

Linux HPC Cluster Installation

Cluster installation using xCAT -
xCluster Administration Tools

Linux clustering based on IBM
eServer xSeries

Installing Red Hat with
Kickstart and xCAT



Luis Ferreira,
Gregory Kettmann,
Andreas Thomasch,
Eileen Silcocks,
Jacob Chen, Jean-Claude Daunois,
Jens Ihamo, Makoto Harada,
Steve Hill, Walter Bernocchi,
Egan Ford



International Technical Support Organization

Linux HPC Cluster Installation

June 2001

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 221.

First Edition (June 2001)

This edition applies to Red Hat® Linux® Version 6.2 for Intel® Architecture.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

| | |
|---|-------|
| Figures | xi |
| Tables | xiii |
| Preface | xv |
| The team that wrote this redbook | xv |
| Acknowledgements | xviii |
| Special notice | xix |
| IBM Trademarks | xx |
| Comments welcome | xx |
| | |
| Chapter 1. Introduction | 1 |
| 1.1 In the beginning | 2 |
| 1.2 Intended audience | 3 |
| 1.3 Open source | 3 |
| 1.4 Linux | 4 |
| 1.5 Linux clusters | 4 |
| 1.5.1 High Availability (HA) clusters | 5 |
| 1.5.2 High-Performance Computing | 7 |
| | |
| Chapter 2. General cluster architecture | 11 |
| 2.1 Applications | 13 |
| 2.1.1 Parallelism | 13 |
| 2.1.2 Computer architecture | 14 |
| 2.1.3 Software application program interface (API) architecture | 16 |
| 2.1.4 Application architecture | 16 |
| 2.1.5 Creating the application | 17 |
| 2.1.6 Bottlenecks | 17 |
| 2.2 Hardware architecture | 19 |
| 2.2.1 Cluster components | 21 |
| 2.2.2 User | 21 |
| 2.2.3 Control | 22 |
| 2.2.4 Management | 22 |
| 2.2.5 Storage | 23 |
| 2.2.6 Installation | 23 |
| 2.2.7 Compute | 24 |
| 2.3 Network architecture | 25 |
| 2.3.1 How to design the cluster network | 27 |
| 2.3.2 Remote access/Equinox terminal server | 29 |

| | | |
|-------|--|-----------|
| 2.4 | Software architecture | 30 |
| 2.4.1 | Operating system | 31 |
| 2.4.2 | File system | 31 |
| 2.4.3 | Interprocess Communication (IPC) | 32 |
| 2.4.4 | Resource management | 34 |
| | Chapter 3. Components overview | 37 |
| 3.1 | Hardware | 38 |
| 3.1.1 | Head node | 38 |
| 3.1.2 | Compute nodes | 39 |
| 3.1.3 | Management | 39 |
| 3.1.4 | Network | 41 |
| 3.2 | Software | 44 |
| 3.2.1 | Operating system | 44 |
| 3.2.2 | System management | 48 |
| 3.2.3 | Development | 50 |
| 3.2.4 | Resource management | 53 |
| | Chapter 4. Solution guide | 57 |
| 4.1 | General considerations | 58 |
| 4.2 | Configuration aids | 60 |
| 4.2.1 | Rack configurator | 60 |
| 4.2.2 | PC configurator | 60 |
| 4.2.3 | Other useful resource and tips | 61 |
| 4.3 | Configuration schemes | 63 |
| 4.4 | Cluster questionnaire | 66 |
| 4.5 | Our cluster configuration | 69 |
| | Chapter 5. Hardware preparation | 73 |
| 5.1 | Node hardware installation | 74 |
| 5.2 | Populating the rack and cabling | 77 |
| 5.3 | Cables in our cluster | 82 |
| 5.4 | ASMA card setup | 84 |
| | Chapter 6. Management node installation | 85 |
| 6.1 | Before you start | 86 |
| 6.2 | Operating system installation | 86 |
| 6.2.1 | Installing the Red Hat CD-ROM | 87 |
| 6.2.2 | System configuration | 88 |
| 6.3 | xCAT installation | 92 |
| 6.4 | Additional software installation | 93 |
| 6.4.1 | The Linux kernel | 93 |
| 6.4.2 | h2n | 99 |
| 6.4.3 | PXELinux | 99 |

| | |
|---|------------|
| 6.4.4 atftp | 100 |
| 6.4.5 Equinox setup | 101 |
| 6.4.6 Conserver | 104 |
| 6.4.7 fping | 105 |
| 6.4.8 OpenSSH | 106 |
| 6.5 Reboot and make sure it all works | 108 |
| Chapter 7. Compute node installation. | 109 |
| 7.1 Populate tables | 110 |
| 7.1.1 Site table | 110 |
| 7.1.2 Node list table | 111 |
| 7.1.3 Node resources table | 112 |
| 7.1.4 Node type table | 112 |
| 7.1.5 Node hardware management table | 113 |
| 7.1.6 ASMA table | 113 |
| 7.2 Configure cluster services | 114 |
| 7.2.1 xntpd | 114 |
| 7.2.2 Domain name system (DNS) | 114 |
| 7.2.3 User ID management/Network Information System (NIS) | 115 |
| 7.2.4 Dynamic Host Configuration Protocol (DHCP) | 116 |
| 7.3 Collect MAC addresses | 117 |
| 7.4 Setup rangers (stage3) | 119 |
| 7.5 Install first node | 119 |
| 7.6 Install remaining nodes | 120 |
| 7.7 Post install | 120 |
| Chapter 8. Installation of additional components | 123 |
| 8.1 Install GCC or PGI Compiler | 124 |
| 8.1.1 GNU C compiler (gcc) | 124 |
| 8.1.2 PGI Workstation code | 124 |
| 8.2 Install MPICH to use either TCP/IP or Myrinet GM | 127 |
| 8.2.1 Remove local area multicomputer/message passing interface | 127 |
| 8.2.2 Command mpimaker | 127 |
| 8.2.3 Configure MPICH to use TCP/IP | 127 |
| 8.2.4 Configure MPICH to use Myrinet GM | 130 |
| 8.3 Installing libraries and debugger | 136 |
| Appendix A. xCAT help | 137 |
| rpower - Remote power control | 140 |
| Description | 140 |
| Synopsis | 140 |
| Example | 140 |
| rreset - Remote hardware reset | 141 |
| Description | 141 |

