each node.

After the job has finished, check the output in the file `job1.lib01.13.0.out`, as shown in Example 5-43 on page 199.

Example 5-43 Output of poetest.bw

```
ibitso@lib01:/data/HPC/TEST/poetest.bw> cat job1.lib01.13.0.out
hello from task 0
hello from task 1
MEASURED BANDWIDTH = xxx.xxxxxx *10**6 Bytes/sec
```



Part 3

Support

This section presents InfiniBand system administration and problem determination tools and techniques. We discuss issues that we have seen in testing, and describe how we have resolved them, along with some best practices for working with pSeries InfiniBand clusters and an introduction to InfiniBand monitoring.



Problem determination

This chapter discusses issues that we have seen in testing, and describes how we have resolved them. This chapter also contains useful links that will help to understand the cause of the issues and resolve them. This chapter is not intended to be an exhaustive method for fixing all IB-related problems, but is rather a guide to help solve common problems, and includes links to more in-depth articles that will aid operators and administrators resolve problems in their environments.

Common System p and InfiniBand problem descriptions and solutions can be found at:

<http://www14.software.ibm.com/webapp/set2/sass/f/networkmanager/problems.html>

One general comment that we hear from the various support teams at IBM was that if a problem is obvious, do not upload all the log files. If the problem seems obscure, it is very beneficial to include all of the log files and messages files from the system when a PMR is opened with IBM. This allows the case to be moved directly to the higher levels of support, and results in a much faster resolution of the problem

6.1 IB switch troubleshooting

At the time of this writing, the switch available for testing was manufactured by Cisco. If you suspect there is an issue with your Cisco InfiniBand switch, here are some simple steps you can take to determine the nature of the problem.

The first step is to open IBM Network Manager and review the port states, and to observe the system logs for any kind of error.

There is an error log that can be useful in diagnosing problems, and is often requested by the phone support staff to aid in resolving issues.

6.1.1 Physical layer issues

Generally speaking, in an IB environment, the most common physical layer problems are defective cables and connectors.

Cable swap problems

This problem usually happens when first setting up the InfiniBand network, or just after maintenance has been done on the server. If the two InfiniBand cables are swapped on the same adapter, communication will be broken.

Resolution:

Label your cables. Careful planning and visible labels will help avoid these issues. We have used red tape for all port0 cables, and blue tape for all port1 cables.

If you did not label your cables and still need to figure out which connection is wrong, simply go in to IBM Network Manager and look at all the port0 and port1 connections. If the cables are swapped between the ports, the port that is set wrong will have a different GID number.

If you observe intermittent errors in communication on you IB network, and the ports and HCAs pass diagnostics, you should consider replacing the cables with new ones or cables you are sure that are reliable.

6.1.2 InfiniBand switch firmware upgrade process

Please note that on the 24 port switch, there is no redundant system controller, and the switch must be fully rebooted as part of the upgrade process.

To upgrade the firmware of the switch, you will need the following items:

- ▶ A computer that can access the InfiniBand switch's Ethernet management port
- ▶ The appropriate firmware upgrade files from the switch manufacturer, located on an FTP or TFTP server
- ▶ The administrator account on the switch

The steps to upgrade the switch firmware are:

1. Log in to the Web Interface on the switch. Make sure your account has administrator level privileges.
2. Click the **Maintenance** menu and select **File Management**. The File Management window opens.
3. Click the line in the **Current Files on System** table that represents the file that you want to install, then click the **Install** button. A verification window opens.

Note: Before you install an image make sure that all of the cards on the chassis have been brought up.

4. Once you click **Yes** in the next step, the switch will reboot. Make sure your environment is ready for the switch to go down.
5. Click the **Yes** button to install the image.

The system installs the new image, and reboots the switch for the changes to take effect.

6.2 System p troubleshooting

For more detailed information about how to troubleshoot hardware and firmware issues on System p Servers, go to:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iph5/troubleshooting_firm.htm

6.2.1 HCA troubleshooting

For more detailed information about how to resolve issues with your IBM InfiniBand GX bus adapter, check the following Web page:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphau/infinibandpdf.pdf>

A common problem that can occur in a shared adapter configuration is that it is possible to run out of resources. Careful planning must be used when implementing shared adapters. It is very important to know the nature of the jobs that you will be running on a shared adapter. If any application is being used that can generate tasks for the HCA, such as a HPC type job, then you should either not share the adapter or know exactly how many tasks the job you are creating will consume.

In an HPC environment, applications like Parallel Environment will generate numerous tasks, which in turn consumes queue pairs. There are few indicators if you are having this problem. The only error messages that you will get are going to be from the application trying to create the next queue pair. This application will deliver a message that says something like `open queue pair failed`. If you see this message on a machine that is using a shared adapter, and you are seeing degraded performance on an application, then the system administrator should review the HCA priority settings and raise the priority level of the HCA allocated to this partition.

6.2.2 Logs available for troubleshooting

When working on InfiniBand related problems, several files can be gathered for further problem determination. These logs are:

- ▶ `/var/log/messages` from a Linux partition.
- ▶ `snap.pax.Z` data from an AIX partition (will be generated with the `snap -gc` command).
- ▶ `ibnm snap` generated on HMC (refer to “HMC/IBM Network Manager troubleshooting” on page 206 for details).
- ▶ `iqyylog` from HMC can be gathered like this: On the HMC GUI, select **Service Applications** → **Service Focal Point** → **Manage Serviceable Events** → **ALL** and then press **OK**. Select one SFP log entry, click **Selected**, click **Manage Problem Data**, select `iqyylog.log`, and then either select **Call Home** or **Save to DVD**.

6.2.3 HMC/IBM Network Manager troubleshooting

An extensive guide to troubleshooting using IBM Network Manager is available at:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/tpspnelemrug.pdf>

The process for gathering error logs on IBM Network Manager can be found in “Gathering the IBNM snap data” on page 68.

You only need to be logged in as hscroot and open a shell window to issue this command. After the files are generated, they can be FTP'd to IBM for analysis. Be sure to note in your PMR that you have uploaded the file to support.

6.3 AIX troubleshooting

An extensive guide to AIX troubleshooting can be found at:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp?topic=/iphau/referenceinfiniband.htm>

Some of the common problems we have observed are shown in the following sections.

Cable disconnected

Log in to the Flexible Service Processor's (FSP) Advanced System Management Interface (ASMI) from the HMC. In the GUI, select **Service Applications** → **Service Focal Point** → **Service Utilities**, select **CEC**, and select **Selected** → **Launch ASM Menu**.

From the ASMI, you can look into the Error/Event log and search for a B70069Ex entry. These entries are "Informational" but will point to a possible InfiniBand problem. When an InfiniBand cable is disconnected, a B70069E6 entry will appear. When the cable has been reconnected, a B70069E9 code will appear in the FSP error/event log. The meaning of these entries is shown in Table 6-1.

Table 6-1 B70069Ex codes

Reference code	Description
B700 69E6	The cable connection to the InfiniBand switch is broken. Verify that the cable is connected and the switch is powered on. After repairing the problem, verify that the connection has been restored by using the InfiniBand Network Manager.
B700 69E9	Initial communication has occurred with the Subnet Manager. No service action required. The CEC and switch firmware are now communicating.

Interface does not configure and goes to the stopped/defined state

- ▶ Check if the HCA driver is available using the `lsdev -Cc adapter` command.
If the HCA Adapter it is not available, and the adapter is a Galaxy adapter, check the HMC partition properties (not the profile) in the hardware tab. Check if the HCA tab is there. If it is not there, the partition does not recognize the Galaxy HCA adapter.
- ▶ Check the Galaxy adapters to see if the HMC LPAR's properties have a GUID index assigned to the adapter.
- ▶ Check if the ICM is configured by using `lsdev -C | grep icm`.
- ▶ If ICM is not configured, run `smitty icm`.
- ▶ Check if the IB interface parameters are within the range.
A well known P_KEY is 0xFFFF or 0x7FFF.
- ▶ Check the port range supported by the adapter (use F4 when possible to see the defaults).
- ▶ Check that the MTU is within the range (32 to 2044).
- ▶ Check if AIX is running in 64-bit kernel mode and if it is V5.3 TL5 or later.

Ping does not work and ARP table shows incomplete entries

An ARP request packet was sent to the network, and the interface is waiting for the ARP reply.

Probable causes:

- ▶ The remote system or interface is not UP and RUNNING.
Run `ifconfig ib0` remotely to see if it has the UP and RUNNING flags set.
- ▶ The local or remote interfaces are not in the broadcast multicast group.
The switch may have kicked out the interface from the multicast group if there are communication problems. Access to the switch will show the multicast groups and the port (GID) belonging to it.

Ping does not work but the ARP entry is complete

Probable causes:

- ▶ The remote system or adapter is not UP and RUNNING.
Run `ifconfig ib0` remotely to see if it has the UP and RUNNING flags set.
- ▶ The local or remote interfaces are not in the broadcast multicast group.
The switch may have kicked out the interface from the multicast group if there are communication problems. Access to the switch will show the multicast groups and the port (GID) belonging to it.

HCA adapter problems

- ▶ Ping does not work and a message showing that the network is not currently available is in the shell.
- ▶ The local Interface is not UP and RUNNING.
- ▶ The ports may be disconnected from the InfiniBand Switch.
- ▶ The Interface may have failed to join the multicast group and is retrying.
This can happen if the switch is not responding, or the HCA did not send the join packet.
- ▶ The interface deliberately was put in the down state.
Try to bring up the IB adapter using `ifconfig ib0 up`.

6.4 Troubleshooting IB on SLES 9

SLES Linux has several tools that can be used to determine the functionality of different components within this system. In this section, we present some methods for identifying and solving IB related issues.

6.4.1 The dmesg tool

The `dmesg` tool is used to examine the kernel ring (circular) buffer containing traces generated by kernel modules. eHCA always writes out error messages into kernel ring buffer with certain information useful for problem determination. Besides error messages, eHCA also produces informative messages, for example, at module load time.

A sample **dmesg** output is shown in Example 6-1.

Example 6-1 Sample dmesg output

```
lib02:~ # dmesg
ibmebus: no version for "struct_module" found: kernel tainted.
ibmebus: module not supported by Novell, setting U taint flag.
eBus Device Driver
ib_core: module not supported by Novell, setting U taint flag.
sym0:15:0:phase change 6-7 6@004c0390 resid=4.
hcad_mod: module not supported by Novell, setting U taint flag.
eHCA Infiniband Device Driver (Rel.: EHCA2_0058)
xics_enable_irq: irq=900b: ibm_int_on returned ffffffff
ib_mad: module not supported by Novell, setting U taint flag.
PID0000 00060139:parse_ec ehca0: port 1 is active.
PID0000 00060139:parse_ec ehca0: port 2 is active.
ib_verbs: module not supported by Novell, setting U taint flag.
ib_sa: module not supported by Novell, setting U taint flag.
ib_ipoib: module not supported by Novell, setting U taint flag.
```

Note: Other kernel modules, for example, `ib_ipoib` and `ib_mad`, also use this mechanism to generate their respective error and debug messages (it is a logging standard for Linux kernel modules).

Depending on the amount of messages produced, the **dmesg** output might be quite long. Cleaning the ring buffer after displaying a set of messages by using option `-c` helps to reduce the **dmesg** content to clean up and makes it easier to find what you are looking for:

```
lib02:~ # dmesg -c
# this will print the current content of the ring buffer and clears it
```

Option `-n <level>` can also be used to filter certain messages to be printed to the console. For example, **dmesg -n 1** will only show panic messages. In order to save a copy of the ring buffer (for example, for problem analysis), issue the following command:

```
lib02:~ # dmesg -c > /tmp/problem.log
# this prints all messages into given file and clears ring buffer
```

You can also use the **grep <pattern>** command to filter the messages that match the specified pattern. The command shown in Example 6-2 on page 211 filters all “ERROR” messages.

Example 6-2 dmesg with grep

```
lib02:~ # dmesg | grep ERROR
[1269626.095888] PU0001 0006006b:print_error_data HCAD_ERROR QP 0x71
(resource=2006000000000071) has errors.
```

The file `/var/log/messages` is the default output location of `syslogd`. Most messages that can be retrieved with `dmesg` are also written into `/var/log/messages`. However, some of these messages might not capture the whole ring buffer. Refer to the `syslog` daemon configuration file in `/etc/syslog.conf` for details about how and where `syslogd` writes out system messages.

To display the contents of `/var/log/messages`, use the following command:

```
lib02:~ # tail -f /var/log/messages
# this will print first the last 10 messages and output appended
messages
```

Use the following command to clear the content of `/var/log/messages`:

```
lib02:~ # echo "" > /var/log/messages
```

Note: The kernel ring buffer is emptied when rebooting the partition, while `/var/log/messages` retains its content. However, in case of errors, especially kernel panic, the last ring buffer content is often lost.

6.4.2 eHCA device driver version

To determine the eHCA device driver version, issue the following command:

```
lib02:~ # modinfo hcad_mod | grep version
version:          EHCA2_0058 A693B35D29EC5C569166E1F
```

If `modinfo` complains that it is not able to locate the version, there is another way to find `hca_mod` by issuing the following command:

```
lib02:~ # modinfo hca_mod | grep version
modinfo: could not find module hca_mod
```

If these both fail, issue the `depmod` command, which updates the module system map file (module dependencies), and to load the module again (use `modprobe` or the eHCA init scripts in `/etc/init.d`; see 5.1.3, “InfiniBand implementation on SLES 9” on page 145). If this still shows the same error message, check if `hcad_mod.ko` is installed properly. Refer to 5.1.2, “Software components and versions” on page 143.

Device driver options

In order to obtain a list of available options, you can use `modinfo` as shown in Example 6-3.

Example 6-3 eHCA device driver options

```
lib02:~ # modinfo hcad_mod
parm:      static_rate:set permanent static rate (default:
disabled) (int)
parm:      poll_all_eqs:polls all event queues periodically0 no,1
yes (default) (int)
parm:      port_act_time:time to wait for port activation(default:
30 sec.) (int)
parm:      use_hp_mr:use high performance MRs,0 no (default),1 yes
(int)
parm:      nr_ports:number of connected ports (default: 2) (int)
parm:      hw_level:0 autosensing,1 v. 0.20,2 v. 0.21 (int)
parm:      tracelevel:0 maximum performance (no messages),9
maximum messages (no performance) (int)
parm:      open_aqp1:0 no define AQP1 on startup (default),1
define AQP1 on startup (int)
version:   EHCA2_0058 A693B35D29EC5C569166E1F
description:  IBM eServer HCA Driver
author:     Christoph Raisch <raisch@de.ibm.com>
license:    Dual BSD/GPL
depends:
vermagic:   2.6.5-7.244-pseries64 SMP gcc-3.3
```

- ▶ The option `static_rate` specifies the Internet Packet Delay value. The default is -1, which causes eHCA to default to the rate provided by the device. A value of zero means no delay regardless of system utilization, and up to 100% of the resources are available. A value of one means 50% utilization is available.
- ▶ The option `poll_all_eqs` accepts a value of zero or one. The default is one, which causes eHCA to poll event queues periodically in addition to notified events. This is a mechanism that prevents completions from being lost due to a very rare race condition in the system. This option is provided for debugging purposes only.
- ▶ The option `port_act_time` specifies the timeout value for the port being activated. The default is 30 seconds. Depending on the complexity of your InfiniBand network, you might need to increase this value.
- ▶ The option `use_hp_mr` (Use High Performance Memory Regions) accepts a value of zero or one. The default is one. This option is provided for debugging purposes only.

- ▶ The option `nr_ports` specifies the Number of Ports connected with the switch. The default is 2. If you have only one port connected, you need to instruct eHCA to initialize and activate respectively one port only. See also `ib_mad: could not create aqp1`.
- ▶ The option `hw_level`. The default is zero, which means eHCA determines the so-called hardware level of available HCAs automatically. This option is provided for debugging purposes only.
- ▶ The option `tracelevel` allows activation\deactivation of debug traces of eHCA. This is a list of trace levels for eHCA components separated by a comma. A trace level can accept a value in the range 0 - 9. The default trace level is 5, which causes only error messages to be written in the kernel ring buffer. A trace level of 6 or 7 will include warning and informative messages, while trace level of 9 will also produce all debug messages.

In order to enable debug messages for all (19) components, issue the following command:

```
lib02:~ # modprobe hcad_mod \
tracelevel=9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
```

The following information is relevant if you have turned debugging to on (see “Turn on debug output at runtime” on page 217).

Each component of the driver has an index starting with one. The most interesting components are:

- ▶ Memory regions, index 10
When the above debug code is set, all registration and de-registration of memory regions will be traced out.
- ▶ Queue pairs, index 12
When this debug code is set, all create, modify, and destroy queue pair calls will be traced out.
- ▶ Queue communication, 19
When this debug code is set, all `post_send`, `post_rcv` and `poll_cq` calls will be traced out.
- ▶ The option `open_aqp1` instructs eHCA to create AQP1 at load time or not. The default is zero. That means eHCA does not create AQP1 at load time, since `ib_mad` creates AQP1 by itself. Thus, do not set this option to one except for debugging purposes.

Example 6-4 presents a list of available options for the `ib_ehca` module (OFED-1.1.1).

Example 6-4 ib_ehca module options

```
lib02:~ # modinfo ib_ehca
filename:
/lib/modules/2.6.17/kernel/drivers/infiniband/hw/ehca/ib_ehca.ko
license:      Dual BSD/GPL
author:       Christoph Raisch <raisch@de.ibm.com>
description:  IBM eServer HCA InfiniBand Device Driver
version:      SVNEHCA_0015
vermagic:     2.6.17 SMP mod_unload gcc-4.1
depends:       ib_core
srcversion:   C7CC7F54B52BC40636EA880
parm:         static_rate:set permanent static rate (default:
disabled) (int)
parm:         poll_all_eqs:polls all event queues periodically (0:
no, 1: yes (default)) (int)
parm:         port_act_time:time to wait for port activation
(default: 30 sec) (int)
parm:         use_hp_mr:high performance MRs (0: no (default), 1:
yes) (int)
parm:         nr_ports:number of connected ports (default: 2) (int)
parm:         hw_level:hardware level (0: autosensing (default), 1:
v. 0.20, 2: v. 0.21) (int)
parm:         debug_level:debug level (0: no debug traces (default),
1: with debug traces) (int)
parm:         open_aqp1:AQP1 on startup (0: no (default), 1: yes)
(int)
```

The only difference from `hcad_mod` is that option `tracelevel` has been renamed to `debug_level`, which accepts one value of zero or one. The default is zero, which turns off all debug traces. Set this to one in order to get all debug traces from `ib_ehca`.