| | |
|------------------------------------|-----|
| Synopsis | 141 |
| Example | 141 |
| rcad - Remote software reset | 142 |
| Description | 142 |
| Synopsis | 142 |
| Example | 142 |
| Files | 142 |
| Diagnostics | 142 |
| See also | 143 |
| rcons - Remote console | 144 |
| Description | 144 |
| Synopsis | 144 |
| Files | 144 |
| Diagnostics | 144 |
| See also | 144 |
| wcons - Remote console | 145 |
| Description | 145 |
| Synopsis | 145 |
| Example | 145 |
| Files | 146 |
| Environment | 146 |
| Diagnostics | 146 |
| See also | 146 |
| rvid - Remote video | 147 |
| Description | 147 |
| Synopsis | 147 |
| Options | 147 |
| Files | 147 |
| Diagnostics | 147 |
| See also | 148 |
| wvid - Remote video | 149 |
| Description | 149 |
| Synopsis | 149 |
| Example | 149 |
| Options | 150 |
| Files | 150 |
| Diagnostics | 150 |
| See also | 150 |
| rvitals - Remote vitals | 151 |
| Description | 151 |
| Synopsis | 151 |
| Example | 151 |
| Options | 151 |

| | |
|--|-----|
| Files | 152 |
| Diagnostics | 152 |
| See also | 152 |
| reventlog - Remote hardware event logs | 153 |
| Description | 153 |
| Synopsis | 153 |
| Examples | 153 |
| Options | 154 |
| Files | 154 |
| Diagnostics | 154 |
| See also | 154 |
| rinv - Remote hardware inventory | 155 |
| Description | 155 |
| Synopsis | 155 |
| Example | 155 |
| Options | 156 |
| Files | 156 |
| Diagnostics | 156 |
| See also | 157 |
| psh - Parallel remote shell | 158 |
| Description | 158 |
| Synopsis | 158 |
| Examples | 158 |
| Options | 158 |
| Files | 158 |
| Diagnostics | 159 |
| See also | 159 |
| pping - Parallel ping | 160 |
| Description | 160 |
| Synopsis | 160 |
| Examples | 160 |
| Files | 160 |
| See also | 160 |
| rinstall - Remote network Linux install | 161 |
| Description | 161 |
| Synopsis | 161 |
| Example | 161 |
| Files | 161 |
| Diagnostics | 161 |
| See also | 162 |
| winstall - Windowed remote network Linux install | 163 |
| Description | 163 |
| Synopsis | 163 |

| | |
|---|------------|
| Example | 163 |
| Files..... | 164 |
| Environment | 164 |
| Diagnostics | 164 |
| See also | 165 |
| Appendix B. xCAT configuration files | 167 |
| site.tab | 169 |
| nodelist.tab | 171 |
| noderes.tab | 172 |
| nodetype.tab | 174 |
| nodehm.tab | 175 |
| asma.tab | 179 |
| mac.tab | 180 |
| passwd.tab | 181 |
| apc.tab | 183 |
| Appendix C. POVRay test | 185 |
| POVRay Installation | 186 |
| Downloads | 186 |
| Install..... | 186 |
| POVRay using TCP/IP and GCC | 187 |
| TEST..... | 187 |
| RESULT | 187 |
| POVRay using Myrinet and PGI | 189 |
| TEST..... | 189 |
| RESULT | 189 |
| Summary | 191 |
| POVRay results..... | 191 |
| Other tests..... | 191 |
| Appendix D. Hardware configuration used in our lab | 193 |
| Hardware Environment for the Lab | 194 |
| Appendix E. Installation experiences | 197 |
| Appendix F. Cluster questionnaire | 205 |
| Establishing expectations questionnaire | 206 |
| Environmental questionnaire | 208 |
| Hardware/configuration questionnaire | 209 |
| Software questionnaire..... | 212 |
| Appendix G. Additional material | 215 |
| Locating the Web material | 215 |

| | |
|--|-----|
| Using the Web material | 215 |
| System requirements for downloading the Web material | 216 |
| How to use the Web material | 216 |
| Related publications | 217 |
| IBM Redbooks | 217 |
| Other resources | 217 |
| Referenced Web sites | 218 |
| How to get IBM Redbooks | 220 |
| IBM Redbooks collections | 220 |
| Special notices | 221 |
| Index | 225 |

Figures

| | | |
|------|---|-----|
| 1-1 | A generic Web cluster | 6 |
| 1-2 | Beowulf logical view | 10 |
| 2-1 | Components of a Beowulf cluster | 12 |
| 2-2 | Beowulf Linux cluster example | 20 |
| 2-3 | Full bisection bandwidth example | 27 |
| 2-4 | Beowulf cluster network scheme | 29 |
| 2-5 | Beowulf software components | 30 |
| 4-1 | Rack configurator tool | 61 |
| 4-2 | PC Configurator tool | 62 |
| 5-1 | x330 as shipped (the serial port connection to be changed is circled) | 74 |
| 5-2 | Serial port jumper before (port A) and after the change (port B) | 75 |
| 5-3 | x330 with Myrinet card installed into short PCI slot | 75 |
| 5-4 | x330 with Myrinet and ASMA card installed. | 76 |
| 5-5 | Node ID labeling on the front | 78 |
| 5-6 | C2T SPN/ASMA cabling | 79 |
| 5-7 | Terminal server cables (left) and “regular” FastEthernet cabling (right). | 80 |
| 5-8 | Rack cabled completely (including Myrinet). | 81 |
| 5-9 | Cables on head node x340 | 82 |
| 5-10 | Cables on compute node x330 | 83 |
| A-1 | wcons output | 145 |
| A-2 | wvid output | 149 |
| A-3 | winstall output | 164 |
| E-1 | 256 node cluster | 199 |
| E-2 | Fibre cables | 200 |
| E-3 | Ethernet cables | 201 |
| E-4 | ASMA and SPN cables | 202 |
| E-5 | Myrinet cables | 203 |

Tables

| | | |
|------|---|-----|
| 2-1 | Parallel computing store summary | 14 |
| 4-1 | Localhost | 64 |
| 4-2 | eth0 on the management node | 64 |
| 4-3 | eth1 on the management node | 64 |
| 4-4 | myri0 on the compute node | 64 |
| 4-5 | eth2 on the management node (management only network) | 65 |
| 4-6 | Our cluster configuration | 69 |
| 4-7 | Cluster software table | 70 |
| A-1 | xCAT commands | 138 |
| B-1 | xCAT tables | 167 |
| B-2 | Definitions of site.tab parameters | 169 |
| B-3 | Definitions of nodelist.tab parameters | 171 |
| B-4 | Definitions of noderes.tab parameters | 172 |
| B-5 | Definitions of nodetype.tab parameters | 174 |
| B-6 | definition of nodehm.tab parameters | 175 |
| B-7 | Definitions of asma.tab parameters | 179 |
| B-8 | Definitions of mac.tab parameters | 180 |
| B-9 | Definitions of passwd.tab parameters | 181 |
| B-10 | Definitions of apc.tab parameters | 183 |
| F-1 | Expectations questionnaire | 206 |
| F-2 | Environment questionnaire | 208 |
| F-3 | Hardware/ configuration questionnaire | 209 |
| F-4 | Software questionnaire | 212 |

Preface

Few people who use computers have not heard of Linux®. Linux is at the heart of a revolution that is fundamentally changing many aspects of how software is written and delivered. It remains to be seen where this revolution will ultimately lead. Far less nebulous is the Linux penetration into the scientific calculation arena, or more specifically, its use in High-Performance Computing clusters.

This redbook will guide system architects and systems engineers through a basic understanding of cluster technology, terms, and Linux High-Performance Computing (HPC) clusters. We discuss some of the design guidelines used when architecting a solution and on how to install and configure a working Linux HPC cluster.

This redbook documents the building of a Linux HPC cluster on IBM @server xSeries hardware. Using this approach, and building a type of Beowulf cluster, we can achieve dramatic performance at a fraction of the price of previous solutions.

This redbook will also teach you the concepts and principals needed to build your own HPC cluster. The installation process is streamlined and simplified by the extensive use of scripts. Management tools are provided to easily manage a large number of compute nodes that remotely use the built-in features of Linux and the advanced management capabilities of the IBM @server xSeries Service Processor Network.

The team that wrote this redbook

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

Luis Ferreira (also known as “Luix”) is a Software Engineer at IBM Corp - International Technical Support Organization, Austin Center working in Linux and AIX projects. He has 18 years of experience in UNIX-based operating systems, and holds a MSc. Degree in System Engineering from Univerdade Federal do Rio de Janeiro, Brazil. Before joining the ITSO, Luis worked at Tivoli Systems as a Certified Tivoli Consultant, at IBM Brasil as a Certified IT Specialist, and at Cobra Computadores as kernel developer and software designer. His e-mail address is luix@us.ibm.com.

Gregory Kettmann (also known as “Greg”) is an IT Architect working for IBM Global Services, ITS, as a Subject Matter Expert in Linux Clusters. His group, the Linux Global Solutions Group, is responsible for establishing Linux strategies and policies worldwide. Greg has 21 years of experience with IBM, the last 10 of which have been as a workstation and networking specialist. He has been involved with Linux and Linux High-Performance Computing clusters for the past several years. He earned his ET degree from NTS and his EE from CREI. His e-mail address is gkettman@us.ibm.com.

Jens Ihamo is IT Specialist at the IBM Techline in Greenock, Scotland. He has more than three years of experience in xSeries and Netfinity servers from Post Sales Support and Technical Pre-Sales environments, He has been involved for four years with Linux and has worked for IBM for more than three years. His areas of expertise include networking, clustering, Fibre Channel technologies and all things Linux. He is a Red Hat Certified Engineer (RHCE), Professional Server expert (PSS) and Microsoft Certified Systems Engineer (MCSE). His e-mail address is jihamo@uk.ibm.com.

Andreas Thomasch is a Linux specialist and consultant at the EMEA Linux Center of Competence in Boeblingen, Germany. He has eight years of experience in the Linux field and still remembers when a Linux distribution was delivered on a dozen floppy disks. He has studied and worked at IBM for five years. His areas of expertise include UNIX server support and consulting (Linux/Intel, Linux/390, AIX, and Solaris). His current job responsibility focuses on technical sales support for xSeries Linux clustering solutions. He holds a degree in Computing Science from Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Germany. His e-mail address is a.thomasch@de.ibm.com.

Steve Hill is a High Performance Computing specialist working in Hursley, England. He has six years of experience with Linux and remembers waiting forty minutes for the 1.3 kernel to compile on a 486. His current responsibilities involve providing services as a consultant for IBM SP2s and Linux clusters. He has built and installed a number of Linux clusters, including some for trade shows. His email address is s_hill@uk.ibm.com.

Walter Bernocchi is an IT Specialist at the xSeries Pre-Sale Technical Support team in Italy. He has been working for 12 years at IBM. Since March 2000, he has been fully dedicated to design and support solutions based on Linux. His areas of expertise include C/C++, Java, SunOS, Solaris, and AIX. He is a Red Hat Certified Engineer (RHCE). His e-mail address is walter_bernocchi@it.ibm.com.