Example 6-5 on page 215 presents a `dmesg` output that shows an error message, prefix `EHCA_ERR`, that port 1 is not active.

Example 6-5 Error message: port 1 not connected

```
[3734362.827385] ib_core: no version for "struct_module" found: kernel tainted.
[3734376.991294] eHCA Infiniband Device Driver (Rel.: $ReleaseTag$)
[3734376.994569] xics_enable_irq: irq=36868: ibm_int_on returned -3
[3734425.496671] ehca B.001.DNW2B12-P1-C6: PU0001 EHCA_ERR:ehca_define_sqp Port 1 is
not active.
[3734425.496684] ehca B.001.DNW2B12-P1-C6: PU0001 EHCA_ERR:ehca_create_qp
ehca_define_sqp() failed rc=ffffffffffffffff
[3734425.496868] ib_mad: Couldn't create ib_mad QP1
[3734425.497067] ib_mad: Couldn't open ehca0 port 1
```

The error message in Example 6-5 is produced when loading module `ib_mad`, which creates AQP1 (queue pair of type General Service Interface/GSI). When such a queue pair is created, eHCA sends a request to the firmware in order to activate the associated port, which is required before any communication can be established, and waits until the port becomes active.

If the port is still not active after a timeout, eHCA returns an error code to `ib_mad` and produces the following error message:

```
Port is not active respective to ib_mad: could not create ib_mad QP1
```

The most common reason for this is that the port is not connected with the switch properly. Confirm that the adapter is seated firmly in the socket and the cable is plugged in correctly at both ends, that is, at adapter and switch.

Example 6-6 shows the same problem with port 2.

Example 6-6 Error output: port 2 not connected

```
[3735142.654156] ib_core: no version for "struct_module" found: kernel tainted.
[3735151.287274] eHCA Infiniband Device Driver (Rel.: EHCA2_0058)
[3735151.290483] xics_enable_irq: irq=36868: ibm_int_on returned -3
[3735168.427157] ehca B.001.DNW2B12-P1-C6: PU0001 EHCA_INFO:parse_ec port 1 is
active.
[3735199.483746] ehca B.001.DNW2B12-P1-C6: PU0000 EHCA_ERR:ehca_define_sqp Port 2 is
not active.
[3735199.483758] ehca B.001.DNW2B12-P1-C6: PU0000 EHCA_ERR:ehca_create_qp
ehca_define_sqp() failed rc=ffffffffffffffff
[3735199.483916] ib_mad: Couldn't create ib_mad QP1
[3735199.484089] ib_mad: Couldn't open ehca0 port 2
[3735199.484316] ehca B.001.DNW2B12-P1-C6: PU0000 EHCA_INFO:ehca_destroy_qp device
ehca0: port 1 is inactive.
```

In this case port 1 has been active, but not port 2. Thus, check the physical connection between port 2 and the switch. If port 2 should not be used for any reason, use option `nr_ports=1` to instruct eHCA to activate only port 1:

```
lib02:~ # modprobe hcad_mod nr_ports=1
```

If both ports are connected properly to the switch, the `dmesg` output should look similar to Example 6-7.

Example 6-7 dmesg output showing both ports connected

```
eHCA Infiniband Device Driver (Rel.: EHCA2_0058)
ib_mad: module not supported by Novell, setting U taint flag.
PID0000 00060139:parse_ec ehca0: port 1 is active.
PID0000 00060139:parse_ec ehca0: port 2 is active.
```

Monitoring resources

eHCA exposes its capability for allocating resources, that is, memory regions, queue pairs, and so on, and the current number of allocated resources in the system device tree integrated in Linux file system. The command `ls -l /sys/bus/ibmebus` shows the following output:

► `devices/`

This sub-directory provides information about all HCA adapters. Each HCA is represented as a sub-directory with a unique name with the following content:

```
adapter_handle: adapter handle assigned by firmware
cur_cq: current number of allocated completion queues
cur_eq: current number of allocated event queues
cur_mr: current number of allocated memory regions
cur_mw: current number of allocated memory windows
cur_qp: current number of allocated queue pairs
hw_ver: hardware version
max_ah: maximum number of available address handles
max_cq: maximum number of available completion queues
max_eq: maximum number of available event queues
max_mr: maximum number of available memory regions
max_mw: maximum number of available memory windows
max_pd: maximum number of available protection domains
max_qp: maximum number of available queue pairs
name: firmware device name, e.g. "lhca"
num_ports: current number of ports connect to the switch
```

► `drivers/ehca`

This sub-directory provides information about eHCA device drivers corresponding to available HCA adapters.


```

[3565304.356914] EHCA_DMP:print_error_data resource=200000000000004e
adr=c000000101bc5020 ofs=0020 a000000500000000 0000000000000000
[3565304.356932] EHCA_DMP:print_error_data resource=200000000000004e
adr=c000000101bc5030 ofs=0030 000000001000000 0000000000000000
...
[3565304.358408] ehca B.001.DNW288B-P1-C6: PU0003 EHCA_ERR:print_error_data EHCA
----- error data end -----

```

Due to the complexity of the error data, in the following paragraphs we present the errors are for queue pairs only.

The first error message gives information about the resource type for which the error event primarily was sent, in this case, a queue pair with qp_num=0x4E. The second line indicates that there is an error data block available. This is followed by an error data block enclosed by "----- error data begin -----" and "----- error data end -----" at the start and end respectively.

Each line within an error data block shows two 64-bit data words (hexadecimal format) consecutively from start offset zero. That means the next line has an offset of 0x10 and so forth.

Let us examine the header (16 bytes) of an error data block, that is, the first line. For better readability, only offset and data words are shown:

```
ofs=0000 0000000000000538 200000000000004e
```

The first data word 0x0000000000000538 contains the following information:

- ▶ Error type, bit 40-47, that is, 0x01

This specifies the resource type, which this error data block was generated for. For queue pairs, this value must be one.
- ▶ Length in byte, bit 52-63, that is, 0x138

This specifies the total length of this error data block in bytes. In this example, it is 312 bytes.

After the header (16 bytes) is a sequence of error entries. Each error entry contains a sub-header followed by actual error data of indicated resource type. That means for the first error entry the sub-header is the data word at offset 0x10:

```
ofs=0010 0100000000000310 8000000000000000
```

This sub-header 0x0100000000000310 contains the following fields:

- ▶ Error type, bit 0-7, that is, 0x01

This specifies the resource type. For queue pairs this value must be one.

- ▶ Length in bytes, bit 56-63, that is, 0x10

This specifies the total length of this error entry in bytes, that is, 16 bytes.

The error data of resource type queue pair contains much HCA specific information. The most interesting content is described as follows:

- ▶ Queue pair control word

This indicates queue pair state and is at offset at 0x08. For the first error entry the offset within an error data block is then $0x08+0x18=0x20$:

```
ofs=0020 a000000500000000 0000000000000000
```

QP debug information is explained in Table 6-2.

Table 6-2 Debug code information

Field	Bit	Meaning
QP Enabled	0	0=disabled, 1=enabled
QP disable state	1	Valid only if QP is not enabled 0=QP disable is in process, 1=QP disable complete
QP Error	2	0=QP not in error state, 1= QP in error state
Requested QP State	16-23	1=Reset, 2=Initialized, 3=ready to receive, 5=ready to send, 6=send queue draining, 8=send queue error, 128= error
Resultant QP State	25-31	See requested QP state. If zero, this is equal to requested QP State

In our previous example we can see that the queue pair is enabled, not in error state, and its current state is ready to send.

- ▶ Queue pair error word

This gives more details about the error type and is at offset at 0x18. For the first entry, the offset within an error data block is then $0x18+0x18=0x30$:

```
ofs=0030 0000000001000000 0000000000000000
```

Each bit of this error word, if set to one, has a certain meaning. The most important error indicators are listed in Table 6-3.

Table 6-3 Debug code bit definitions

Bit	Definition
38	Send Queue overflow error
39	Send completion error
40	Double link send\recieve queue error
41	Receive queue overflow error
42	Receive queue error
43	Malformed WQE error
45	Remote access protection error
46	Completion queue overflow error
47	Completion queue processing error
49	Memory region table processing error
54	Data access memory error

Field bit meaning

In Example 6-9, we can see that the queue pair is enabled, not in an error state, and its current state is ready to send.

Example 6-9 Queue pair state

QP enabled	0	0 = disabled, 1 = enabled
QP disable state	1	Valid only if QP is not enable 0 = QP disable is in process, 1 = QP disable complete
QP error	2	0 = QP not in error state, 1 = QP in error state
Requested QP state 16-23		1 = Reset, 2 = Initialized, 3 = Ready to receive, 5 = Ready to send, 6 = Send queue draining, 8 = Send queue error, 128 = Error
Resultant QP state 25-31		See requested QP state. If zero, this is equal requested QP state.

6.4.3 Troubleshooting IP over InfiniBand issues in Linux

In this section, we describe some of the most common errors for IPoIB on Linux.

Ping does not work

Due to the complex nature of this problem, here we focus more or less on InfiniBand and try to give some hints for problem determination. Refer also to related literature or solution sources for IP.

Check the following conditions:

- ▶ Make sure cable connections of the source and target HCAs are firmly connected, and connected to the correct switch ports.
- ▶ Make sure the Linux modules `ib_ipoib`, `ib_sa`, `ib_mad`, and `hcad_mod` are loaded properly.
- ▶ Make sure eHCA has put an entry into the messages file indicating that the port is active.
- ▶ Check the `dmesg` command output.
- ▶ Confirm that the source and target IP address are not blocked in IPtables.
- ▶ Make sure the `ibv_devinfo` command works on both the source and destination system.
- ▶ Make sure the `ibv_ud_pingpong` works on both the source and target destinations.

If the conditions in the previous list check out positive, use the `arp` command to verify the system Address Resolution Protocol/ARP cache:

```
lib02:~ # arp -a -i ib0
? (192.168.178.71) at <incomplete> on ib0
```

If an entry with the target host name or IP address exists but is incomplete, this means that the target could not be reached (and thus resolved). In this case, check the physical connection between the adapter port and the switch.

If there is no entry with the target host name or IP address in ARP, check if the ports are registered properly in the multicast group in the switch by investigating its trace file.

If a port has not registered, this is the reason why ping does not work. In this case, verify again the connection between HCA and the switch and make sure the port is enabled. For Cisco/TopSpin switches, use the **TopspinEM** command to call Topspin Element Manager to check the port state. Figure 6-1 displays the status of our IB switch.



Figure 6-1 How check port states in Cisco Element Manager

As you can see in Figure 6-1, port #4 has a red color indicating that it is down. From the menu area, select **H**elp, then **L**egend to get a description of port states.

Figure 6-2 on page 223 presents an image generated with the Topology Viewer. Here you can also check if the switch identifies the HCAs in question.

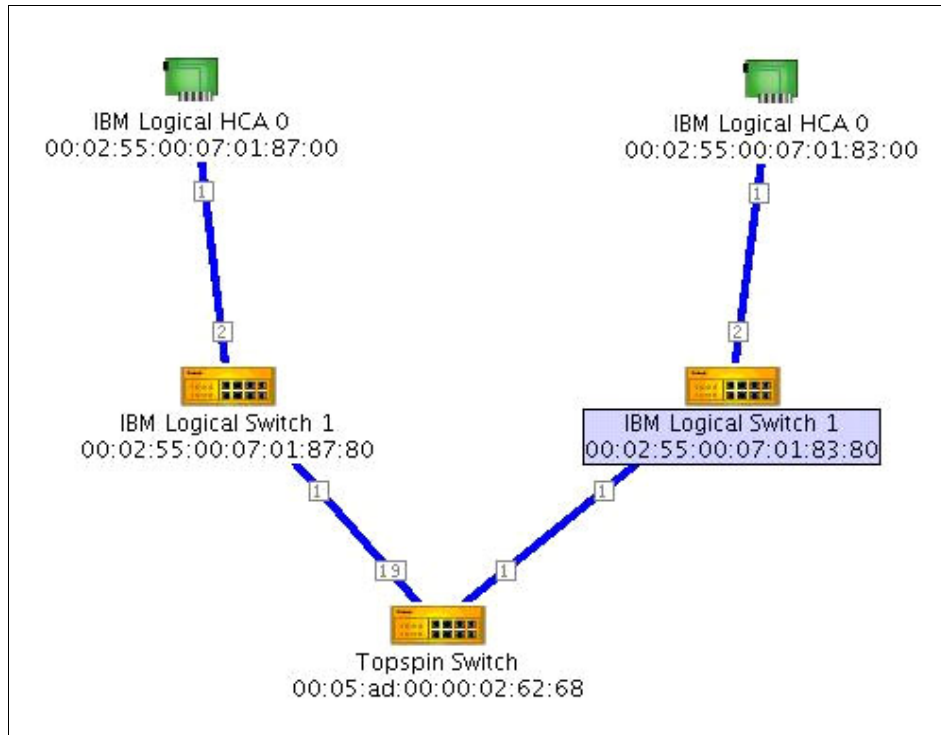


Figure 6-2 Topology view of InfiniBand network

Note: Topology Viewer identifies HCAs by their GID. If you do not see either the source or target HCA, you need to recheck the cables and adapters to make sure they are plugged in firmly.

If you have two ports (or more) plugged in into the same switch, you should set the `arp_ignore` flag to "2" on your system before you configure the IP address on the IB adapters (using the `ifconfig` command), as shown in Example 6-10.

Example 6-10 Setting the `arp_ignore` flag

```
# set arp_ignore flag
echo 2 > /proc/sys/net/ipv4/conf/all/arp_ignore
# verify arp_ignore flag
cat /proc/sys/net/ipv4/conf/all/arp_ignore
```

Getting the trace file from the switch

This section describes briefly how to obtain a trace file from a Topspin switch. For more details, refer to the *Cisco SFS 7000 Series Product Family Command Reference Guide, Release 2.5.0 (CISCO Customer Order Number: Text Part Number: OL-9163-01)*.

- ▶ Start a telnet session to the switch. You must have a user account and password to access this feature.
- ▶ By default, you are in User Exec mode. Enter the command **enable** in order to switch to Privileged Exec mode.
- ▶ Use the command **configure terminal** to enter Global Configuration mode. This is required since we want to enable debug traces on the switch.
- ▶ Issue the following command to turn on debug traces:

```
trace app 26 mod 10 level very-verbose flowmask 0x8002
```

For a detailed description of trace's options, refer to *Cisco SFS 7000 Series Product Family Command Reference Guide, Release 2.5.0 (CISCO Customer Order Number: Text Part Number: OL-9163-01)*.

- ▶ Leave the configuration mode by using the **exit** command.

All traces will be stored in a file named `ts_log`. Enter the command **dir syslog** to get a full list of all log files, as shown in Example 6-11.

Example 6-11 Dir Syslog command output sample

```
topspin> dir syslog
===== Existing Syslog-files on System =====
slot date-created          size      file-name
1   Mon May 22 10:59:15 2006 18998    hwif_log
1   Wed Oct 25 11:08:22 2006 527135   ts_log
1   Thu Nov  3 16:15:03 2005 39346    ts_log.1.gz
1   Tue Sep  6 10:30:03 2005 42368    ts_log.2.gz
```

The previous log files have an enumerated suffix and are compressed, that is, `ts_log.1.gz` and so on. In order to view and examine the log file more comfortably, transfer it to another system with an editor with better search or browsing capabilities, such as `vi` or `emacs`. Use the command **copy** to transfer the log file(s) from the switch to another host:

```
copy syslog:ts_log ftp://<username>:<password>@<host>/<filename>
```

Another option is to watch the log file in the switch console as you would with the **tail** command in Linux:

```
show logging end
```

6.4.4 Troubleshooting an HPC User Space issue under Linux

If you have an HPC system and your applications use the User Space Protocol, this section describes some errors you may encounter. This section presents a method for analyzing errors rather than a comprehensive set of errors.

For example, if you get the following message (in the application logs):

```
Couldn't open sysfs class 'infiniband_verbs'
```

Investigate it using the following command:

```
lib02:~ # ibv_devices
libibverbs: Fatal: couldn't open sysfs class 'infiniband_verbs'.
      device                               node GUID
      -----                               -
```

Make sure that kernel module `ib_uverbs` is loaded. If is not, issue `modprobe ib_uverbs` at the command prompt on the server.

Also check if `libsysfs.so` is installed properly. Use the `whereis` command to search for it:

```
lib02:~ # whereis libsysfs.so
libsysfs: /usr/lib/libsysfs.a /usr/lib/libsysfs.la /usr/lib/libsysfs.so
```

You can also check if `libsysfs.so` was installed using the commands shown in Example 6-12.

Example 6-12 Checking if libsysfs.so is installed (rpm method)

```
lib02:~ # rpm -qa | xargs rpm -ql | grep libsysfs.so
/usr/local/lib64/infiniband/libsysfs.so.1
/usr/local/lib64/infiniband/libsysfs.so.1.0.2
/lib/libsysfs.so.1
/lib/libsysfs.so.1.0.2
/usr/lib/libsysfs.so
```

If the library is not found, change to the directories `/lib`, `/lib64`, `/usr/local/lib`, `/usr/local/lib64` to search for it. Otherwise, you need to install `libsysfs`, as described in 5.1.2, “Software components and versions” on page 143.

Failed to open device

If you get the following error:

```
lib02:~ # ibv_devinfo -v
Failed to open device
```

The reason is that no device node(s) have been found in order to run a user space application such as **ibv_devinfo**. Issue the following command to create the device nodes:

```
lib02:~ # mkdir /dev/infiniband
lib02:~ # mknod /dev/infiniband/uverbs0 c 231 192
lib02:~ # mknod /dev/infiniband/uverbs1 c 231 193
```

No user space device-specific driver found for uverbs0

Let us say that the **ibv_devinfo** command returns the message shown in Example 6-13.

Example 6-13 No user space device loaded

```
lib02:~ # ibv_devinfo
libibverbs: Warning: couldn't load driver
/usr/local/lib/infiniband/libehca.so:
/usr/local/lib/infiniband/libehca.so: cannot open shared object file:
No such file or directory
libibverbs: Warning: no userspace device-specific driver found for
uverbs0
        driver search path: /usr/local/lib/infiniband
No IB devices found
```

Check if the path `/usr/local/lib/infiniband/libehca.so` is valid and accessible. If it exists, check if the executable, in this case **ibv_devinfo**, and the indicated library are compatible, that is, they must be both either 32-bit or 64-bit, but not mixed as shown in Example 6-14.

Example 6-14 Checking executable and library compatibility

```
lib02:~ # file /usr/local/bin/ibv_devinfo
/usr/local/bin/ibv_devinfo: ELF 32-bit MSB executable, PowerPC or Cisco
4500, version 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses
shared libs), not stripped
lib02:~ # file /usr/local/lib/infiniband/libehca.so.1.0.0
/usr/local/lib/infiniband/libehca.so.1.0.0: ELF 64-bit MSB shared
object, cisco 7500, version 1 (SYSV), not stripped
```

Debug traces from libehca

Let us assume the local `/usr/local/libehca/etc/libehca.conf` file has the following content:

```
# This is a comment
log.trlevel 6
log.filename /tmp/libehca.log
```

By default, `log.trlevel` is set to 6, which prints only error and warning messages. Set it to 9 in order to obtain the full debug traces.

The `log.filename` must point to a writable location. Use `STDOUT` or `STDERR` to redirect the traces to `stdout` or `stderr` respectively of the console.

```
log.trlevel 9
#log.filename /tmp/libehca.log
log.filename STDOUT
```

6.5 Application layer troubleshooting

Our team ran across some issues in the different management and application systems we used in our test environment.

6.5.1 LoadLeveler issues

The `llstatus -a` command reports that the status of an adapter port is `NOT READY`. Sometimes, you might find the status of all ports is `NOT READY` in a node. This means that an unspecified problem occurred when accessing the adapter state (see Example 6-15).

Example 6-15 Adapter port "NOT READY"

```
loadl@lib01:~> llstatus -a
... ..
=====
lib02.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt B1ks,101,READY)
network1833865768265265971s(striped,,,-1,0/0,0/0 rCxt B1ks,101,NOT READY)
network18338657682652659712(aggregate,,,-1,0/0,0/0 rCxt B1ks,1,NOT READY)
ib1(InfiniBand,lib02sw2,192.168.9.162,,2,0/0,0/0 rCxt B1ks,1,NOT READY,2)
network18338657682652659714(aggregate,,,-1,64/64,0/0 rCxt B1ks,1,READY)
ib0(InfiniBand,lib02sw1,192.168.8.162,,2,64/64,0/0 rCxt B1ks,1,READY,1)
... ..
```

You can see that the status of port ib1 is NOT READY on node lib02. If this happens, your parallel job may not be able to apply enough User Space windows to run. To debug the problem, you can follow the steps in the following sections.

Step 1. Check if PNSD is seeing the adapter as up

If all ports in a node are in the status of NOT READY, there is a possibility that the PNSD daemon is down. For more information about PNSD, refer to 5.2.2, “System management software for SLES9 clustering” on page 155.

The log file of the PNSD daemon is /tmp/serverlog. In order to check the log file, you need root access rights. You should read the information in the file to check and see if PNSD is working normally, as shown in Example 6-16.

Example 6-16 Check the log file of the PNSD daemon

```
lib02:~ # tail /tmp/serverlog
```

```
Wed Oct 25 20:48:28 2006  
Request from connection 2029  
Close connection 2029
```

```
Wed Oct 25 20:49:32 2006  
PNSD shutting down  
Save NAM table  
Save network tables and window states  
Save connections
```

If the log file shows that the PNSD daemon is not in an active state, use the command **startsrc -s pnsd** to start the daemon. Use **llstatus -a** to check the adapter status after a while (1 minute, for example) to see if the problem has been solved.

In our case, the PNSD log file shows that PNSD daemon is working normally, as shown in Example 6-17 on page 229.

Example 6-17 Check log file of PNSD daemon

```
lib02:~ # tail /tmp/serverlog
QUERY_ADAP_RESOURCES: device ehca0
node: 2 device: ehca0 type: 32 max port id: 2
  port 1 - ip: 0xc0a808a2 net: 0xfe80000000000002 LID: 131108 [up]
           GUID: fe 80 00 00 00 00 02 00 02 55 00 50 00 1c 3d
  port 2 - ip: 0xc0a809a2 net: 0xfe80000000000000 LID: 131104 [up]
           GUID: fe 80 00 00 00 00 00 00 02 55 00 50 00 1c 7d

Wed Oct 25 22:18:34 2006
Request from connection 28
Close connection 28
```

Be sure to also check the admin file for LoadLeveler if the previous actions do not resolve the issue.

Step 2 Check if the LoadLever admin file is correct

If you have defined an adapter stanza for an InfiniBand adapter in the LoadLeveler admin file `LoadL_admin`, verify if the keywords attribute in the admin file are consistent with the ones retrieved from RSCT Peer Domain.

Example 6-18 shows the attributes of keywords in the LoadLeveler admin file.

Example 6-18 Adapter stanza for lib02sw2 in LoadL_admin file

```
load1@lib02:~> more /home/load1/LoadL_admin
... ..
lib02sw2: type = adapter
         adapter_name = ib1
         network_type = InfiniBand
         interface_address = 192.168.9.162
         interface_netmask = 255.255.255.0
         interface_name = lib02sw2
         logical_id = 22
         adapter_type = InfiniBand
         device_driver_name = ehca0
         network_id = 18338657682652659712
         port_number = 2
... ..
```

Use the **llextRPD** command to show the adapter information saved in the RSCT Peer Domain database, as shown in Example 6-19.

Example 6-19 Adapter information of lib02sw2 in RPD database

```
load1@lib02:~> llextRPD

#llextRPD: Cluster = "ib_rpdomain" ID = "2VGgMFn9eYpf0SrGovVHwm" on Tue
Oct 24 21:41:49 2006
... ..
lib02sw2: type = adapter
         adapter_name = ib1
         network_type = InfiniBand
         interface_address = 192.168.9.162
         interface_netmask = 255.255.255.0
         interface_name = lib02sw2
         logical_id = 32
         adapter_type = InfiniBand
         device_driver_name = ehca0
         network_id = 18338657682652659712
         port_number = 2
... ..
```

From Example 6-19, you can see that the keyword `logical_id` in the `LoadL_admin` file is inconsistent with that in the RSCT Peer Domain database. The inconsistency maybe caused by re-configuring the InfiniBand switches.

To solve the problem, modify the keyword attributes in the `LoadLeveler` admin file to make them consistent with the RPD database. Then run the **llctl -g reconfig** command to activate changes. After that, run the **llstatus -a** command again to verify the status of the adapter ports.

6.5.2 CSM issues

This section describes some common problems and solutions for Cluster Systems Management (CSM).

Probemgr

CSM diagnostic probes are programs that diagnose a specific part of a cluster system or subsystem. By handling specific diagnostic tasks, they help you to maintain system performance across diverse CSM hardware and software configurations.

A first step in CSM problem determination is to verify CSM on the management server. This can be done by running a probe:

```
probemgr -p ibm.csm.ms -l 0
```

Note: We recommend you always run the **probemgr** with the option **-l 0**. This will give the most detailed output.

To run all predefined diagnostics probes, we issue this command:

```
probemgr -a -l 0
```

A list of all predefined diagnostic probes can be found in the *CSM for AIX 5L and Linux Administration Guide, SA23-1343*

Consideration: At the time of this writing, there were no InfiniBand probes available.

Node installation failed

If the installation fails for any reason and another run of **installnode** fails immediately, check the Mode attribute for the node and run the following command if the mode is set to installing:

```
chnode Mode=PreManaged -n <node_name>
```

Adapter not ready

CSM may falsely show that network adapters are not in the ready state. Wait a minute or two. CSM takes some time to refresh the status information. If that does not help, run the **csmdat** command and that will update the adapter status.

Node cannot be managed

You may get into a state where you cannot bring a node into the cluster. We have seen this problem when the node has `AllowedManageRequest` set to zero (0). You can use the **chnode** command to reset this attribute:

```
/opt/csm/bin/chnode -n <nodename> AllowManageRequest=1
```

RSCT problems

If **updatenode** fails to exchange keys after installing the CSM client filesets, the problem could be the IP name resolution for that node. Check your `/etc/hosts` or the DNS node definitions on the management server to resolve the node's host name resolution problem.

Depending on the security method you use for your cluster, you may encounter problems with public keys getting out of synchronization. This may happen when a node gets reinstalled. You can update the RSCT public keys for a node by running `/opt/csm/bin/updatenode -k <nodename>`. Note that you should have a trusted network before running this command to prevent another machine from inserting a public key in place of the real node.

Hardware control problems

The hardware control commands `rconsole` and `rpower` are sensitive to configuration changes made on the hardware control point, which is the HMC in our case. Deleting LPARs, creating new LPARs, and redefining just deleted LPARs may result in some “mysterious” hardware control problems. Commands such as `netboot` and `getadapters` depend on correct working hardware control and will fail if the `rconsole` and `rpower` commands are failing.

Errors shown in Example 6-20 are the result of configuration or usage errors. Many errors display messages, including error numbers. *CSM for AIX 5L and Linux Command and Technical Reference*, SA23-1345 contains a list of all CSM error numbers with an explanation and a suggested user response to solve the problem.

General information about hardware control can be found in *CSM for AIX 5L and Linux Administration Guide*, SA23-1343.

Example 6-20 Hardware control point problem

```
msib01:/tmp>rconsole -t -n aib01
[Enter ^Ec?' for help]
csp_console: 2651-872 The ConsolePortNum attribute is not defined in the CSM
database for node aib01
rconsole: 2651-993 Issuing the command "/opt/csm/bin/csp_console hmcib01p aib01 -1 -1
" gave a return code of 255. The routine will continue.
msib01:/tmp>rpower -a query
2651-636 [aib01] Invalid hardware control point address specified "hmcib01p"
aib02 on
aib03 on
aib04 on
```



Best practices

This chapter provides some best practices for working with pSeries InfiniBand clusters. This includes tips on:

- ▶ CSM
- ▶ AIX
- ▶ SLES
- ▶ Physical architectural considerations

7.1 CSM

The fundamental system administration model that CSM provides allows the administrator to store all information about a node on the management server. This not only enables you to manipulate the information for many nodes in one place, but also enables you to quickly rebuild nodes that fail.

7.1.1 CSM and NIM strategy

Extending this strategy of “everything in one place” to NIM resources will provide more benefits for maintaining the administrative infrastructure.

This can be easily implemented by specifying a file system parameter to the **nim_master_setup** command:

```
msib01: /> nim_master_setup -a file_system=/csminstall -a  
volume_group=csmvg -B
```

This will set up NIM resources in the /csminstall directory.

7.1.2 Back up CSM data

You should periodically back up your CSM data independent of any mksysb images that you may create. This will provide you with the ability to quickly restore your CSM data without restoring your entire system image. This is particularly important before major upgrades, or before performing operations that may potentially corrupt your CSM data. For a detailed description of CSM backup procedures, go to:

<http://www.redbooks.ibm.com/abstracts/tips0262.html?open>

7.1.3 NIM and resolv.conf

When setting up NIM, it is most convenient to use the **nim_setup_server** **nim_master_setup** command. This will create all the NIM resources necessary to install the AIX nodes. One of the resources created is a resolv.conf resource for the name resolution configuration file /etc/resolv.conf. This file is optional. In particular for small setups not connected to your company’s intranet or to the Internet, like in our lab setup, it is not necessary.

However, **nim_setup_server** **nim_master_setup** will silently fail if /etc/resolv.conf is not present. As a result, not all NIM resources needed for node installation will be created and the NIM install will fail. Therefore it is a good idea to have an /etc/resolv.conf file even if it is not necessary.

This behavior is a known issue. The APAR number of this issue is IY 88770. At the time of this writing, there is no fix available for this issue.

7.1.4 Nodegroups

When performing CSM operations, on many, but not all of the cluster nodes, the -n nodelist parameter of the commands can be very long. All CSM commands that support the -n nodelist parameter also support -N nodegroup parameters. CSM comes with some predefined nodegroups, shown in Example 7-1.

Example 7-1 CSM predefined nodegroups

```
msib01:/cfmroot>nodegrp
AIXNodes
APCNodes
AllNodes
AutoyastNodes
BMCNodes
BladeNodes
CSPNodes
EmptyGroup
FSPNodes
HMCNodes
KickstartNodes
LinuxNodes
ManagedNodes
MinManagedNodes
NIMNodes
PreManagedNodes
RedHatELAS3Nodes
RedHatELAS4Nodes
RedHatELES3Nodes
RedHatELES4Nodes
RedHatELWS3Nodes
RedHatELWS4Nodes
SLES81Nodes
SLES9Nodes
pSeriesNodes
ppcRedHatELAS3Nodes
ppcRedHatELAS4Nodes
ppcSLES81Nodes
ppcSLES9Nodes
xSeriesNodes
msib01@/cfmroot>nodegrp
AIXNodes
```

APCNodes
AllNodes
AutoyastNodes
BMCNodes
BladeNodes
CSPNodes
EmptyGroup
FSPNodes
HMCNodes
KickstartNodes
LinuxNodes
ManagedNodes
MinManagedNodes
NIMNodes
PreManagedNodes
RedHatELAS3Nodes
RedHatELAS4Nodes
RedHatELES3Nodes
RedHatELES4Nodes
RedHatELWS3Nodes
RedHatELWS4Nodes
SLES81Nodes
SLES9Nodes
aibnodes
pSeriesNodes
ppcRedHatELAS3Nodes
ppcRedHatELAS4Nodes
ppcSLES81Nodes
ppcSLES9Nodes
xSeriesNodes

Administrators can define their own nodegroups. This is very helpful for large or non-homogenous clusters.

If nodegroups are to be used in NIM operations, the CSM nodegroup definitions have to be converted to NIM nodegroup definitions. This is done by the **esm2nimgrps** command shown in Example 7-2 on page 237.

```
msib01:/tmp> csm2nimgrps -N aibnodes
msib01:/tmp> ls nim -l aibnodes
aibnodes:
  class   = groups
  type    = mac_group
  member1 = aib04
  member2 = aib01
  member3 = aib02
  member4 = aib03
```

7.2 Automatic InfiniBand configuration for many nodes

Currently CSM (V1.5.1) does not support installation and configuration of InfiniBand adapters and devices during node installation. However, InfiniBand adapter information can be collected with the **getadapters** command using the **dsh** method after AIX is up and running on the nodes.

The InfiniBand Connection Manager ICM and the InfiniBand IP devices can be configured with the **updatenode** command using this adapter information. This process needs a modified stanza file from the **getadapters** command. In large cluster configurations, modifying the stanza files is very time consuming and error-prone. Therefore this process should be automated.

The main issue with automatically setting up IP over InfiniBand devices is the mapping of IP addresses to InfiniBand adapters and ports. We recommend setting up a numbering scheme so that an IP address for IPoIB can be automatically derived from the node information available. As an alternative, a simple mapping file could be created manually and then transformed automatically to a stanza file for AIX or sysconfig file for SLES.

7.2.1 Configuring IB adapters in CSM for AIX

We wrote a simple Perl script (see Example 7-3) that reads a stanzafile and a map file. It then writes an updated stanzafile. File names are hardcoded in the script and little error checking is done. The output file should be reviewed carefully.

Example 7-3 Perl script to populate CSM database

```
#!/usr/bin/perl

# convert stanza file from "getatapters -m dsh -t iba .."   output
#                       to  "getadapters -W -f ..."     input

# modify filenames below as necessary
#
# Input file name
my $StanzaFileIn = "/tmp/myIBstanzafile";
# Output file name
my $StanzaFileOut = "/tmp/myIBstanzafileModified";
# mapfile
my $mapfile = "/tmp/IB2IPmap";

#####

sub stripvalue() {
    if ( $field[$j] =~ /^(\\w+=)\\S+/ ) { $field[$j] = $1 }
}

# read mapfile
open( MAP, $mapfile ) || die "Cannot open Mapfile $mapfile: $!\n";
while ( defined( $line = <MAP> ) ) {
    chomp $line;
    ( $node, $a, $b, $c, $d ) = split( " ", $line );
    $ip{"$node 1"} = $a; $mask{"$node 1"} = $b;
    $iname{"$node 1"} = "ib0";
    $ip{"$node 2"} = $c; $mask{"$node 2"} = $d;
    $iname{"$node 2"} = "ib1";
}
close(MAP);

# read input stanzafile
$node = "";
$i = -1;
open( INP, $StanzaFileIn )
```



```

|| die "Cannot open input stanza file $StanzaFileIn: $!\n";
open( OUT, ">$StanzaFileOut" )
|| die "Cannot open output stanza file $StanzaFileOut: $!\n";
while ( defined( $line = <INP> ) ) {
    chomp $line;
    if ( $line =~ /^\/s*#./ ) { # comment
        print OUT "$line\n"; next;
    }
    if ( $line =~ /^(w+)\.:/ ) { # new input stanza
        $node = $1; $i = 0; next;
    }
    if ( $line =~ /^\/s+(.+)$/ ) { $i++; $field[$i] = $1; next; }
    if ( $line =~ /^\/s*$/ ) {
        if ($node) { # an input stanza is complete
            foreach $p ( 1, 2 )
            { # output two IP stanzas for every adapter stanza
                $s = "$node $p";
                print OUT "$node:\n";
                for ( $j = 1 ; $j <= $i ; $j++ ) {
                    $parm = "";
                    if ( $field[$j] =~ /netaddr=/ ) {
                        stripvalue(); $parm = "$ip{$s}";
                    }
                    if ( $field[$j] =~ /subnet_mask=/ ) {
                        stripvalue(); $parm = "$mask{$s}";
                    }
                    if ( $field[$j] =~ /interface_name=/ ) {
                        stripvalue(); $parm = "$iname{$s}";
                    }
                    if ( $field[$j] =~ /ib_adapter=/ ) {
                        stripvalue(); $parm = "iba0";
                    }
                    if ( $field[$j] =~ /ib_port=/ ) {
                        stripvalue(); $parm = "$p";
                    }
                    if ( $field[$j] =~ /srq_size=/ ) {
                        stripvalue(); $parm = "";
                    }
                }
                print OUT "\t$field[$j]$parm\n";
            }
            print OUT "\n";
        }
    }
}
next;
}

```

```
}  
close(INP);  
close(OUT);
```

The input stanza file is created by the **getadapters** command described in 4.5.6, “Configuring InfiniBand adapters on AIX nodes” on page 117 (see also Example 4-25 on page 119). The map file has one line per node. Each line lists a node name and the two IP addresses and netmasks to be defined on this node. See Example 7-4 for a sample map file.

Example 7-4 Adapter map file

```
aib01 192.168.8.161 255.255.255.0 192.168.9.161 255.255.255.0  
aib02 192.168.8.162 255.255.255.0 192.168.9.162 255.255.255.0  
aib03 192.168.8.163 255.255.255.0 192.168.9.163 255.255.255.0  
aib04 192.168.8.164 255.255.255.0 192.168.9.164 255.255.255.0
```

The output file from our script can be written to the CSM database:

```
msi01:/tmp>getadapters -W -f /tmp/myIBstanzafileModified
```

Finally, we configure the adapters using the **updatenode -c** command using the file shown in Example 7-5.