Jean Claude Daunois is a System Engineer in France. He has 11 years of experience within IBM. He works as an Advisory IT specialist supporting Netfinity technologies and network operating systems implementations on the Netfinity Pre-Sales Technical Support team. His area of expertise includes Linux, Santa-Cruz Operating System (SCO), Windows NT®, Netfinity hardware, and software engineering. His e-mail address is daunois@fr.ibm.com.

Eileen Silcocks is an IBM Technical Trainer in Greenock, Scotland. She has trained people on the latest desktop technologies, and currently specializes in Linux training. She researches and develops training materials that are used in training courses and workshops. She holds an Honors Music degree and a Masters degree in Information Technologies. Her e-mail address is esilcock@uk.ibm.com.

Jacob Chen is an IT Specialist of IBM Taiwan. He joined IBM Taiwan in 1992. He has more than 20 years of experience in the field of high performance computing. His areas of expertise include FORTRAN programming, performance tuning for POWER architecture, and MPI parallel programming and tuning for the RS/6000 SP machines. He has been assigned to support the IBM 3090 Vector Facility and MPI parallel programming in IBM Taiwan since 1992. He graduated from Taipei Institute of Technology, Civil Engineering in 1974. His e-mail address is jacobche@tw.ibm.com.

Makoto Harada is an IT Engineer at PS Server, ATS IBM Japan. He has been involved with UNIX-based operating systems (such as SunOS, Solaris, FreeBSD, UNIXWare, and Linux) for 12 years. He is a Red Hat Certified Engineer (RHCE), Professional Server expert (PSS), Microsoft Certified Professional (MCP), and Red Hat Certified Examiner (RHCE). His e-mail address is makohara@jp.ibm.com.



This picture shows the Blue Tuxedo Team. On the top, left to right, are Greg, Jean, Walter, Jens and Jacob. On the bottom, left to right, are Makoto, Andreas, Luis, Eileen and Steve.

Acknowledgements

This redbook was produced using the management mechanism called xCluster Administration Tools (also known as xCAT), which was designed and written by Egan Ford.

The team would like to express **special thanks** to Egan Ford for his major contribution to this project.

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

International Technical Support Organization, Austin Center

Caroline Cooper, Lupe Brown, Gwen Monroe, Matthew Parente, Wade Wallace, Chris Blatchley

International Technical Support Organization, Poughkeepsie Center

Dino Quintero, Fred Borchers, Michael Schwartz, Thomas Marone, Yann Guerin

IBM North America

Sharon Dobbs, Joseph Banas, Jay Urbanski, Alan Fishman, Bruce Potter, Kevin Gildea, Edward Jedlicka, Scott Denham, Mark Atkinson, Mike Galicki, Tim Strickland, Michele Perry

IBM Scotland

Gabriel Sallah, Adam Hammond, Jim Mckay, Paul McWatt

IBM Germany

Andreas Hermelink, Michael Weisbach and Jan-Rainer Lahmann

IBM Italy

Mauro Gatti

IBM France

Eric Monjoin and Stephan Hilby

Extreme Networks

Tony Riccio

Myricom, Inc.

Nan Boden

Equinox Systems Inc.

Sonny Edmundson

Special Thanks to Joanne Luedtke (International Technical Support Organization Manager, Austin Center) for her effort and support for this project.



Also, thanks to Linus Torvalds for rescuing the dream.

Special notice

This publication is intended to help system architect and system engineers to understand, design, and install an IBM *@server* xSeries Linux High-Performance Computing Cluster. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM *@server* and Linux Red Hat. See the PUBLICATIONS section of the IBM Programming Announcement for more information about what publications are considered to be product documentation.

IBM Trademarks

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

| | |
|---|---|
| 3090 | AIX |
| AS/400 | AT |
| CT | Current |
| DB2 | e (logo)  |
| IBM ® | LoadLeveler |
| Netfinity | NetView |
| pSeries | Redbooks |
| Redbooks Logo  | RS/6000 |
| ServeRAID | SP |
| SP2 | Tivoli |
| TME | Wave |
| WebSphere | xSeries |
| XT | |

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review redbook form found at
ibm.com/redbooks
- ▶ Send your comments in an Internet note to
redbook@us.ibm.com
- ▶ Mail your comments to the address listed on Page ii



Introduction

This chapter provides a general overview of Linux clusters and High-Performance Computing. It also introduces the concepts which will be employed throughout this book.

In this chapter, we briefly discuss the following topics:

- ▶ Who should read the book
- ▶ Open source and Linux
- ▶ Types of clusters

1.1 In the beginning

Over the years there have been dramatic increases in computing power and capabilities, but none so dramatic as recently. Early mathematical computations were facilitated by lines drawn in the sand. This eventually led to the abacus, the first mechanical device for assisting with mathematics. Much later came punch cards, a mechanical method to assist with tabulation. Ultimately, this led to ever more complex machines, mechanical and electronic, for computation.

Today, a small handheld calculator has more computing power than that available to the Apollo missions that went to the moon. Early computers used small toroids to store hundreds or thousands of bits of information in an area the size of a broom closet. Modern computers use silicon to store billions of bits of information in a space not much larger than a postage stamp.

But even as computers become more capable, certain constraints still arise. Early computers worked with 8 bits, or a byte, to solve problems. Most modern computers work with 32 bits at a time, with many dealing with 64 bits per operation, which is similar to increasing the width of a highway. Another method for increasing performance is to increase the clock speed, which is similar to raising the speed limits. So, modern computers are the equivalent of very wide highways with very fast speed limits.

However, there are limits to the performance benefits that can be achieved by simply increasing the clock speed or bus width. In this redbook, we present an alternative approach to increasing computing power. Instead of using one computer to solve a problem, why not use many computers, in concert, to solve the same problem?