Example 7-5 CSM adapter stanza file for IB adapters

```
###CSM_ADAPTERS_STANZA_FILE###--do not remove this line  
#---Stanza Summary-----  
#   Date: Tue Oct 10 11:32:25 EDT 2006  
#   Stanzas Added: 4  
#---End Of Summary-----
```

```
aib01:  
    adapter_type=iba  
    ib_adapter=  
    ib_port=1  
    interface_name=ib0  
    interface_type=ib  
    machine_type=secondary  
    mtu=2044  
    netaddr=  
    p_key=  
    srq_size=1  
    subnet_mask=
```

```
aib02:
```

```
adapter_type=iba
ib_adapter=
ib_port=1
interface_name=ib0
interface_type=ib
machine_type=secondary
mtu=2044
netaddr=
p_key=
srq_size=1
subnet_mask=
```

aib03:

```
adapter_type=iba
ib_adapter=
ib_port=1
interface_name=ib0
interface_type=ib
machine_type=secondary
mtu=2044
netaddr=
p_key=
srq_size=1
subnet_mask=
```

aib04:

```
adapter_type=iba
ib_adapter=
ib_port=1
interface_name=ib0
interface_type=ib
machine_type=secondary
mtu=2044
netaddr=
p_key=
srq_size=1
subnet_mask=
```

7.2.2 SLES

We wrote a simple script named `/data/scripts/ip_config` to configure the IP addresses of all the HCA ports in the cluster. Run the following command to configure IP addresses for port-1 (ib0) and port-2 (ib1) on all nodes:

```
dsh -a "/data/scripts/ip_config ib0"  
dsh -a "/data/scripts/ip_config ib1"
```

The executable script `/data/scripts/ip_config` is shown in Example 7-6.

Example 7-6 Script for configuring IP address of HCA

```
msib02:/data/scripts # cat ip_config  
#!/usr/bin/ksh  
export HOSTNAME=`hostname`  
export ip=`cat /data/scripts/hosts | grep $HOSTNAME | awk '{print $2}'`  
  
if [[ $# != 1 ]]  
then  
    echo "please use ib0 or ib1 as a parameter"  
    exit -1  
elif [[ $1 == "ib0" ]]  
then  
    subnet="192.168.8"  
elif [[ $1 == "ib1" ]]  
then  
    subnet="192.168.9"  
else  
    echo "please use ib0 or ib1 as a parameter"  
    exit -1  
fi  
  
echo "BOOTPROTO='static'" > /etc/sysconfig/network/ifcfg-$1  
echo "BROADCAST='$subnet.255'" >> /etc/sysconfig/network/ifcfg-$1  
echo "IPADDR='$subnet.$ip'" >> /etc/sysconfig/network/ifcfg-$1  
echo "MTU=''" >> /etc/sysconfig/network/ifcfg-$1  
echo "NETMASK='255.255.255.0'" >> /etc/sysconfig/network/ifcfg-$1  
echo "NETWORK='$subnet.0'" >> /etc/sysconfig/network/ifcfg-$1  
echo "REMOTE_IPADDR=''" >> /etc/sysconfig/network/ifcfg-$1  
echo "STARTMODE='onboot'" >> /etc/sysconfig/network/ifcfg-$1
```

Note: To execute `/data/scripts/ip_config`, you need to create another file named `/data/scripts/hosts`, as shown in Example 7-7.

Example 7-7 /data/script/hosts

#hostname	ip address
lib01	161
lib02	162
lib03	163
lib04	164

7.3 PowerPC productivity tools for SLES

IBM System p servers have some features that are not available on other platforms supported by SUSE Linux Enterprise Server (SLES). Therefore software support for these features is not integrated in SLES. IBM provides some RPM packages collectively called *Service and productivity tools for Linux on POWER systems* to make full use of the System p hardware.

In addition, they provide some tools for service and support to list and collect system specific information and error reports. For descriptions, current versions, and downloads, go to:

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html>

Some of these tools are necessary for running InfiniBand on SLES. For a list of some useful optional tools, see Table 7-1.

For users who are familiar with the IBM AIX operating system, most of these commands are well known.

Table 7-1 Productivity tools

Command	RPM package	Purpose
bootlist	powerpc-utils	The bootlist command sets the boot device for next system boot. Unlike fdisk , this command knows about AIX hdisks.
invscout	IBMinvscout	The Inventory Scout tool surveys one or more systems for hardware and software information. The gathered data can be used by Web services such as the Microcode Discovery Service, which generates a report indicating if installed microcode needs to be updated.
lscfg	lsvpd	lscfg lists hardware configuration information for the system and its components.
lsmcode	lsvpd	lsmcode lists hardware microcode and firmware levels.
lsvio	lsvpd	lsvio lists virtual I/O adapters and devices.

Command	RPM package	Purpose
ofpathname	powerpc-utils	ofpathname provides the ability to translate logical device names to their Open Firmware device path names for PowerPC-64 systems. It can also translate an Open Firmware device path to its logical device name.
log_repair_action	servicelog	The log_repair_action command creates an entry in the error log to indicate that the device at the specified location code has been repaired. When viewing a list of platform errors, all errors on the device at the specified location code prior to the specified date will be considered closed (fixed).
servicelog	servicelog	The servicelog command queries and displays the contents of the system servicelog. Events may be queried by their unique ID in the servicelog, or by some combination of parameters of the logged events, as specified by the command-line options.
uesensor	powerpc-utils-papr	The uesensor utility is used to view the state of environmental sensors on PowerPC-64 machines.
rtas_errd	diagela	The Error Log Analysis tool provides automatic analysis and notification of errors reported by the platform firmware on IBM eServer pSeries systems. Errors written to /var/log/platform are analyzed. If a corrective action is required, notification is sent to the Service Focal Point on the Hardware Management Console (HMC), if so equipped, or to users subscribed for notification through the file /etc/diagela/mail_list. The Serviceable Event sent to the Service Focal Point and listed in the e-mail notification may contain a Service Request Number.
snap	powerpc-utils	The snap script copies several system status and config files and the output of several commands from the system. System servicers may ask that this be run in order to collect data to diagnose a problem.

See Example 7-8 for a useful output of the **lscfg** command from the productivity tools.

Example 7-8 lscfg command output

```
lib01:~ # lscfg
INSTALLED RESOURCE LIST
```

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
* = Diagnostic support not available.

```
Model Architecture: chrp
Model Implementation: Multiple Processor, PCI Bus
```

```

+ sys0                               System Object
+ sysplanar0                          System Planar
+ pci0                                U787F.001.DPM18BL-P1 PCI Bus
+ eth0                                U787F.001.DPM18BL-P1-C4-T1
                                       IBM 10 Gigabit Ethernet SR DDR PCI-X
                                       Adapter (1410eb02)
+ pci1                                U787F.001.DPM18BL-P1 PCI Bus
+ pci2                                U787F.001.DPM18BL-P1 PCI Bus
+ pci3                                U787F.001.DPM18BL-P1 PCI Bus
+ scsi3                               U787F.001.DPM18BL-P1-C3-T1
                                       Emulex LightPulse Fibre Channel Host
                                       Adapter LP10000 Common
+ sde                                U787F.001.DPM18BL-P1-C3-T1-L0-L0
                                       Fibre Channel Disk Drive (26800 MB)
+ sdf                                U787F.001.DPM18BL-P1-C3-T1-L0-L1
                                       Fibre Channel Disk Drive (26800 MB)
+ sdg                                U787F.001.DPM18BL-P1-C3-T1-L1-L0
                                       Fibre Channel Disk Drive (26800 MB)
+ sdh                                U787F.001.DPM18BL-P1-C3-T1-L1-L1
                                       Fibre Channel Disk Drive (26800 MB)
+ pci4                                U787F.001.DPM18BL-P1 PCI Bus
+ scsi0-1                             U787F.001.DPM18BL-P1-T12
                                       ATA Adapter (5a107512)
+ scd0                                U787F.001.DPM18BL-P4-D2
                                       ATA CD-ROM Drive
+ pci5                                U787F.001.DPM18BL-P1 PCI Bus
+ scsi2                               U787F.001.DPM18BL-P1 SCSI I/O Controller (1410d302)
+ scsi2:0                             U787F.001.DPM18BL-P1-T10
                                       SCSI I/O Controller Channel
+ sg5                                U787F.001.DPM18BL-P1-T10-L15-L0
                                       SCSI Enclosure Services Device
+ sda                                U787F.001.DPM18BL-P1-T10-L3-L0
                                       16 Bit LVD SCSI Disk Drive (73400 MB)
+ sdb                                U787F.001.DPM18BL-P1-T10-L4-L0
                                       16 Bit LVD SCSI Disk Drive (73400 MB)
+ sdc                                U787F.001.DPM18BL-P1-T10-L5-L0
                                       16 Bit LVD SCSI Disk Drive (73400 MB)
+ sdd                                U787F.001.DPM18BL-P1-T10-L8-L0
                                       16 Bit LVD SCSI Disk Drive (73400 MB)
+ scsi2:1                             U787F.001.DPM18BL-P1-T11
                                       SCSI I/O Controller Channel
+ pci6                                U787F.001.DPM18BL-P1 PCI Bus
+ eth1                                U787F.001.DPM18BL-P1-C5-T1
                                       IBM 10/100/1000 Base-TX PCI-X Adapter

```

```

(14106902)
+ pci7          U787F.001.DPM18BL-P1 PCI Bus
+ pci8          U787F.001.DPM18BL-P1 PCI Bus
+ eth4          U787F.001.DPM18BL-P1-C1-T1
                  IBM 10/100/1000 Base-TX PCI-X Adapter
                  (14106902)
+ pci9          U787F.001.DPM18BL-P1 PCI Bus
+ pci10         U787F.001.DPM18BL-P1 PCI Bus
+ usb1          U787F.001.DPM18BL-P1 USB Host Controller (33103500)
+ usb2          U787F.001.DPM18BL-P1 USB Host Controller (33103500)
+ usb3          U787F.001.DPM18BL-P1 USB Host Controller (3310e000)
+ pci11         U787F.001.DPM18BL-P1 PCI Bus
+ eth2          U787F.001.DPM18BL-P1-T5
                  Port 1 - IBM 2 PORT 10/100/1000
                  Base-TX PCI-X Adapter (14108902)
+ eth3          U787F.001.DPM18BL-P1-T6
                  Port 2 - IBM 2 PORT 10/100/1000
                  Base-TX PCI-X Adapter (14108902)
+ L2cache0     L2 Cache
+ mem0         Memory
+ proc0        Processor
+ proc1        Processor

```

Note: At the time of this writing, the current version for the `lscfg` command from the `lsvpd-0.15.1-1` package does not show the InfiniBand GX adapter.

7.4 Physical server build considerations

In the following paragraphs, we share some ideas that will hopefully help your implementation of IB go more smoothly. This section is primarily concerned with cable runs and connections.

When using large switch designs, such as the 96 port Cisco 7008p, it is important to consider the physical layout (see Figure 7-1 on page 247) of the space into which the servers and switches will be installed. Because of the limited lengths of the InfiniBand cables, it is necessary to consider all of the space elements to make sure that your servers will be within reach of the cables that will be used.

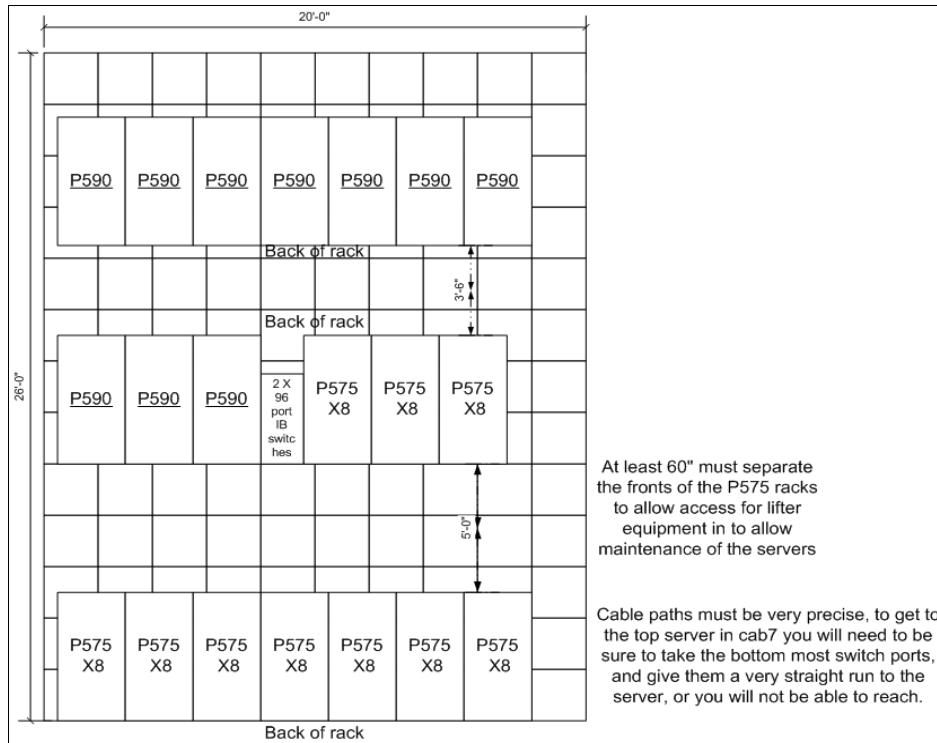


Figure 7-1 Rack and floor space diagram

Other things to consider

The floor tile behind each server should be completely open for the full width of the tile, which will allow the IB cables to come straight up into the rack.

Be sure to allow distance in your cable runs for the cables to be “dressed into” the cabinet. The short bends that the cables take as they go up the sides of the rack and the bend that the cables take as they go into the adapter can account for about 2 feet of cable length.

It is necessary to bring the cables in from the side as the cables are quite bulky and can substantially reduce the airflow out of the back of the server. This can have negative thermal impacts to the server.

Brushes should be used to fill in the remainder of the space in the holes in the floor tile. This helps to improve the efficiency of the cooling systems in the datacenter.

A minimum of 60 inches service clearance is needed in front of a p575 rack so that the servers can be slid out and worked on by a service engineer. However, it is recommended that 70 inches be allocated for this to give room for rack doors and airflow to the servers across from the server being worked on. How this impacts your IB fabric design must be considered when you run cables and the cable lengths you purchase.



Monitoring tools for InfiniBand adapter

This chapter provides an introduction to InfiniBand monitoring concepts.

- ▶ Monitoring tools for AIX 5L and SLES 9
- ▶ Useful commands for LoadLeveler with InfiniBand

8.1 Monitoring tools for AIX 5L and SLES 9

After finishing the installation of the management system and each node, we have to check how the InfiniBand adapter is installed and is operated correctly. In this chapter, we provide you with hints and tips on how to verify and operate the InfiniBand adapter on your servers running AIX 5L or SLES 9.

8.1.1 Useful commands for AIX 5L

topas command

The **topas** command is a performance monitoring tool that is ideal for broad spectrum performance analysis. However, this chapter talks about looking at the **topas** command and output, as shown in Example 8-1.

Example 8-1 Sample topas output of InfiniBand adapter

Topas Monitor for host:		aib04		EVENTS/QUEUES		FILE/TTY	
Tue Oct 3 09:50:32 2006		Interval: 2		Cswitch	815	Readch	2205.4K
				Syscall	1320	Writech	2129.8K
Kernel	1.6	#		Reads	223	Rawin	0
User	0.3	#		Writes	295	Ttyout	284
Wait	0.0			Forks	0	Igets	0
Idle	98.1	#####		Execs	0	Namei	561
				Runqueue	0.0	Dirblk	0
Network	KBPS	I-Pack	O-Pack	KB-In	KB-Out	Waitqueue	0.0
ib0	2334.7	1170.5	241.0	2334.7	0.0		
en3	0.4	1.0	0.5	0.0	0.3	PAGING	MEMORY
ib1	0.0	0.0	0.0	0.0	0.0	Faults	4 Real,MB 3712
						Steals	0 % Comp 20.7
Disk	Busy%	KBPS	TPS	KB-Read	KB-Writ	PgspIn	0 % Noncomp 27.0
hdisk0	100.0	2154.0	247.0	0.0	2154.0	PgspOut	0 % Client 27.0
hdisk3	0.0	0.0	0.0	0.0	0.0	PageIn	0
hdisk1	0.0	0.0	0.0	0.0	0.0	PageOut	303 PAGING SPACE
						Sios	303 Size,MB 512
Name	PID	CPU%	PgSp	Owner			% Used 1.2
xmwlM	250042	0.2	0.6	root	NFS (calls/sec)	% Free	98.7
CqKp	151626	0.2	0.4	root	ServerV2	0	
topas	221334	0.1	1.0	root	ClientV2	0	Press:
getty	311454	0.0	0.4	root	ServerV3	0	"h" for help
gil	69666	0.0	0.9	root	ClientV3	0	"q" to quit

ibstat command

The **ibstat** command displays InfiniBand operational information pertaining to a specified Host Channel Adapter Device (HCAD). If an HCAD device name is not entered, the status for all available HC.ADs are displayed as shown in Example 8-2 on page 251

Example 8-2 ibstat command sample output

```
aib03@/>ibstat -v
=====
  INFINIBAND DEVICE INFORMATION (iba0)
=====
Infiniband Debug Disabled

-----
  IB NODE INFORMATION (iba0)
-----
Number of Ports:                2
Globally Unique ID (GUID):      00.02.55.00.50.00.08.00
Maximum Number of Queue Pairs:  16367
Maximum Outstanding Work Requests: 32768
Maximum Scatter Gather per WQE: 252
Maximum Number of Completion Queues: 16380
Maximum Multicast Groups:       256
Maximum Memory Regions:         61382
Maximum Memory Windows:         61382

-----
  IB PORT 1 INFORMATION (iba0)
-----
Global ID Prefix:                fe.80.00.00.00.00.00.02
Local ID (LID):                  0009
Port State:                       Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
  P_Key[0]:                       ffff
Current Number of GUID's:         1
Globally Unique ID List:
  GUID[0]:                         00.02.55.00.50.00.08.3d

-----
  IB PORT 2 INFORMATION (iba0)
-----
Global ID Prefix:                fe.80.00.00.00.00.00.00
Local ID (LID):                  0015
Port State:                       Active
Maximum Transmission Unit Capacity: 2048
Current Number of Partition Keys: 1
Partition Key List:
  P_Key[0]:                       ffff
Current Number of GUID's:         1
Globally Unique ID List:
  GUID[0]:                         00.02.55.00.50.00.08.7d

-----
```

IB INTERFACE ARP TABLE

```
-----  
aib04sw2 (192.168.9.164) at slid:0x0015 sqp:0x003b dlid:0x0013 rqp:0x001d  
DGID:fe:80:00:00:00:00:00:00:02:55:00:50:00:22:7d  
aib04sw1 (192.168.8.164) at slid:0x0009 sqp:0x003a dlid:0x000a rqp:0x001c  
DGID:fe:80:00:00:00:00:00:02:00:02:55:00:50:00:22:3d
```

Total number of entries: 2

IB INTERFACE (ib0) INFORMATION

```
-----  
ib0: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>  
inet 192.168.8.163 netmask 0xffffffff broadcast 192.168.8.255  
tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

device		N/A	True
ib_adapter	iba0	Infiniband Host Channel Adapter	True
ib_port	1	Infiniband Host Channel Adapter Port	True
mtu	2044	Maximum IP Packet Size for This Device	True
netaddr	192.168.8.163	Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
p_key	-1	Partition Key	True
rfc1323	1	Enable/Disable TCP RFC 1323 Window Scaling	True
srq_size	4000	Transmit or Receive Hardware Queue Size	True
state	up	Current Interface Status	True
tcp_recvspace	131072	Set Socket Buffer Space for Receiving	True
tcp_sendspace	131072	Set Socket Buffer Space for Sending	True

IB INTERFACE (ib1) INFORMATION

```
-----  
ib1: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>  
inet 192.168.9.163 netmask 0xffffffff broadcast 192.168.9.255  
tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

device		N/A	True
ib_adapter	iba0	Infiniband Host Channel Adapter	True
ib_port	2	Infiniband Host Channel Adapter Port	True
mtu	2044	Maximum IP Packet Size for This Device	True
netaddr	192.168.9.163	Internet Address	True
netmask	255.255.255.0	Subnet Mask	True
p_key	-1	Partition Key	True
rfc1323	1	Enable/Disable TCP RFC 1323 Window Scaling	True
srq_size	4000	Transmit or Receive Hardware Queue Size	True
state	up	Current Interface Status	True
tcp_recvspace	131072	Set Socket Buffer Space for Receiving	True
tcp_sendspace	131072	Set Socket Buffer Space for Sending	True

The following fields display information for all valid calls:

Device Name Displays the name of an available HCAD (for example, iba0).

Port State

Down	Port is disabled.
Initialized	Port is enabled and issuing training sequences.
Down	Port is trained and attempting to configure to the active states.
Active	Port is in a normal operational state.
Unknown	Port is in an invalid or unknown state.

lsrsrc command

This command displays the persistent and dynamic attributes and their values for a resource or a resource class. We can find the InfiniBand network interface attributes in Example 8-3. And the **lsrsrc** command can also use the SLES 9.

Example 8-3 lsrsrc command output of SLES9

```
lib02:~ # lsrsrc IBM.NetworkInterface
Resource Persistent Attributes for IBM.NetworkInterface
.
.....<< Omitted lines for ethernet (enX) interfaces >>.....
.
resource 2:
    Name = "ib1"
    DeviceName = "ehca0"
    IPAddress = "192.168.9.162"
    SubnetMask = "255.255.255.0"
    Subnet = "192.168.9.0"
    CommGroup = "CG2"
    HeartbeatActive = 1
    Aliases = {}
    DeviceSubType = 32
    LogicalID = 131104
    NetworkID = 0
    NetworkID64 = 18338657682652659712
    PortID = 2
    HardwareAddress = "fe80000000000000000000002550050001c7d"
    DevicePathName = ""
    ActivePeerDomain = "ib_rpdomain"
    NodeNameList = {"lib02"}
resource 3:
```

```

Name           = "ib0"
DeviceName     = "ehca0"
IPAddress      = "192.168.8.162"
SubnetMask     = "255.255.255.0"
Subnet         = "192.168.8.0"
CommGroup      = "CG3"
HeartbeatActive = 1
Aliases        = {}
DeviceSubType  = 32
LogicalID      = 131108
NetworkID      = 2
NetworkID64    = 18338657682652659714
PortID         = 1
HardwareAddress = "fe800000000000020002550050001c3d"
DevicePathName = ""
ActivePeerDomain = "ib_rpdomain"
NodeNameList   = {"lib02"}

```

8.1.2 Monitoring tools for SLES 9

Here we discuss monitoring tools for SLES 9.

csmstat command

If CSM is running and managing each SLES 9 node, you can get the information about network interfaces using the **csmstat** command, as shown in Example 8-4.

Example 8-4 csmstat command output

```

msib02:~ # csmstat
-----
Hostname           HWControlPoint  Status  PowerStatus  Network-Interfaces
-----
lib01.itso.ibm.c~ hmcib01         on      on            eth3-Online
ib0-Online ib1-Online
lib02.itso.ibm.c~ hmcib01         on      on            eth3-Online
ib0-Online ib1-Online
lib03.itso.ibm.c~ hmcib01         off     on            unknown
lib04.itso.ibm.c~ hmcib01         off     on            unknown

```

ibv_devices command

This command is for listing InfiniBand devices available for use with the user space protocol. See Example 8-5 on page 255.

Example 8-5 ibv_devices sample output

```
lib01:/usr/local/bin # ibv_devices
device                node GUID
-----                -
```

ehca0	0002550050002700
-------	------------------

ibv_devinfo command

This command is for print information about InfiniBand devices available for use from user space, as shown in Example 8-6.

Example 8-6 ibv_devinfo sample output

```
lib01:/usr/local # ibv_devinfo -v
hca_id: ehca0
  node_guid:                0002:5500:5000:2700
  sys_image_guid:           0000:0000:0000:0000
  vendor_id:                 0x5076
  vendor_part_id:           0
  hw_ver:                    0x1000002
  phys_port_cnt:            2
  max_mr_size:               0x100000000
  page_size_cap:            0x0
  max_qp:                    16361
  max_qp_wr:                 32768
  device_cap_flags:         0x00000000
  max_sge:                   252
  max_sge_rd:               0
  max_cq:                    16376
  max_cqe:                   -64
  max_mr:                    61382
  max_pd:                    2147483647
  max_qp_rd_atom:           0
  max_ee_rd_atom:           0
  max_res_rd_atom:          0
  max_qp_init_rd_atom:      0
  max_ee_init_rd_atom:      0
  atomic_cap:                ATOMIC_NONE (0)
  max_ee:                    0
  max_rdd:                   0
  max_mw:                    61382
  max_raw_ipv6_qp:          0
  max_raw_ethy_qp:          0
  max_mcast_grp:            256
  max_mcast_qp_attach:      8
```

```

max_total_mcast_qp_attach:    256
max_ah:                       2147483647
max_fmr:                      61382
max_map_per_fmr:             0
max_srq:                      0
max_pkeys:                   16
local_ca_ack_delay:          0
    port: 1
        state:                 PORT_ACTIVE (4)
        max_mtu:               2048 (4)
        active_mtu:           2048 (4)
        sm_lid:                4
        port_lid:              24
        port_lmc:              0x02
        max_msg_sz:            0x0
        port_cap_flags:        0x00000000
        max_vl_num:            0
        bad_pkey_cntr:         0x0
        qkey_viol_cntr:        0x0
        sm_sl:                 0
        pkey_tbl_len:          1
        gid_tbl_len:           1
        subnet_timeout:        8
        init_type_reply:       0
        active_width:          12X (8)
        active_speed:           2.5 Gbps (1)
        phys_state:            invalid physical state
(0)
                                GID[ 0]:
fe80:0000:0000:0002:0002:5500:5000:273d

```

As Example 8-6 on page 255 shows, **ibv_devinfo -v** displays the capabilities of each HCA, in this case, of eHCA. For example, the field `max_qp` tells the maximum number of queue pairs that can be allocated totally.

Another command that lists all available HCAs is shown in Example 8-7 on page 257.

Example 8-7 ibv_devinfo -l command output

```
lib01:/usr/local # ibv_devinfo -l
1 HCA found:
    ehca0
```

The flags introduced to the **ibv_devinfo** command have the following meanings:

- d, --ib-dev=DEVICE** Use the IB device DEVICE (the default is the first device found).
- i, --ib-port=PORT** Query the port PORT (the default is all ports).
- l, --list** Only list the names of InfiniBand devices.
- v, --verbose** Print all available information about the device.

To monitor the real-time status of HCA, you may read information about the files under the directory `/sys/bus/ibmebus/devices` (see Example 8-8), in our case, `/sys/bus/ibmebus/devices/.001.DPM18BL-P1-C21`.

Example 8-8 Real-time status of HCA

```
lib01:/sys/bus/ibmebus/devices/.001.DPM18BL-P1-C21 # ls
.          cur_eq  detach_state  max_cq  max_pd
..         cur_mr  devspec       max_eq  max_qp
adapter_handle cur_mw  hw_ver       max_mr  name
cur_cq     cur_qp  max_ah       max_mw  num_ports

lib01:/sys/bus/ibmebus/devices/.001.DPM18BL-P1-C21 # cat cur_qp
6
lib01:/sys/bus/ibmebus/devices/.001.DPM18BL-P1-C21 # cat max_qp
16367
```

As you can see in Example 8-8, the current number of queue pairs is 6, while the maximum number of queue pairs is 16 K.

For more information about queue pairs, refer to 4.2.1, “Implementation of InfiniBand architecture (IBA) on System p5” on page 86.

ibv_ud_pingpong

This command provides a simple UD transport test and requires two communication partners, that is, server and client. The server needs to run first and waits for packets from a client. The client sends a number of packets to the given server, and the server replies on each received packet. Thus, **ibv_ud_pingpong** is a very helpful tool in order to verify the interconnections within an InfiniBand network at a low level as it uses IB verbs, i.e. native InfiniBand.

As IB verbs requires that both communication partners need to know some InfiniBand specific parameters like destination queue pair number and destination LID before establishing any data transfer, **ibv_ud_pingpong** uses a socket connection in order to exchange the mentioned data. Make sure that server and client can reach each other, for example, by a ping over IP test.

Example 8-9 shows how to start a server and client on the same partition.

Example 8-9 ibv_ud_pingpong command output

```
# start the server
lib01:/usr/local # ibv_ud_pingpong
  local address: LID 0x0018, QPN 0x00009c, PSN 0x58a748
#start the client
lib01:/usr/local/bin # ibv_ud_pingpong localhost
  local address: LID 0x0018, QPN 0x00009d, PSN 0x7edc7e
  remote address: LID 0x0018, QPN 0x00009c, PSN 0x58a748
4096000 bytes in 0.02 seconds = 2159.06 Mbit/sec
1000 iters in 0.02 seconds = 15.18 usec/iter
```

If running the client on another partition or machine, replace localhost with the server's host name or IP address.

The flags introduced to the **ibv_ud_pingpong** command have the following meanings.

Usage:

ibv_ud_pingpong Start a server and wait for connection.

ibv_ud_pingpong <host>

Connect to server at <host>.

Options:

-p, --port=<port>	Listen on/connect to port <port> (default 18515).
-d, --ib-dev=<dev>	Use IB device <dev> (default first device found).
-i, --ib-port=<port>	Use port <port> of IB device (default 1).
-s, --size=<size>	Size of message to exchange (default 2048).
-r, --rx-depth=<dep>	Number of receives to post at a time (default 500).
-n, --iters=<iters>	Number of exchanges (default 1000).
-e, --events	Sleep on CQ events (default poll).

A more detailed explanation:

Option -p	Specifies the socket port number ibv_ud_pingpong uses to exchange InfiniBand specific data as described above. The default port number is 18515. Using another number allows you to run this command multiple times on the same partition. Note that both server and client instances must use the same port number.
Option -d	Specifies the HCA device name. If called without this option, ibv_ud_pingpong will use the first available device, that is, ehca0. Specify ehca1, ehca2, and so forth in order to instruct ibv_ud_pingpong to use a second, third, and so on adapter respectively.
Option -i	Specifies the port to be used. The default is port 1. Specify 2 for port 2.
Option -s	Specifies the message size. The default is 2048.
Option -r	Specifies the so-called receive depth N, which instructs ibv_ud_pingpong to post N receive work requests before polling for packets coming in. The default is 500. Increasing this number will avoid completion queue overflow that occurs if the receiving path is slower than the sending path for some reasons.
Option -n	Specifies the number of exchanges between client and server. The default is 1000.
Option -e	Instructs ibv_ud_pingpong to utilize completion queue events mechanism to detect received messages. The default without this option is polling.

ibv_rc_pingpong

This command provides a simple RC transport test and functions similar to **ibv_ud_pingpong** as previously explained, and shown in Example 8-10.

Example 8-10 *ibv_rc_pingpong* command output

```
#start the server on lib01 - lib01sw1: ib adapter
lib01:/usr/local # ibv_rc_pingpong
  local address: LID 0x0018, QPN 0x0000a0, PSN 0xb135d5
#start the client on lib02
lib02:/usr/local/bin # ibv_rc_pingpong lib01sw1
  local address: LID 0x0024, QPN 0x00009d, PSN 0xf02b2
  remote address: LID 0x0018, QPN 0x0000a0, PSN 0xb135d5
8192000 bytes in 0.02 seconds = 2898.16 Mbit/sec
1000 iters in 0.02 seconds = 22.61 usec/iter
```

The flags introduced to the **ibv_rc_pingpong** command have the following meanings.

Usage:

ibv_rc_pingpong Start a server and wait for connection.

ibv_rc_pingpong <host>
Connect to server at <host>.

Options:

-p, --port=<port> Listen on/connect to port <port> (the default is 18515).
-d, --ib-dev=<dev> Use the IB device <dev> (the default is the first device found).
-i, --ib-port=<port> Use the port <port> of the IB device (the default is 1).
-s, --size=<size> Size of message to exchange (the default is 4096).
-m, --mtu=<size> Path MTU (the default is 1024).
-r, --rx-depth=<dep> Number of receives to post at a time (the default is 500).
-n, --iters=<iters> Number of exchanges (the default is 1000).
-e, --events Sleep on CQ events (the default is poll).

All other options have the same meaning and usage as explained above except for option -m.

Option -m Specifies the so-called patch MTU. The default is 1024. The same MTU size must be used for both server and client. Note that eHCA allows a maximum MTU of 2048.

Example 8-11 shows the **ibv_rc_pingpong** using options -m and -s.

Example 8-11 ibv_rc_pingpong usage with option m and s

```
# start the server on lib01
lib01:/usr/local # ibv_rc_pingpong -m 2048 -s 4096
  local address: LID 0x0018, QPN 0x0000a1, PSN 0x3c9a83
# start the client on lib02 - lib01sw1: ib adapter
lib02:/usr/local/bin # ibv_rc_pingpong -m 2048 -s 4096 lib01sw1
  local address: LID 0x0024, QPN 0x00009e, PSN 0xe8b89e
  remote address: LID 0x0018, QPN 0x0000a1, PSN 0x3c9a83
8192000 bytes in 0.02 seconds = 2957.13 Mbit/sec
1000 iters in 0.02 seconds = 22.16 usec/iter
```

ibv_uc_pingpong

This command provides a simple UC transport test and functions similar to **ibv_rc_pingpong**. See Example 8-12.

Example 8-12 ibv_uc_pingpong command output

```
# start the server on lib01
lib01:/usr/local # ibv_uc_pingpong
  local address: LID 0x0018, QPN 0x0000a7, PSN 0x8303b4
# start the client on lib02 - lib01sw1: ib adapter
lib02:/usr/local/bin # ibv_uc_pingpong lib01sw1
  local address: LID 0x0024, QPN 0x0000a4, PSN 0x2d1aa5
  remote address: LID 0x0018, QPN 0x0000a7, PSN 0x8303b4
8192000 bytes in 0.02 seconds = 2909.61 Mbit/sec
1000 iters in 0.02 seconds = 22.52 usec/iter
```

Refer to the **ibv_rc_pingpong** command for available options.

8.2 Useful commands for LoadLeveler with InfiniBand

Several LoadLeveler commands have been updated for InfiniBand adapter support. This section describes the basic commands of the LoadLeveler and how to show that the InfiniBand adapter is up and running.

llextrPD command

This command extracts the data for LoadLeveler to set up the administration file. The **llextrPD** command must be run on one of the nodes in an active RSCT peer domain to obtain the RSCT peer nodes and network interface data from that cluster.

The `port_number` keyword specifies the port number of the InfiniBand adapter port. The adapter stanza for InfiniBand support only contains the adapter port information. There is no InfiniBand adapter information in the adapter stanza. Example 8-13 shows an adapter stanza.

Example 8-13 llextrPD command sample output

```
lib01:/opt/ibm11/LoadL/full/bin # llextrPD -a lib01.itso.ibm.com

#llextrPD: Cluster = "ib_peerdomain" ID = "3sAmzEn9eYpf0RIBJSLoCm" on
Mon Oct 16 16:54:34 2006

lib01: type = machine
llextrPD: 2512-688 The "-a" parameter was specified but no adapters of
type "lib01.itso.ibm.com" could be found for the OSI (machine) "lib01".
The IBM.PeerNode name is used for this machine stanza name.
    adapter_stanzas = lib01sw2 lib01sw1 lib01.itso.ibm.com
    alias = lib01sw2 lib01sw1 lib01.itso.ibm.com

lib01sw2: type = adapter
    adapter_name = ib1
    network_type = InfiniBand
    interface_address = 192.168.9.161
    interface_netmask = 255.255.255.0
    interface_name = lib01sw2
    logical_id = 22
    adapter_type = InfiniBand
    device_driver_name = ehca0
    network_id = 18338657682652659712
    port_number = 2

lib01sw1: type = adapter
    adapter_name = ib0
    network_type = InfiniBand
    interface_address = 192.168.8.161
    interface_netmask = 255.255.255.0
    interface_name = lib01sw1
    logical_id = 7
    adapter_type = InfiniBand
    device_driver_name = ehca0
    network_id = 18338657682652659714
    port_number = 1
```

llstatus command

The `llstatus -a` command now displays the port number on each InfiniBand adapter. The `llstatus` command does not show port information for adapters that are not InfiniBand adapters. Example 8-14 shows the `llstatus -a` command output.

Example 8-14 llstatus -a sample output

```
lib01:/opt/ibm11/LoadL/full/bin # llstatus -a
=====
msib02.itso.ibm.com
=====
lib02.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,64/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib02sw2,192.168.9.162,,2,64/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib02sw1,192.168.8.162,,2,64/64,0/0 rCxt Blks,1,READY,1)
=====
lib01.itso.ibm.com
ehca0(InfiniBand,,,-1,0/0,0/0 rCxt Blks,101,READY)
network1833865768265265971s(striped,,,-1,64/64,0/0 rCxt Blks,101,READY)
network18338657682652659712(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib1(InfiniBand,lib01sw2,192.168.9.161,,1,64/64,0/0 rCxt Blks,1,READY,2)
network18338657682652659714(aggregate,,,-1,64/64,0/0 rCxt Blks,1,READY)
ib0(InfiniBand,lib01sw1,192.168.8.161,,1,64/64,0/0 rCxt Blks,1,READY,1)
```

llq command

The `llq -l` command has been updated to include the port number for the InfiniBand resources used by the running job. Example 8-15 is a fragment of the `llq -l` command output.