A computer is not just based on hardware. There is also the operating system and the software. There have been noteworthy developments in operating systems that will help us in our quest for more processing power. A fairly recent development is Linux, an operating system with very robust multi-user and multi-tasking capabilities. Linux has the added advantage of being freely available and completely non-proprietary. The Linux source code is openly available, allowing a level of control and modification unavailable in a proprietary environment.

In this redbook, we describe a procedure to build a Linux cluster, generically known as a Beowulf cluster, by using Linux running on an IBM `@server` xSeries machine. Linux clusters use relatively inexpensive hardware and achieve results that rival the largest and most expensive computers available.

The scope of this redbook is to define how to use many computers, in concert, to solve a problem. There are many different approaches to this problem, with as many different solutions. We will explore some of these solutions and describe their relative merits. However, the focus of this redbook will be a set of detailed steps to set up an HPC Linux cluster.

First, we discuss some of the design guidelines used when architecting a High-Performance Computing Solution. This is followed by detailed procedures on how to install and configure a working HPC cluster.

1.2 Intended audience

It is assumed that the reader is knowledgeable in basic Linux Skills, such as installation, configuration and use. If not, the redbook *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5853 should prove helpful.

The reader should find all the information to guide them through a basic understanding of cluster technology and terms in this redbook.

1.3 Open source

Before discussing clusters, it is important to address open source software and Linux, since this is the foundation on which our solution is built.

In 1984 Richard Stallman, then working for MIT, became distressed with the way the software industry was evolving. Equipment and software were once shipped with its source code. Now, software is usually sent in binary and proprietary formats. Stallman felt that this closed source software defeated many of the mechanisms which he felt were important to software's continued growth. This concept was documented, much later, in *The Cathedral and the Bazaar*, by Eric Raymond.

Later that same year, Stallman quit MIT so he could pursue the development of software. He began developing new, free software. He called this the GNU project (pronounced "new"). GNU is a recursive acronym for "GNU's Not UNIX." To protect this free software, Copyleft and the GNU GPL (GNU General Public License) were written. In addition, the OSFTM or Free Software Foundation was created to promote this software and philosophy. For more information on this subject, see:

<http://www.gnu.org/gnu/thegnuproject.html>

Stallman began by creating a compiler (called the GNU C compiler or gcc) and a text editor (GNU Emacs) as a basis for further development. This has evolved, over time, to be a very complete suite of applications and infrastructure code that supports today's computing environment.

1.4 Linux

Stallman's project was still moving along nearly a decade later, yet the kernel—the core code running the computer—was not ready.

Then, in 1991, a Finnish student by the name of Linus Torvalds decided to write his own operating system. It was based on the concepts of UNIX®, but was entirely open source code. Torvalds wrote some of the core code. Then he did something quite original: he posted his code in a newsgroup on the growing Internet. His development efforts were complimented by others around the world and the Linux kernel was born.

Today's Linux distributions combine the Linux kernel with the GNU software to create a complete and robust working environment. In addition, many companies, such as Red Hat®, SuSE® or TurboLinux®, add their own installation scripts and special features and sell this as a distribution.

1.5 Linux clusters

This discussion extends beyond Linux clusters and encompasses clusters based on any operating system. Whenever you use two or more computers together to solve a problem, you have a cluster. However, this covers a tremendous amount of territory. This can be simplified by breaking clusters into increasing granular categories based on characteristics.

All clusters basically fall into two broad categories: High Availability (HA) and High-Performance Computing (HPC). HA clusters strive to provide extremely reliable services. HPC is a cluster configuration designed to provide greater computational power than one computer alone could provide.

1.5.1 High Availability (HA) clusters

HA clusters are not easily categorized. Indeed, we are sure that many people can offer valid reasons for why a different logical structure of organization would be appropriate. Our logical structure of organization is based on function. For example, we would organize a database cluster or a server consolidation cluster under the heading of an HA cluster, since their paramount design consideration is usually high availability.

Web clusters, while certainly an HA type of cluster, are often categorized by themselves, and we will discuss them in “Web clusters and Web farms” on page 5.

In a typical HA cluster, there are two or more fairly robust machines which mirror each other’s functions. Two schemes are typically used to achieve this.

In the first scheme, one machine is quietly watching the other machine and waiting to take over in case of a failure.

The other scheme allows both machines to be active. In this environment, care should be taken to keep the load below 50 percent on each box or else there could be capacity issues if a node were to fail. These two nodes typically have a shared disk drive array comprised of either a small computer system interface (SCSI) or a Fibre Channel; both nodes talk to the same disk array.

Or, instead of having both nodes talking to the same array, you can have two separate arrays that constantly replicate each other to provide for fault tolerance. Within this subsystem, it is necessary to guarantee data integrity with file and/or record locking. There must also be a management system in place allowing each system to monitor and control the other in order to detect an error. If there is a problem, one system must be able to incapacitate the other machine, thus preserving data integrity.

There are many ways of designing an HA cluster and the list is growing.

Web clusters and Web farms

Web clusters generally bring elements from many other computing platforms or other clusters and are often a hybrid of various technologies. A typical Web cluster is more a collection of machines creating an infrastructure than an actual cluster. We will explore a typical Web cluster by starting at the top, where it interfaces with the Internet, and finish at the bottom, where the data content is kept.

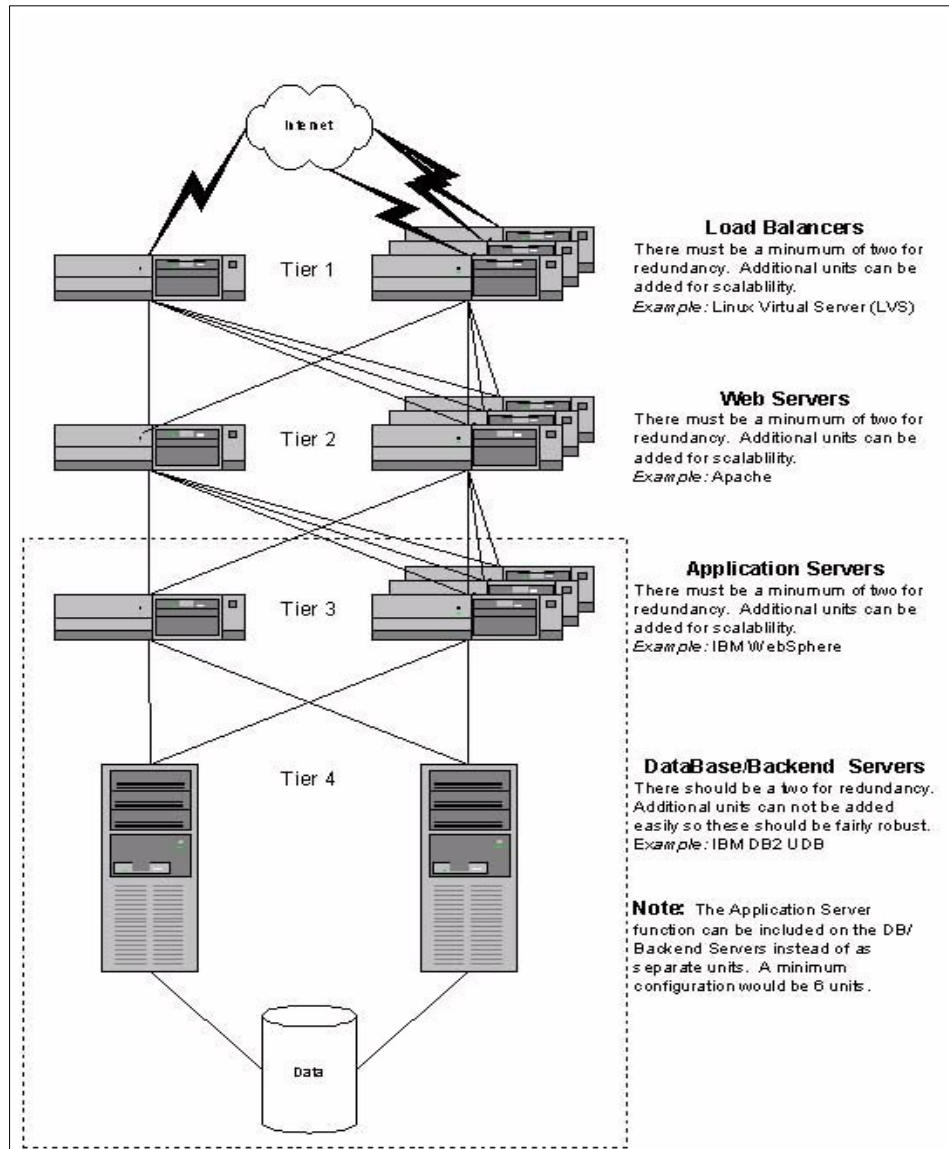


Figure 1-1 A generic Web cluster

Load balancing cluster

Figure 1-1 shows a typical, high availability Web cluster environment. At the top of the diagram is the Internet. Note that there are multiple redundant connections. These will tie in to a method of load balancing, either through dedicated hardware or through various software products, such as Linux Virtual Server (LVS).

Web server cluster

The next layer is where the Web servers reside. This is simply a group of machines running a Web server application, such as Apache. These servers can present static pages (sometimes called brochure ware) or they can have the core infrastructure of more complex pages with dynamic content. If a server fails, it will be noted by the load balancers, and future requests will be sent to other servers. If the load increases dramatically, additional servers can be easily added.

Application server cluster

The application layer is where server side code is kept and run. Server side Java™ is kept and run at this layer. This will generally be a type of high availability solution. At this level, it is still fairly easy to add additional machines to increase capacity.

Database server cluster

Finally comes the database layer. This layer usually requires some form of HA solution. It is not uncommon, in a very large operation, for this to be a mainframe database. It is not difficult, using the procedures outlined above, to create a robust, high availability database for the backend using Linux and various choices for databases, such as IBM's DB2 or the open source MySQL. It is fairly difficult to increase capacity at this level without some thought and planning in the beginning.

Also note that, in the Web server space, that various layers can be combined in one machine or one pair of machines. This is largely a consideration of the expected number of hits and how much hardware can be justified.

Server consolidation cluster

It is not uncommon in today's IT shops to have more and more file servers as departmental servers get added. At some point, there can be a pronounced cost and support advantage to consolidating these servers into a single large machine. If much of your data is stored here, it makes sense that some form of high availability solution be developed. The infrastructure defined in "Database server cluster" on page 7 can certainly be used as a solution. There are also several methods of attaching network storage with multiple servers capable of sharing the data. Regardless of the architecture, the goal is to take many existing servers and combine them into a single solution that never fails.

1.5.2 High-Performance Computing

High-Performance Computing (HPC) is a branch of computer science that focuses on developing supercomputers, parallel processing algorithms, and related software. HPC is important because of its lower cost and because it is implemented in sectors where distributed parallel computing is needed to:

- ▶ Solve large scientific problems
 - Advanced product design
 - Environmental studies (weather prediction and geological studies)
 - Research
- ▶ Store and process large amounts of data
 - Data mining
 - Genomics research
 - Internet engine search
 - Image processing

Embarrassingly parallel (high-throughput computing)

Some applications, by their very definition, can be subdivided into smaller tasks. Suppose you wanted to search the encyclopedia for the phrase “never on sunday.” You could pass out sections of the encyclopedia to different people and thus subdivide the task. However, sometimes the pattern you wish to find is less defined. Say, for example, it is OK to replace Sunday with other days of the week. This would mean seven times the search as in the first example, yet it would still be easy to subdivide.

This is generally called embarrassingly parallel (more recently called high-throughput computing). SETI (Search for Extraterrestrial Intelligence) at Home is an excellent example of this. Each parcel of information needs to be searched for patterns. So each computer is given a piece of information with the patterns that apply to it and the data is scanned. The results are then sent back to the original location to be compiled into a whole. Many processes, such as looking for similar gene sequences, lend themselves to this type of cluster architecture.