Example 8-15 llq command output

```
load1@lib02:/data/HPC/TEST/poetest.bw> llq -l lib01.itso.ibm.com.23
===== Job Step lib01.itso.ibm.com.23.0 =====
      Job Step Id: lib01.itso.ibm.com.23.0
      Job Name: lib01.itso.ibm.com.23
      Step Name: 0
      Structure Version: 10
      Owner: ibitso
      Queue Date: Mon 23 Oct 2006 03:00:41 PM EDT
      Status: Running
      .
      .
      .
      .
      .
Master Task
-----

      Executable   : /data/HPC/TEST/poetest.bw/lltest.cmd
      Exec Args    :
      Num Task Inst: 1
      Task Instance: lib01:-1,

Task
----

      Num Task Inst: 2
      Task Instance: lib01:0:ib1(MPI,US,22,Shared,0 rCxt
Blks,2),ib0(MPI,US,87,Shared,0 rCxt Blks,1),
      Task Instance: lib02:1:ib1(MPI,US,22,Shared,0 rCxt
Blks,2),ib0(MPI,US,87,Shared,0 rCxt Blks,1),

1 job step(s) in queue, 0 waiting, 0 pending, 1 running, 0 held, 0
preempted
```

llsummary command

The `llsummary -l -x` command has been updated to include the port number for the allocated hosts and task instances. The following listing is a fragment of the `llsummary -l -x` command output, as shown in Example 8-16.

Example 8-16 llsummary command output

```
===== Job msib02.itso.ibm.com.31 =====
      Job Id: msib02.itso.ibm.com.31
      Job Name: msib02.itso.ibm.com.31
      Structure Version: 210
      Owner: ibitso
      Unix Group: users
      Submitting Host: msib02.itso.ibm.com
      Submitting Userid: 1000
      Submitting Groupid: 100
      Number of Steps: 1
----- Step msib02.itso.ibm.com.31.0 -----
      Job Step Id: msib02.itso.ibm.com.31.0
      Step Name: 0
      Queue Date: Mon 23 Oct 2006 02:23:25 PM EDT
      Job Accounting Key: 4989153432599317796
      Dependency:
      Status: Completed
      Dispatch Time: Mon 23 Oct 2006 02:23:25 PM EDT
      .
      .
      .
      .
      .
      .
Node
----

      Name          :
      Requirements  :
      Preferences   :
      Node minimum  : 2
      Node maximum  : 2
      Node actual   : 2
      Allocated Hosts : lib01.itso.ibm.com:PENDING:ib1(MPI,US,29,Shared,0
rCxt Blks,2)ib0(MPI,US,94,Shared,0
rCxt Blks,1)
      + lib02.itso.ibm.com:PENDING:ib1(MPI,US,29,Shared,0
rCxt Blks,2)ib0(MPI,US,94,Shared,0
```

rCxt Blks,1)

Master Task

Executable : /data/HPC/TEST/poetest.bw/medium.cmd
Exec Args :
Num Task Inst: 1
Task Instance: lib01:-1

Task

Num Task Inst: 2
Task Instance: lib01:0:ib1(MPI,US,29,Shared,0 rCxt
Blks,2),ib0(MPI,US,94,Shared,0 rCxt Blks,1)
Task Instance: lib02:1:ib1(MPI,US,29,Shared,0 rCxt
Blks,2),ib0(MPI,US,94,Shared,0 rCxt Blks,1)

Messages Guide for InfiniBand in LoadLeveler

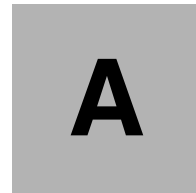
The following new messages have been added to the *IBM Tivoli Workload Scheduler LoadLeveler Diagnosis and Messages Guide, GA22-7882*:

- | | |
|-----------------|---|
| 2512-625 | An <i>adapter_name</i> adapter was specified for adapter <i>adapter_stanza_name</i> , but a <i>port_number</i> was not found. |
| 2512-626 | An <i>adapter_name</i> adapter was specified for adapter <i>adapter_stanza_name</i> , but a <i>logical_id</i> was not found. |
| 2512-627 | An <i>adapter_name</i> adapter was specified for adapter <i>adapter_stanza_name</i> , but a <i>network_id</i> was not found. |
| 2512-628 | An <i>ib</i> adapter was specified for adapter <i>adapter_stanza_name</i> , but a <i>network_id</i> was not valid was found. That is, a <i>network_id</i> keyword that is not valid is being associated with the specified adapter stanza. A valid <i>network_id</i> for the InfiniBand adapter is converted from a <i>GID</i> of the InfiniBand adapter. |



Part 4

Appendixes



InfiniBand security

In this appendix, we briefly discuss security related topics in IBM eServer pSeries Clusters using an InfiniBand interconnect.

IB Protocol layer

Like any other networking technology, InfiniBand adds additional security issues to computer systems. The risks could come from local peers or remote networked computers. They include spoofing, tampering, information disclosure, denial of service, and elevation of privilege.

The design of InfiniBand includes some security features. InfiniBand implements a very basic security infrastructure by using two types of keys, Q_Keys and P_Keys for communication, and L_Keys and R_Keys for remote memory access.

However, there are no mechanisms for encryption or authentication built into InfiniBand. Higher levels of security must be implemented by higher protocol layers or applications using IB.

Partitions

A partition is a set of nodes that can talk to each other. A node can be member of several partitions. InfiniBand partitions can be compared to Ethernet VLANs or Fibre Channel zones. They help enforce security rules.

A partition is defined by its unique P_key. The P_Key is a 15-bit number. Bit number 16 indicates the type of group membership. The Subnet Manager supplies the nodes with its respective P_Keys. The device driver will drop all packets with invalid P_Keys. The P_Key 0xffff is always valid and cannot be altered (default partition).

Q_Keys

When setting up communication, the InfiniBand endpoints exchange Q_Keys. They must always use these Q_Keys in subsequent communications. Multicast packets use the Q_Key associated with the multicast group.

Q_Keys are used to implement access control for reliable and unreliable IB datagram services. Raw datagram services do not use Q_Keys.

Q_Keys with the most significant bit set are called *controlled* Q_Keys. A Host Channel Adapter (HCA) does not accept a controlled Q_Key from an arbitrary consumer. The device driver of the operating system maintains control by restricting the use of controlled Q_Key to privileged consumers.

RDMA

Remote Direct Memory Access (RDMA) allows access to memory in the address space of a user process on a remote node. Data transfer is carried out by the HCA without the intervention of the operating system.

Memory regions that are source or destination of RDMA operations must be registered with the HCA. Registration of memory regions generates keys (L_Keys or R_Keys) that subsequently authenticate access requests and specify access permissions. L_Keys are required for local access and R_Keys are required for remote access. R_Keys must be passed to the remote endpoint and subsequently sent back to the requesting node. Access to memory addresses outside registered regions or bad keys is denied on both ends by the HCA.

Subnet manager

The subnet manager is responsible for configuration and accessibility of the local InfiniBand subnet. It might be susceptible to tampering attacks. Therefore it should be operated in a secure manner like every active networking component.

IBM InfiniBand GX-bus adapters can trigger a serviceable event in case someone has been trying to access the subnet manager without proper authorization. The reference number for this event is CB102800, and the event name is *authenticationFailure*.

If a service action is the likely cause for this, close the event. Otherwise, perform a security audit to determine where this security breach is occurring. Some potential access points are the switch chassis serial port or the service network's Ethernet network.

IP layer

From an IP perspective, InfiniBand is just another network transport like Ethernet. Therefore, security for IPoIB is implemented the same way as for conventional IP over Ethernet by means like firewalls, certificates, and encryption.



B

Cluster Ready Hardware Server

In this appendix, we provide a brief overview of Cluster Ready Hardware Server (CRHS), its installation, configuration, and integration of InfiniBand switches with CRHS.

Cluster Ready Hardware Server basics

The term Cluster Ready Hardware Server (CRHS) refers to a set of software that enhances the ability to control POWER5 servers and HMCs. CRHS enables access to multiple Hardware Management Consoles (HMCs) and other functions that simplify communication to the CRHS service network. These functions include:

- ▶ Automated discovery of POWER5 servers and HMCs, database registration initialization, and automatic hardware configuration updates
- ▶ A distributed database for cluster hardware information
- ▶ An updated hardware server daemon
- ▶ Reduced HMC requirements for recovery
- ▶ Ease of movement of POWER5 servers between HMCs
- ▶ Reduced number of Dynamic Host Configuration Protocol (DHCP) servers for larger clusters

Most of these functions will not be addressed in this appendix. For detailed information about these functions, see the *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344. CRHS is an optional CSM component for environments that do not use an IBM High Performance Switch (HPS). It is recommended for medium to large setups with more than one HMC. It is required if your cluster uses an HPS.

Components

A shared repository is created on each HMC that is included in the cluster. Once this is done the hardware server discovery agent (HSDA) begins polling for the hardware, and it automatically adds to the shared repository all of the hardware found. The hardware server daemon (hdwr_svr) uses the information in the shared database to determine which servers are managed by a specific HMC. This information is listed and modified from the management server. It is used to define, install, and administer cluster nodes. The **systemid** command allows you to set the password for the pSeries service processor. The password is encrypted and stored within the repository.

Requirements

CRHS is supported on HMC-managed pSeries nodes, that is, POWER5 and OpenPower servers. POWER4 servers can still be managed by the CSM management server, but are not included in the enhanced CRHS support that is provided for the POWER5 systems.

CRHS requires openssh and openssl filesets installed on the management server. For details, see *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344.

CRHS components are included in the CSM filesets. See 4.1.3, “Overview of cluster software components” on page 74 for CSM minimum versions and requirements. CRHS has specific requirements for the firmware level of the managed nodes. At the time of this writing, the minimum supported firmware level is GA 5.

CRHS requires that the flexible Service Processor (FSP) of the cluster nodes uses dynamic IP addresses provided by a DHCP server.

Note: The factory default for POWER5 systems FSPs is correct for this setup. However, if the nodes have been used or tested in another setup, we recommend verifying that the FSPs are set to dynamic addresses through ASMI.

Although the location of the DHCP server is not important, it is most convenient to run it on the CSM management server.

CRHS is using port 427 (SLP) to discover hardware devices. This port is disabled on the HMC firewall by default. Port 427 must be enabled for eth0 in the Customize Network Settings of the HMC Configuration GUI.

InfiniBand considerations

The HMC that is running the IBM network manager (see 3.11, “IBM Network Manager (IBM NM)” on page 59) needs access to the InfiniBand switches. Since current versions of the supported InfiniBand switches do not support the service location protocol (SLP), they cannot be discovered automatically. Therefore, they have to be added to the shared database manually. See “CRHS and InfiniBand switches” on page 275.

CRHS and InfiniBand switches

As mentioned before, currently supported IB switches do not support SLP. Therefore they have to be added manually to the distributed database. This is done again with the `mkrhws` command, this time adding an `element_type` `IB_SWITCH`:

```
msib01:/tmp>mkrhws -a Element_Type=IB_SWITCH,Element_IP_A=10.10.10.31
msib01:/tmp>mkrhws -a Element_Type=IB_SWITCH,Element_IP_A=10.10.10.32
```

Now we can verify all of our elements in the distributed database (see Example B-1). We see two IB switches, and four CECs. At this time, the Element_FrameRH of our CECs is non-zero. The CECs and their LPARs can be managed by our HMC.

Example: B-1 Hardware elements in the CRHS distributed database

```
msib01@/>>lsrhws -e
-----
Element_Type      = "IB_SWITCH"
Element_IP_A      = "10.10.10.32"
Element_IP_B      = ""
Element_MTMS      = ""
Element_CEC_Name  = " "
Element_FrameRH   = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
Element_Slot      = ""
Element_Frame_ID  = 0
Element_BPA_MTMS  = ""
-----
Element_Type      = "IB_SWITCH"
Element_IP_A      = "10.10.10.31"
Element_IP_B      = ""
Element_MTMS      = ""
Element_CEC_Name  = " "
Element_FrameRH   = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
Element_Slot      = ""
Element_Frame_ID  = 0
Element_BPA_MTMS  = ""
-----
Element_Type      = "FSP"
Element_IP_A      = "10.10.10.4"
Element_IP_B      = ""
Element_MTMS      = "9131-52A*103920G"
Element_CEC_Name  = " "
Element_FrameRH   = "0x2047 0xffff 0x7a8b698f 0xa133fc01 0x901e046b 0xb80b03a0"
Element_Slot      = "A"
Element_Frame_ID  = 0
Element_BPA_MTMS  = ""
-----
Element_Type      = "FSP"
Element_IP_A      = "10.10.10.3"
Element_IP_B      = ""
Element_MTMS      = "9131-52A*10391DG"
Element_CEC_Name  = " "
Element_FrameRH   = "0x2047 0xffff 0x7a8b698f 0xa133fc01 0x901e046b 0xb80b03a0"
```

```

Element_Slot      = "A"
Element_Frame_ID = 0
Element_BPA_MTMS = ""
-----
Element_Type      = "FSP"
Element_IP_A      = "10.10.10.2"
Element_IP_B      = ""
Element_MTMS      = "9131-52A*10391EG"
Element_CEC_Name  = " "
Element_FrameRH   = "0x2047 0xffff 0x7a8b698f 0xa133fc01 0x901e046b 0xb80b03a0"
Element_Slot      = "A"
Element_Frame_ID = 0
Element_BPA_MTMS = ""
-----
Element_Type      = "FSP"
Element_IP_A      = "10.10.10.1"
Element_IP_B      = ""
Element_MTMS      = "9131-52A*10391FG"
Element_CEC_Name  = " "
Element_FrameRH   = "0x2047 0xffff 0x7a8b698f 0xa133fc01 0x901e046b 0xb80b03a0"
Element_Slot      = "A"
Element_Frame_ID = 0
Element_BPA_MTMS = ""

```

Now that the HMC knows about the InfiniBand switches, we can activate the IBM Network Manager on our HMC. See 3.11, “IBM Network Manager (IBM NM)” on page 59 for details on how to configure and manage the IBM Network Manager.



Function cross table: Linux for AIX sysadmins

In this appendix, we introduce Linux to AIX system administrators who co-manage AIX and Linux systems in their environments. This is a high level comparison of common administrative tasks and locations of system-specific files that are somewhat common to both Linux and AIX.

Note that this appendix does not provide an in-depth Linux comparison for AIX system administrators; its purpose is to help simplify the life of UNIX® administrators managing servers in mixed environments. You can also use ppc utilities compiled by IBM for Linux to run some AIX commands on Linux.

In this appendix, we compare:

- ▶ Major features
- ▶ Common system files
- ▶ Task-specific command comparison

Major features

Table C-1 lists side-by-side comparisons of major features of AIX and Linux.

Table C-1 Major features

Feature	AIX	Linux
Logical Volume Manager	LVM (can only increase the file system size on the fly)	LVM (can increase and also decrease file systems on the fly)
JFS	JFS and JFS2	Reiserfs (SUSE)
Admin interface	SMIT, Smitty, and WebSM	Webmin (Openware) YaST2 (SUSE)
Boot options	SMS menus (Press 1 or F1), Service Processor menus (Press 1 @beep)	SMS menus (Press 1 or F1), Open Firmware menus (Press 8 or F8)
Hardware monitoring	Inventory Scout, diagnostics, and error logger	/var/adm/messages and custom scripts
Clustering	HACMP	Heartbeat (SUSE)
Recovering root access	Boot from CD/network to maintenance mode	Boot from CD/network to "Rescue" mode
Network install	Network Install Manager (NIM)	Cluster Systems Management (CSM) AutoYaST (SUSE)
Image backups	mksysb and sysback	Storix (third party)
File system backups	smitty lvm, Tivoli Storage Manager, and sysback	
Default shell	ksh	bash
Run levels	0-1 (Reserved) 2 (Multiuser-default) 3-9 (User preferences) S,s,m,M (Maintenance mode)	0 (Halt) 1 (Single user mode) 2 (Multiuser-no NFS) 3 (Multiuser-with NFS-default) 4 Unused 5 (default with X11, xdm) 6 (Reboot)
Default window manager	CDE	GNOME,KDE

Common system files

Table C-2 lists common system files and the locations of those files in AIX and Linux.

Table C-2 Common system files

File	AIX	Linux
Password file	/etc/passwd	/etc/passwd
Encrypted password file	/etc/security/passwd	/etc/shadow
Error logs	/var/adm/ras/errpt /var/adm/messages	/var/log/messages
Group files	/etc/group /etc/security/group	/etc/group /etc/gshadow
Allow/Deny/remote login	/etc/security/user	/etc/securetty
DNS	/etc/resolv.conf /etc/netsvc.conf	/etc/resolv.conf /etc/nsswitch.conf
Host Name definitions	/etc/hosts	/etc/hosts
Services	/etc/services	/etc/services
Kernel	/usr/lib/boot/unix_64	/boot/vmlinuz
Device files	ODM database at /etc/objrepos	/dev
Inittab	/etc/inittab	/etc/inittab
inetd.conf	/etc/inetd.conf	/etc/inetd.conf
File systems	/etc/filesystems	/etc/fstab
Network definitions	ODM database at /etc/objrepos	/etc/sysconfig/network (SUSE)
NFS exports	/etc/exports	/etc/exports
System environment	/etc/environment	/etc/profile /etc/bash.bashrc
Common User-related	/etc/security/user	/etc/default/useradd
Profile scripts with new id	/etc/.profile	/etc/skel/*.*

Task-specific command comparison

Table C-3 compares task-specific commands for AIX and Linux.