Distributed computing

Certain jobs, although broken apart, still require a fair amount of communication from one node to the next. The IBM Linux Cluster Solution uses Myrinet™ as the high-speed switch fabric to provide low-latency, high-bandwidth, interprocess communications from node to node. Myrinet is configured to provide full bisectional bandwidth which gives full bandwidth capability from any node to any node at any time and can be scaled to thousands of nodes simply by adding additional switches.

Beowulf clusters

Beowulf is mainly based on commodity hardware, software, and standards. It is one of the architectures used when intensive computing applications are essential for a successful result. It is a union of several components that, if tuned and selected appropriately, can speed up the execution of a well written application.

A logical view of Beowulf architecture is illustrated in Figure 1-2 on page 10.

Beowulf systems can be built using different pieces of hardware and software. There are classifications for this:

- ▶ **CLASS I:** Built entirely using commodity hardware and software. The advantages are price, and the use of standard technology (SCSI, Ethernet, IDE).
- ▶ **CLASS II:** Not necessarily built using commodity hardware and software alone. The performance is better than CLASS I.

The choice should be based on budget and the needs that you have; CLASS II is not necessarily the best choice.

In this redbook, we discuss a CLASS II Beowulf Linux cluster. The same approach is valid for 16, 32, and 64 node clusters.

Beowulf programs are usually written using languages such as C and FORTRAN, and use message passing to achieve parallel computation.

A Beowulf cluster can be as simple as two networked computers, each running Linux and sharing a file system via NFS and trusting each other to use the **rsh** command (remote shell). Or it can be as complicated as a 1024 nodes with a high- speed, low-latency network consisting of management and master nodes, and so on.

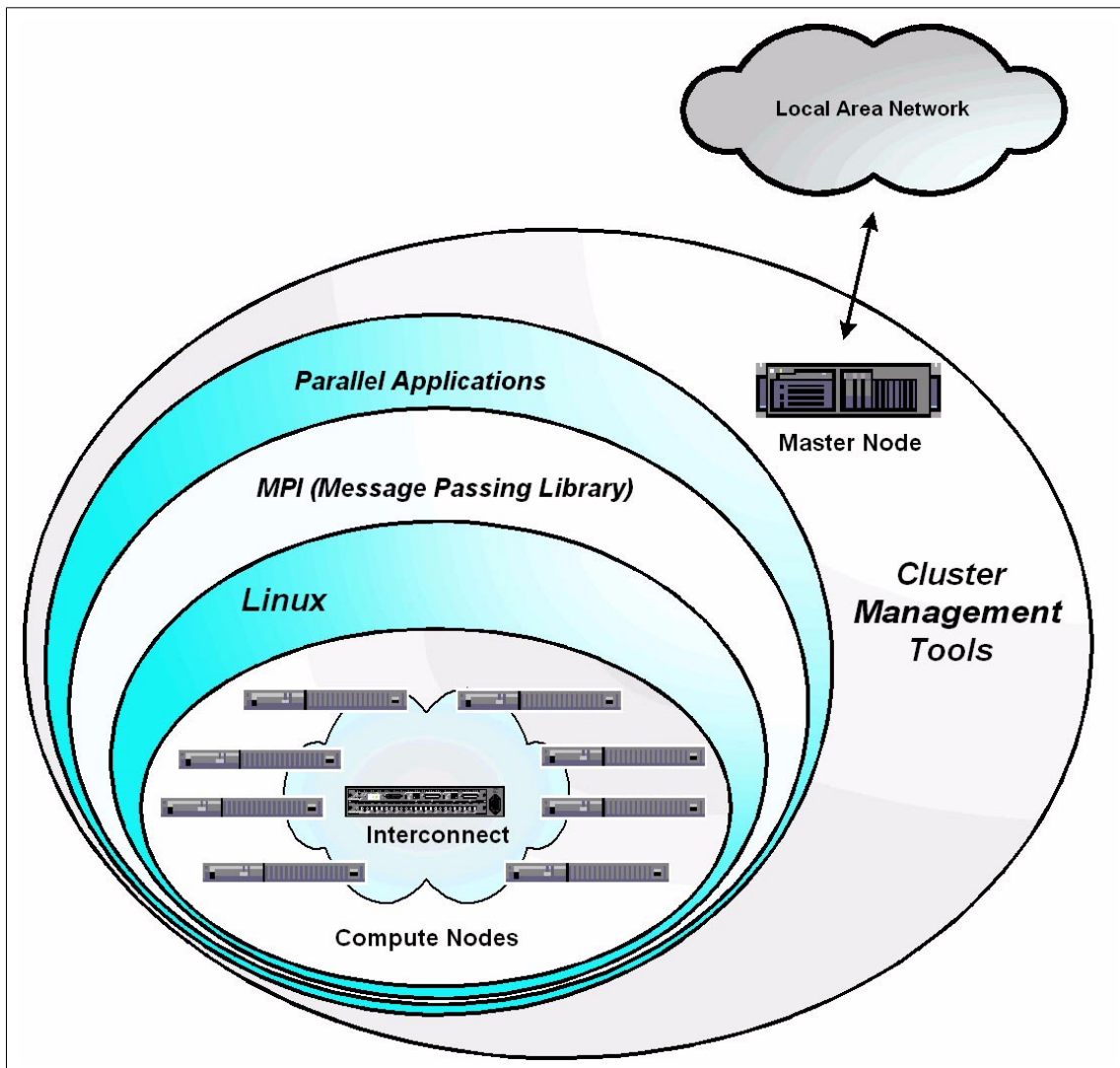


Figure 1-2 Beowulf logical view



General cluster architecture

In this chapter, we talk about concepts that can help you design a solution based on Beowulf architecture. It is intended for people who would like to have an idea about Beowulf architecture, which components make up the cluster, how to connect the components, and what to consider when you want to implement a Beowulf cluster as your solution.

We start by discussing application design and parallelism. Then we discuss hardware and software architecture from a theoretical perspective.

As an example for this discussion, we are using an eight node cluster. The same approach may also be used for up to 64 node clusters. Scalability issues are involved in clusters composed of 64 or more nodes, and it would be necessary to go into a discussion that is beyond the purpose of this chapter.

Figure 2-1 on page 12 illustrates where you can find some of the components in a Beowulf cluster.

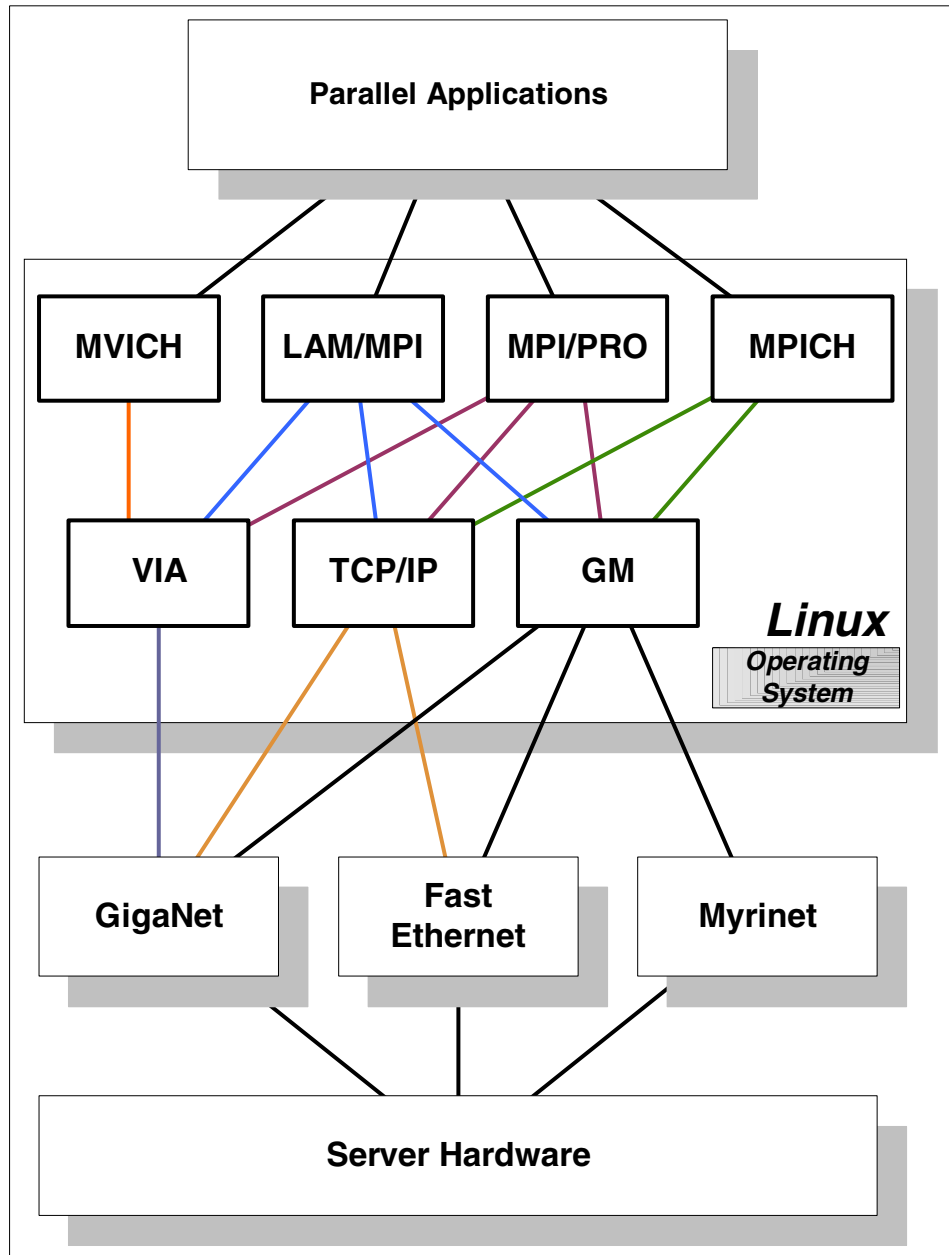


Figure 2-1 Components of a Beowulf cluster

2.1 Applications

Applications are the first elements that should be considered before buying any components for a Beowulf cluster. To that end, we introduce the concept of parallelism. To design a parallel application, we need to have a clear idea of what parallel means.

2.1.1 Parallelism

A definition of parallelism, found in *How to Build a Beowulf*, by Thomas L. Sterling, et al., is:

“Parallelism is the ability of many independent threads of control to make progress simultaneously toward the completion of a task.”

Another definition is:

“The simultaneous use of more than one computer to solve a problem.”

Both definitions are true because parallelism may appear in several ways. However, its usual definition is based on the connection and use of the memory subsystem.

Flynn’s taxonomy classifies parallel computing as:

- ▶ Single Instruction/Multiple Data (SIMD): When the processor executes the same instruction on multiple data at the same time.
- ▶ Multiple Instruction/Multiple Data (MIMD): When the processor executes different instructions on multiple data.

Visit <http://csep1.phy.ornl.gov/ca/node11.html> for more information on Flynn’s taxonomy.

CPUs may either run independently or cooperate in solving a problem. In either case, they are usually under the control of another processor, which distributes the workload and collects results from them.

This concept of parallelism is very important to keep in mind when designing and selecting components for a Beowulf cluster.

The following sections explain the three main constituents of parallel computing:

- ▶ Section 2.1.2, “Computer architecture” on page 14
- ▶ Section 2.1.3, “Software application program interface (API) architecture” on page 16

- ▶ Section 2.1.4, “Application architecture” on page 16

2