Table C-3 Command comparison

Task	AIX	Linux
Listing physical volumes	lspv	pvdisk
List partitions in a disk	lspv -l <disk>	fdisk -l <disk>
List volume groups	lsvg	vgdisplay
Create volume group	mkvg	vgcreate
Remove volume group	rmvg	vgremove
Add a physical volume to volume groups	extendvg	vgextend
Remove VG definition	exportvg	vgexport
Remove a physical volume from a volume group	reducevg	vgreduce
Import volume groups	importvg	vgimport
Activate volume group	varyonvg	vgchange
List logical volume	lslv	lvdisplay
Create logical volumes	mklv	lvcreate
Grow file systems with LV	chfs	resize_reiserfs and resize2fs
Shrink file system	-	resize_reiserfs and resize2fs
Paging/Swap space	lspv -a	procinfo cat /proc/swaps
OS level	oslevel	uname -a
Run level	who -r	runlevel
Uptime	uptime	uptime
Performance monitoring	vmstat , ps , and sar	vmstat , ps , and sar

Task	AIX	Linux
List installed filesets	<code>lslpp -l</code>	<code>rpm -qa</code>
Which fileset a file is in	<code>which_fileset <file_name></code>	<code>rpm -qf <file_name></code>
Verify installed filesets	<code>lppchk -v</code>	<code>rpm -V <packagename></code>
List files in a fileset/package	-	<code>rpm -ql <package_name></code>
List running kernel modules	<code>genkex</code>	<code>lsmod</code>
Insert module	N/A (dynamic)	<code>insmod</code> and <code>modprobe</code>
Unload modules	N/A (dynamic)	<code>rmmod</code> and <code>modprobe</code>
List memory installed	<code>bootinfo -r</code>	<code>free</code> , <code>procinfo</code> , and <code>cat /proc/meminfo</code>
Create users	<code>mkuser</code>	<code>useradd</code>
Change user details	<code>chuser</code>	<code>usermod</code> and <code>chage</code>
Delete users	<code>rmuser</code>	<code>userdel</code>
Create group	<code>mkgrp</code>	<code>groupadd</code>
Change group details	<code>chgrp</code>	<code>groupmod</code>
Delete group	<code>rmgrp</code>	<code>groupdel</code>
Install software	<code>installp</code> , <code>smitty installp</code> , <code>rpm</code> , and <code>geninstall</code>	<code>rpm -iv</code> , <code>yast -i</code> , and <code>yast2</code>
Update software	<code>smitty update_all</code> , <code>installp</code> , and <code>rpm</code>	<code>rpm -Uv</code> , <code>yast2</code>
Remove software	<code>smitty installp</code> and <code>rpm</code>	<code>rpm -e</code> , <code>yast2</code>
IP configuration	<code>smitty tcpip</code>	Yast2 network (SUSE)
IP alias	<code>ifconfig en0 alias <IP></code>	<code>ifconfig eth0:1<IP></code>
Network interfaces	<code>netstat -ni</code>	<code>ifconfig</code>
Network routes	<code>netstat -nr</code>	<code>netstat -nr</code>
staticroutes	<code>ODM</code> and <code>/etc/staticroutes</code> & <code>/etc/rc.net</code>	<code>/etc/sysconfig/routes</code> (SUSE)
Network options	<code>no -a</code>	<code>sysctl -a</code>

Task	AIX	Linux
Error logs	errpt and alog	syslog , evlog , tail /var/log/messages , and dmesg
Start daemons	startsrc -s (SRC-controlled subsystems)	rc.svc_name start , chkconfig , and /etc/init.d/<svc_name> start
Stop daemons	stopsrc -s	rc.svc_name stop , chkconfig , and /etc/init.d/svc_name stop
Refresh daemons	refresh -p	rc.svc_name restart and /etc/init.d/svc_name restart
Shutdown halt	shutdown -h	shutdown -h
Fast reboot	shutdown -Fr	shutdown -r now
System dump	sysdumpdev -l	N/A
Kernel tuning	vmo (virtual memory - was vmtune) schedo (scheduler tuning - was schedtune) no (network options)	sysctl (for all)
Change kernel	Change symlink in /usr/lib/boot/unix_mp	Change in /etc/yaboot.conf
PAM authentication	/etc/pam.conf	/etc/pam.d/*
Boot image	bosboot	lilo (SUSE)
Change bootlist	bootlist	N/A
System boot messages	alog	dmesg

Installing OFED and eHCA on Linux Kernel 2.6.16, 2.6.17, and 2.6.18

This appendix describes the processes to install OFED and eHCA on Linux kernel 2.6.16, 2.6.17, and 2.6.18, in case you are interested in testing it on your system.

Table D-1 shows the releases dependencies between SLES 9, OFED, or device drivers.

Table D-1 Kernel version and OFED

Kernel version	OFED	Device driver	Comment
SLES 9 SP3	svn_trunk_6454	EHCA2_0058	Strongly recommended, as the device driver is supported by SUSE and IBM
Kernel 2.6.16	OFED-1.1.1	SVNEHCA2_0015	
Kernel 2.6.17	OFED-1.1.1	SVNEHCA2_0015	
Kernel 2.6.18	OFED-1.1.1	SVNEHCA2_0015	

Steps to install OFED and eHCA on Kernel 2.6.16, 2.6.17, and 2.6.18

1. Download OFED-1.1-rc7.tgz.

Open a Web browser and point it to this URL:

<https://openfabrics.org/svn/gen2/branches/1.1/ofed/releases/OFED-1.1-rc7.tgz>

This will open a dialog box, where you will select **Save to Disk**, and specify a destination directory to save the file.

If there is no Web browser available, issue the following command in the shell to download the file:

```
wget
https://openfabrics.org/svn/gen2/branches/1.1/ofed/releases/OFED-1.1-rc7.tgz
```

We recommend visiting the OpenFabrics Wiki page at <https://openib.org/tiki/tiki-index.php>, as it contains some helpful information about installation and development processes.

2. Unpack and verify the downloaded file.

Unpack the downloaded file above by running the following command:

```
tar zxf OFED-1.1-rc7.tgz
```

This will create a directory OFED-1.1-rc7 with all the required files. The Table D-2 below describes some important files and directories.

Table D-2 Files and directories for OFED-1.1.1

File/Directory	Comment
LICENSE	Licensing information of provided software components
README.txt	Important information and installation guide
install.sh	Shell script for installing the provided software components
build.sh	Shell script for building RPMs of software components
docs/	Contains release notes of provided device drivers and so on
RPMS/	Initially empty, but serves as storage for RPMs created by build.sh or install.sh
SRPMS/	Contains source RPMs required to build binaries RPMs in RPMS/
SOURCES/	Contains complete source code of provided software components as TGZ files

3. Install procedure.

In order to perform this procedure, root access rights are required!

Change to directory OFED-1.1-rc7 and run `install.sh`. See Figure D-1.

```
InfiniBand OFED Distribution Software Installation Menu

    1) View OFED Installation Guide
    2) Install OFED Software
    3) Show Installed Software
    4) Configure IPoIB Network Interface and OpenSM Server
    5) Uninstall InfiniBand Software
    6) Build OFED Software RPMs

    Q) Exit

Select Option [1-6]:2
```

Figure D-1 Installing OFED-1.1.1 - Step 1

Enter 2 to select Install OFED Software. See Figure D-2.

```
Select OFED Software

    1) Basic (InfiniBand modules and basic user level
libraries)
    2) HPC (InfiniBand modules and libraries, mpi and
diagnostic tools)
    3) All packages (all of openib, mpi)
    4) Customize

    Q) Exit

Select Option [1-4]:4
```

Figure D-2 Installing OFED-1.1.1 - Step 2

Since eHCA is not selected by default by the menu items 1) Basic and 2) HPC in OFED-1.1, enter 4 in order to customize the software components to be built or installed. See Figure D-3.

```
Select Option [1-4]:4

RPM packages:
Install kernel-ib: [y/N]:y
Kernel level modules:
Install ib_verbs: [y/N]:y
Install ib_mthca: [y/N]:N
Install ib_ipoib: [y/N]:y
Install ib_ehca: [y/N]:y
Install ib_sdp: [y/N]:y
Install ib_srp: [y/N]:N
Install kernel-ib-devel: [y/N]:N
User level libraries/applications:
Install libibverbs: [y/N]:y
Install libibverbs-devel: [y/N]:N
Install libibverbs-utils: [y/N]:y
Install libibcm: [y/N]:N
Install libibcm-devel: [y/N]:N
Install libmthca: [y/N]:N
Install libmthca-devel: [y/N]:N
Install perftest: [y/N]:N
Install mstflint: [y/N]:N
Install libehca: [y/N]:y
Install libehca-devel: [y/N]:N
Install ofed-docs: [y/N]:
```

Figure D-3 Installing OFED-1.1.1 - Step 3

The install script will walk through the list of available software components step by step. Enter y to select the prompted component or n to exclude it appropriately as shown in the example above. At the end of this questionnaire a list of selected software components is listed. See Figure D-4 on page 289.

```
Following is the list of OFED packages that you have chosen
      (some may have been added by the installation program
due to package dependencies):
```

```
ib_ehca
ib_ipoib
ib_mthca
ib_sdp
ib_verbs
kernel-ib
libehca
libibverbs
libibverbs-utils
```

```
WARNING: This installation program will remove any previously
installed IB packages on your machine.
```

```
Do you want to continue? [Y/n]:Y
```

Figure D-4 Installing OFED-1.1.1 - Step 4

Enter n in order to modify the list or y to continue with next step. See Figure D-5.

```
Preparing to build the OFED RPMs:
```

```
Do you want to include IPoIB configuration files (ifcfg-ib*)?
```

```
[Y/n]:Y
```

```
RPM build process requires a temporary directory.
```

```
Please enter the temporary directory [/var/tmp/OFED]:
```

```
Please enter the OFED installation directory [/usr/local/ofed]:
```

```
Checking dependencies. Please wait ...
```

```
Building InfiniBand Software RPMs. Please wait...
```

```
Building openib RPMs. Please wait...
```

Figure D-5 Installing OFED-1.1.1 - Step 5

Enter `y` in order to include a configuration file for IP over InfiniBand, which is recommended as it will configure the IP over InfiniBand network at boot time automatically.

After accepting the default location for the temporary directory and installation directory, the script will unpack the complete source code, and compile and build the kernel modules, user space libraries, and related tools. It then builds the corresponding RPMs and places them under the directory `OFED-1.1/RPMS/<dist_release>`. In order to determine `<dist_release>`, issue the following command for SLES 9:

```
rpm -qf /etc/SUSE-release
output: sles-release-9.82.17
```

Here is an example list of the created RPMs:

- `kernel-ib-1.1-2.6.16.14_6_ppc64.ppc64.rpm`

This RPM contains all the compiled kernel modules as selected:

- `ib_ehca.ko` - eHCA InfiniBand device driver module
- `ib_addr.ko` - OFED's kernel module, address translation
- `ib_cm.ko` - OFED's kernel module, connection manager
- `ib_core.ko` - OFED's kernel module, core layer used by all device drivers
- `ib_ipoib.ko` - OFED's kernel module, IP over InfiniBand
- `ib_mad.ko` - OFED's kernel module, management datagram protocol
- `ib_sa.ko` - OFED's kernel module, subnet agent
- `ib_ucm.ko` - OFED's kernel module, user space support for `ib_cm`
- `ib_umad.ko` - OFED's kernel module, user space support for `ib_mad`
- `ib_uverbs.ko` - OFED's kernel module, user space support for InfiniBand

Note that the list above shows a subset of available modules only!

- `libehca-1.0-0.ppc64.rpm`

This RPM contains eHCA user space shared libraries for 32- and 64-bit.

- `libehca-devel-1.0-0.ppc64.rpm`

This RPM contains eHCA user space archive library for 32- and 64-bit.

- `libibverbs-1.0.4-0.ppc64.rpm`

This RPM contains OFED user space shared libraries for 32- and 64-bit.

- `libibverbs-devel-1.0.4-0.ppc64.rpm`

This RPM contains OFED user space archive library for 32- and 64-bit and InfiniBand user verbs header files.

- `libibverbs-utils-1.0.4-0.ppc64.rpm`

This RPM contains OFED user space tools like `ibv_devices`, `ibv_devinfo`, and so on, and their manual pages.

When the build is complete, the script asks for IP over InfiniBand configuration. See Figure D-6.

```
Do you want to configure IPoIB interfaces [Y/n]?Y

Configuring IPoIB:

The default IPoIB interface configuration is based on DHCP.
Note that a special patch for DHCP is required for supporting IPoIB.
The patch is available under OFED-1.1/docs/dhcp
If you do not have DHCP, you must change this configuration in the
following steps.

ib0 configuration:

    The current IPOIB configuration for ib0 is:

BOOTPROTO='dhcp'
REMOTE_IPADDR=''
STARTMODE='onboot'
WIRELESS='no'

Do you want to change this configuration? [y/N]:
```

Figure D-6 Installing OFED-1.1.1 - Step 6

In order to use DHCP for IP over InfiniBand, a patch needs to be applied to DHCP; for details refer to OFED-1.1/docs/dhcp. Otherwise, enter y here and specify the IP address for each port as required.

4. Verify installation.

All kernel modules will be stored under the standard directory
/lib/modules/<kernel version>/drivers/infiniband:

- core/ib_addr.ko
- core/ib_cm.ko
- core/ib_core.ko
- core/ib_mad.ko
- core/ib_sa.ko
- core/ib_ucm.ko
- core/ib_umad.ko
- core/ib_uverbs.ko
- core/rdma_cm.ko
- core/rdma_ucm.ko

- hw/ehca/ib_ehca.ko
- ulp/ipoib/ib_ipoib.ko

OFED install script creates a so-called init.d script `/etc/init.d/openibd`, which loads all selected kernel modules at boot time. In order to pass optional parameters to a particular module, edit the file `/etc/modprobe.conf` accordingly. Below is an example to tell `ib_ehca` module that only one link is connected to the switch:

```
# add this line in /etc/modprobe.conf before the line alias ib0
ib_ipoib
options ib_ehca nr_ports=1
```

The user space libraries and tools are located in the installation directory specified above; the default is `/usr/local/ofed`. Note that the libraries are stored under the sub-directory `lib` and `lib64` for 32- and 64-bit respectively, while the binaries under the sub-directory `bin`.

After rebooting the partition, all kernel modules should be loaded automatically by `/etc/init.d/openibd` and `lsmod` should show at least the following modules:

```
ib_core, ib_ehca, ib_sa, ib_mad, ib_ipoib, ib_uverbs
```

5. TCP/IP respective IP over InfiniBand configuration

If not configured by the install script above, each available port can be configured with an IP address manually, as shown below:

```
ifconfig ib0 192.168.178.120
```

As with Ethernet devices, the InfiniBand ports are enumerated as `ib0`, `ib1`, and so on by default. That means:

```
ifconfig ib1 192.168.178.121
```

will configure the second port/HCA with a new IP address.

6. Simple functional tests

In order to perform the following tests, two HCAs are required!

Supposed that the involved HCAs have been configured for TCP/IP so that every well known TCP/IP tool can be utilized. For example:

```
ping -f -l 20 192.168.178.121
```

which performs a flood ping that preloads 20 packets.

Another example is `scp`:

```
scp -R /usr/src/linux 192.168.178.121:/tmp
```

In addition to the standard TCP/IP tools, OFED provides some useful user space applications. For details, refer to 8.1.2, “Monitoring tools for SLES 9” on page 254.

Abbreviations and acronyms

ACL	Access Control List	DMAPI	Data Management API
AIX	Advanced Executive Interactive	DNS	Dynamic Name Service
API	Application Programming Interface	DVD	Digital Video Disk
ARP	Address Resolution Protocol	DWDM	Dense Wave Division Multiplexing
ASM	Advanced System Management	FC	Fibre Channel
ASMI	ASM Interface	FRU	Field Replaceable Unit
BOS	Base Operating System	FSP	Flexible Service Processor
BPA	Bulk Power Assembly	FTP	File Transfer Protocol
BPC	Bulk Power Controller	GFW	Global Firmware
BTH	Base Transport Header	GID	Global IDentifier
CEC	Central Electronics Complex	GPFS	General Parallel File System
CFM	Configuration File Manager	GPL	General Public License
CoD	Capacity on Demand	GRH	Global Route Header
CQE	Completion Queue Entry	GS	Group Services (RSCT)
CRC	Cyclical Redundant Checksum	GSI	General Service Interface
CRHS	Cluster Ready Hardware Server	GUI	Graphical User Interface
CSM	Cluster Systems Management	GUID	Globally Unique IDentifier
CuOD	Capacity upgrade On Demand	HACMP	High Availability Cluster Multi-Processing
CWDM	Coarse Wave Division Multiplexing	HCA	Host Channel Adapter
DAPL	Direct Access Programming Library	HMC	Hardware Management Console
DDR	Double Data Rate	HPC	High Performance Computing
DHCP	Dynamic Host Configuration Protocol	HPS	High Performance Switch
DLPAR	Dynamic LPAR	IB	InfiniBand
DMA	Direct Memory Access	IBA	InfiniBand Adapter
		IBM	International Business Machines Corporation
		IBNM	InfiniBand Network Manager
		IBTA	InfiniBand Trade Association
		ICRC	Invariant Cyclical Redundant Checksum

IP	Internet Protocol	QoS	Quality of Service
IPoIB	Internet Protocol over InfiniBand	QP	Queue Pair
ITSO	International Technical Support Organization	QPN	Queue Pair Number
LAN	Local Area Network	RAC	Real Application Cluster
LAPI	Low-level Application Programming Interface	RAM	Random Access Memory
LIC	Licensed Internal Code	RAS	Reliability, Availability, and Serviceability
LID	Logical Identifier	RC	Return Code
LL	LoadLeveler	RDMA	Remote Direct Memory Access
LMC	LID Mask Count	RFC	Request For Comment
LPAR	Logical PARTition	RM	Resource Manager
LRH	Local Route Header	RMC	Resource Monitoring and Control
LUN	Logical Unit Number (SCSI/SAN)	RPD	RSCT Peer Domain
MAC	Media Access Control	RPM	RPM Package Manager
MPI	Message Passing Interface	RQ	Receive Queue
MSB	Most Significant Bit	RSCT	Reliable Scalable Clustering Technology
MTU	Maximum Transfer Unit	SA	Subnet Administration
NFS	Network File System	SAN	Storage Area Network
NIM	Network Installation Manager	SCSI	Small Computer System Interconnect
NM	Network Manager	SDP	Sockets Direct Protocol
NSD	Network Shared Disk	SDR	Single Data Rate
NTP	Network Time Protocol	SFP	Service Focal Point
PCI	Peripheral Component Interconnect	SL	Service Level
PE	Parallel Environment	SLES	SUSE Linux Enterprise Server
PID	Process IDentifier	SLP	Service Location Protocol
PMR	Problem Management Record	SM	Subnet Manager
POE	Parallel Operating Environment	SMA	Subnet Management Agent
PSSP	Parallel Systems Support Program	SMIT	System Management Interface Tool
PTF	Program Temporary Fix	SMP	Symmetric Multi-Processing
PXE	Pre-boot Execution Environment	SMS	Systems Management Services
QDR	Quadruple Data Rate	SMT	Simultaneous Multi-Threading

SP	System Parallel
SPOT	Shared Product Object Tree
SQ	Send Queue
SRP	SCSI RDMA Protocol
TCA	Target Channel Adapter
TCP	Transmission Control Protocol
TFTP	Trivial Transfer File Protocol
TPS	Transactions Per Second
TWS	Tivoli Workload Scheduler
UDP	Universal Datagram Protocol
UID	User IDentifier
URL	Universal Resource Locator
VCRC	Variant Cyclical Redundancy Checksum
VFS	Virtual File System
VL	Virtual Lane
VLAN	Virtual local area network
WAN	Wide Area Network
WQE	Work Queue Entry
WQP	Work Queue Pair
XML	eXtended Markup Language

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 publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 300. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Configuration and Tuning GPFS for Digital Media Environments*, SG24-6700
- ▶ *Introduction to InfiniBand*, REDP-4095
- ▶ *NIM from A to Z in AIX 5L*, SG24-7296

Other publications

These publications are also relevant as further information sources:

- ▶ *AIX 5L Version 5.3 Installation and Migration*, SC23-4389
- ▶ *AIX 5L Version 5.3 Networks and Communication Management*, SC23-5203
- ▶ *CSM for AIX 5L and Linux Administration Guide*, SA23-1343
- ▶ *CSM for AIX 5L and Linux Command and Technical Reference*, SA23-1345.
- ▶ *RFC 4391 Transmission of IP over InfiniBand (IPoIB)*
- ▶ *RFC 4392 IP over InfiniBand (IPoIB) Architecture*
- ▶ *RSCT: Administration Guide*, SA22-7889
- ▶ *RSCT for AIX 5L: Technical Reference*, SA22-7890
- ▶ *RSCT: Messages*, GA22-7891
- ▶ *RSCT for Linux Technical Reference*, SA22-7893
- ▶ *GPFS V3.1 Administration and Programming Reference*, SA23-2221-00
- ▶ *GPFS V3.1 Advanced Administration Guide*, SC23-5182-00
- ▶ *GPFS V3.1 Concepts, Planning, and Installation Guide*, GA76-0413-00
- ▶ *IBM Cluster Systems Management for AIX 5L and Linux V1.5 Planning and Installation Guide*, SA23-1344

- ▶ *IBM Tivoli Workload Scheduler LoadLeveler Diagnosis and Messages Guide, GA22-7882*
- ▶ *Cisco SFS 7000 Series Product Family Command Reference Guide, Release 2.5.0 (CISCO Customer Order Number: Text Part Number: OL-9163-01)*

Online resources

These Web sites are also relevant as further information sources:

- ▶ Cisco
 - http://www.cisco.com/application/pdf/en/us/guest/products/ps6758/c1031/cdccont_0900aecd803688d9.pdf
- ▶ IBM
 - Cisco Element Manager documentation
 - <http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphau/tpspnelemgrug.pdf>
 - Cisco Chassis Manager documentation
 - <http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphau/tpspnchasmgrug.pdf>
 - Clustering Systems using InfiniBand Networks documentation
 - <http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphau/infinbdpdf.pdf>
 - InfiniBand readme file
 - <http://www14.software.ibm.com/webapp/set2/sass/f/networkmanager/home.html>
- ▶ IBTA
 - <http://www.infinibandta.org>

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Hints and Tips, draft publications and additional materials, as well as order hardcopy IBM Redbooks publications or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols

/csminstall 174
/etc/init.d 211
/etc/sysconfig/network/ifcfg-ib0 181
/etc/sysconfig/network/ifcfg-ib1 181
/etc/sysctl.conf 180
/usr/local/lib/infiniband/libehca.so 226
/usr/local/libehca/etc/libehca.conf 227
/var/ct/cfg/ct_has.thl 78
/var/ct/cfg/ctrmc.acls 78
/var/log/messages 206

Numerics

64-bit open source applications 32

A

ACL 137
adapter sharing 27–28
Adapter Stanza 164
adapter, 12x 167
Address Handle 87
AIX xi–xiii, 4, 30, 32, 36, 39, 71, 73–76, 156, 158, 160, 169, 206–208, 231, 233–234, 237, 243, 274, 279–282
API 21, 26, 80, 87, 143, 159, 163
ARP 88, 91, 123, 126, 208, 221, 252
ASM 207
ASMI 207, 275
authentication 78
authenticationFailure 271
authorization 78

B

Bandwidth Out of the Box 14
Base Transport Header 20
BOS 114
BPC 59
BTH 20
Bulk Power Controller 59
bus 12

C

cables 16, 53
 octopus 55
cables 12x 56
cables feature codes 53
cables, supported 54
CEC 14, 40, 58, 109, 117, 207
CFM 112–113, 117, 121, 193
channel adapter 23
channel based architecture 7
channel multiplexing 20
Chipkill™ memory 32
Cluster Ready Hardware Server 273
Cluster Systems Manager 75
Clustering techniques 11
compilers 188
Configuration File Manager 112
Configuration resource manager 76
connector 12x 55
connectors 16
considerations 30
controlled Q_Keys 270
copycdfs 177
CPU 14
CQ verbs 87
CQE 26
CRC 19, 22
Credit based flow control 19
CRHS 37, 39, 59–60, 100, 156–157, 166, 168, 171, 273–275
CSM 32, 39, 57–58, 75–77, 79, 144, 154–156, 166, 230–236, 254, 274–275, 280
 Management server 57, 59
CSM Management server 95
csm2nimgrps 236
csm2nimnodes 113
csmconfig 100, 174
csmsetupnim 113
csmsetupyast 177
csmstat 122, 178
ctcasd 79
cthags 79
cthats 79
CUoD 40

D

DAPL 26
Data Integrity 19
Data packets 18
DB2 4
DDR 9, 15–16, 245
Dedicated HCA 27
default partition 270
definenode 105, 176
DGID 123
DHCP 37, 39, 58–59, 95, 166, 168, 170–171, 274–275, 291
DHCP server 168
Direct Access Programming Language (DAPL) 26
Direct Access Programming Library 26
DLID 123
DLPAR 37
DMA 15
DMAPI 138
dmesg 209–211
DNS 231, 281
DVD 67–68, 206
DWDM 11
dynamic adapter configuration 163
Dynamic logical partitioning 32

E

eHCA device driver 150, 211
eHCA user space library 154
Error data block 217
Ethernet 11

F

fabric 12
FC 40, 42–43, 45, 132, 135, 157
Fibre Channel 11
FibreChannel zones 270
First Failure Data Capture 32
flow control 15, 17, 19
four wire serial differential connection 16
Frame 59
Frames 165
FRU 52
FSP 38, 58–59, 207, 275
FTP 205, 207

G

General Parallel File System 84, 157
General Services Interface 26
getadapters 82, 110, 118, 121
GFW 35
GID 65, 88, 90, 123, 167–168, 171, 182, 204, 208, 223, 229, 256, 266
Global firmware 35
Global ID 65
Global ID prefix 65
global identifier 90
Globally Unique Identifier 23
GPFS 4, 84–85, 131, 144, 154–155, 157, 190
 disk descriptor file 186
 node descriptor file 185
GPFS portability layer 184
GPL 184, 212, 214
GRH 19
GS 159
GSI 26, 88, 93, 215
GUI 36, 47–48, 66, 178, 206–207, 275
GUID 19, 23, 27, 41–42, 47–48, 125, 127–128, 142, 167, 182, 208, 225, 251, 255
GUID index 42
GX adapter 4, 26
GX adapters 40
GX bus 40

H

HACMP xiii, 76, 82, 280
hardware control 58
Hardware Management Console 37, 59, 94
hardware server daemon 274
hardware server discovery agent 274
HCA 8–10, 22, 40–42, 46–47, 86–88, 92, 142–144, 149, 206, 208, 222, 242, 256–257, 259, 270–271, 292
HCA installation 46
HCA usage modes 47
HCAD 92, 250, 253
hdwr_svr 274
High Availability Cluster Multi-Processing 76
High Performance Computing 8
High Performance Switch 274
High utilization 27
HMC 27, 35–37, 94–96, 104, 142, 156–157, 166, 206–208, 232, 244, 274–275
HMC BIOS 35

Host Channel Adapter 167
host channel adapters 41
Host Channel Adapters (HCA) 23
HPC xi, xiii, 1, 4, 7–8, 10, 21, 28, 31, 65, 132,
154–155, 157–158, 206, 225, 264, 266, 287
HPC cluster 155
HPS 159, 274
hscroot 207
HSDA 274

I

IB 8, 15, 18, 21, 30, 40, 42, 71, 74, 82–83, 86,
141–144, 203–204, 208–209, 246–248, 251–252,
257, 259, 270, 275, 289
IB communication stack 89
IB drivers 146
IBA 86, 89
IBM xi–xiv, 1, 3–5, 9, 14, 16, 18, 30, 32, 36, 71,
73–74, 76, 141–142, 144–145, 203–206, 243–246,
253, 262, 271, 274–275, 279, 285
IBM GX adapter 10
IBM Network Manager 36, 166
IBM.ConfigRM 79
IBM.CSMAgentRM 77
IBM.DMSRM 77
IBM.ManagedNode 98
IBNM 64, 68
ibstat 94, 124
IBTA 9, 23, 26, 28, 143
ibv_devices 182
ibv_devinfo 182
ICM 89
ICRC 19
ifconfig 122
InfiniBand 7, 14, 74
InfiniBand Connection Manager 89
InfiniBand device driver
 Linux installation 179
InfiniBand device driver, Linux 156
InfiniBand gateway 5
InfiniBand Host Card Adapter (HCA) 10
InfiniBand Layers 16
install server 75
installms 174
installnodes 178
instfix 81
Invariant CRC 19
IP xi, 5, 8, 10, 15, 20, 37, 39, 58, 75, 80, 85–86,

143, 147, 149, 158, 221, 223, 231, 237, 242, 252,
258, 271, 275, 283, 290–292
IP over InfiniBand 180
IPoB xi, 75, 85–86, 89–90, 132–133, 158, 170,
180, 221, 237, 271, 287, 289, 291
ITSO xi, xiii

J

job preemption 160

K

kernel ring buffer 211

L

L_Keys 271
LAN 5, 10–11, 32, 38, 84
LAPI 4, 154, 157–160
libsfs 144
LID 18–19, 24, 65, 90, 125, 229, 251, 258, 260
LID assignments 25
limitations 30
link layer 17
Linux 4, 30, 32, 85
LL 17, 144, 164, 193–194
llctl 196
llextrPD 194, 230
llstatus 157, 164, 196
llsummary 157
LMC 65, 171
LMC value 167
LoadL_admin file 196
LoadLeveler 158, 162, 229
LoadLeveler configuration 193
LoadLeveler installation 193
LoadLever 30
LoadLever admin file 229
Local Area Networks (LAN) 11
local identifier 90
location code 46
location codes 47
Logical HCA 167
Logical Host Channel Adapter 63
Logical Identifier 24
Logical Switch 63
low latency 8
Low utilization 27
low-level application programming interface 158

LPAR 27, 37, 42, 48, 57, 84, 86, 96, 105, 167, 208
LPARs 28
lpp_source 103
LRH 18–19, 90
lsattr 82, 128
lsdev 94
lshmc 36
lshwinfo 104, 175
lshwres 41
lsmcode 36
lsnim 104, 113–114
lsnode 105, 115, 176
lsrsrc 83, 98

M

MAC 41, 111
Managed devices 75
Managed nodes 75
Management Domain 77
Management infrastructure 25
management node 77
Management packets 18
Management Server 171
Management server 75
mechanical specifications 16
Medium utilization 27
Memory Region 87
Memory regions 213
message queuing 15
micro partitioning 31
micro partitions 49
mmcrcluster 134
mmcrfs 136, 188
mmcrnsd 135–136
mmlscluster 134
mmlsfs 137
mmlsnsd 136
mmmount 188
mmstartup 135, 187
modprobe 213, 216
monitorinstall 178
MPI 4, 28, 89, 144, 158, 161, 264–266
MSB 226
MTU 20, 181, 208, 242, 260
Multicast 15
multicast packets 23
multithreading 31

N

NAM 160
netstat 124
network availability matrix (NAM) 159
Network Fabric 23
Network Layer 19
Network Manager 59, 66, 68
Network shared disk 84
network table (NTBL) 159
network tables 160
NFS 89, 131, 139, 172, 179, 181, 183, 190, 250, 280–281
NIM 75, 98, 100–101, 234, 236, 280
nim 104, 114
NIM customization script 117
NIM server 100
nim_master_setup 101
NM 30, 36–37, 47, 175
nodegrp 107, 176
NSD 84, 131–132, 186–188
NSD servers 131
NTP 64

O

OFED 143
OFED stack 143
 Application layer 144
 Core layer 143
 Driver layer 143
 Upper level protocol layer 143
 User space API layer 143
Open Fabrics Enterprise Distribution 143
OpenIB 154

P

P_Key 90
P_key 270
packet forwarding 18
packet layout 17
Packets 18
Parallel Environment 161
Partitions 270
password switch 53
Path Reply 91
PCI 10, 13–14, 32, 40, 59, 244–246
PCI bus 11
PCI Express 13
PCIe 13

PE 4, 144–145, 154, 158
PE installation 188
Physical Host Channel Adapter 63
Physical Layer 16
PID 65, 101, 131, 139, 250
ping 121
PMR 203, 207
PNSD 159–160, 228
POE 158, 163, 189
port ID 46
Power Subsystem Microcode 35
Preemption 30
premanaged 115
primary NSD server 131, 135
probemgr 100, 175, 177
Protection Domain 87
Protocol Network Services Daemon 159
PSSP xiii, 82
PTF 36, 184
PXE 98

Q

Q_Keys 270
QDR 9, 16
QoS 10–12, 18
QP 22, 87–88, 90, 211, 217, 219–220
QPN 88, 90, 258, 260–261
Quality of Service 18
Queue communication 213
Queue Pair 25, 87, 94
queue pair 28
Queue pair control word 219
Queue pair error word 219
queue pair number 90
Queue Pair operations 25
Queue pairs 213

R

R_Keys 271
RAC 4
RAM 22, 68, 97, 142
Rapid Service Architecture 52
RAS 7, 10–11, 32, 51
RC 260
rconsole 114, 178
RDMA 5, 8, 12, 15, 21–22, 160, 270–271
Receive Queue 87
receptacles 16

recv_queue_size 181
Redbooks Web site 300
 Contact us xv
redundant service processor 32
Reference Code 207
reliability and serviceability capabilities 31
Reliable and unreliable datagrams 20
Reliable datagrams 20
Reliable Scalable Cluster Technology 76
remote command 183
remote copy 183
Remote Direct Memory Access 5, 22, 270
resolv.conf 234
Resource Monitoring and Control 76
resource monitoring and control 156
RFC 126–127, 129, 252
RM 80
RMC 76–79, 156–157, 163, 191
RMC API 163
Router 24
routers 24
RPD 76, 78, 191, 194, 230
RPM 36, 146–147, 150, 156, 243, 288–289
rpm 148
rpower 108, 176
RQ 87
RQP 123
RSCT 75–76, 78, 81, 144, 154–156, 166, 229–231, 261
RSCT group services 157
RSCT peer domain 76, 157
RSCT topology services 157

S

SA 41
SAN 5, 10–11, 21, 131, 157
scp 132, 183
SCSI 5, 21, 27, 245
SDP 21, 89
SDR 9
secondary NSD server 131, 135
security infrastructure 270
Send Queue 87
send_queue_size 181
service location protocol 275
serviceable event 271
SFP 206
shared adapter 206

- shared bus 13
- shared processor 31, 49
- Shared Product Object Tree 100
- Simultaneous Multi Threading 112
- SL 18
- SLES xi, 71, 75–76, 99, 141–142, 144–145, 209, 233, 237, 242–243, 250, 253–254, 285, 290
- SLES9 distribution server 176
- SLID 123
- SLP 171, 275
- SM 22, 41, 47, 87, 90
- SMA 22, 47, 88
- SMIT 93, 109, 111–112
- SMP 88, 212, 214
- SMS 156, 280
- SMT 112
- Sockets Direct Protocol 21
- software requirements 74
- SP xii
- SPOT 100
- SQ 87
- SQP 123
- SRP 5, 21, 27
- ssh 132, 183
- Storage Area Network 131
- Storage Area Networks (SAN) 11
- Subnet Management 25
- Subnet Management Agent (SMA) 22
- Subnet Manager 18, 25, 41
- Subnet manager 25
- switch 23, 50
- Switch name 167
- switch topology 61
- Switching 18
- System p 3, 28, 31, 38, 58, 141, 154
- System p cluster 4
- systemid 174

T

- Target Channel Adapter 21
- Target Channel Adapters (TCA) 23
- TCA 21, 23
- TCP 22, 85, 89, 98, 126, 149, 252, 292
- TFTP 205
- topas 138
- TPS 130, 139, 250
- trace file 224
- Transport Layer 20

TWS xi, 30, 144–145, 157–158, 163

U

- UID 134, 186
- Unicast 15
- unicast packets 23
- unreliable datagram 90
- Unreliable datagrams 21
- unreliable datagrams 21
- updatenode 121, 179
- Updating the switch software 67
- URL 156–157, 161, 163, 243, 286
- user space (US) protocol 159
- user space interface 161
- user space libraries 154
- user space library 150
- User Space Protocol 225

V

- Variant CRC 19
- VCRC 19
- vector processor 155
- VFS 85
- Virtual Adapter 27
- Virtual I/O Server 32
- Virtual LAN 32
- Virtual Lane 18
- Virtual Lanes 18
- VL 17–19
- VLAN 270

W

- WAN 11
- wget 146
- Wide Area Network (WAN) 11
- window resources 160
- work queue entry WQE 26
- work queue pair (WQP 26
- WQE 26, 125, 127, 220, 251
- WQP 26

X

- XL C/C++ 188
- XL Fortran 188
- XML 177



Implementing InfiniBand on IBM System p

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Implementing InfiniBand on IBM System p



Understanding and exploiting InfiniBand

HPC and commercial solution explored

AIX 5L V5.3 and SLES 9 implementation

This IBM Redbooks publication will illustrate the installation procedures of InfiniBand on the IBM System p5 with Linux and AIX 5L. InfiniBand adapters, switches, and network management software will be described in this publication. The IBM HPC stack will be tested with InfiniBand (Parallel Environment, LoadLeveler, GPFS, ESSL, and Parallel ESSL). Communication protocols such as MPI and LAPI will be tested and observations will be illustrated in this book.

This book is the complete guide on how to implement InfiniBand on the IBM System p5. It is targeted at all IT professionals looking to understand what is behind the InfiniBand technologies, how to deploy it, and what is the IBM solution incorporating this technology.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks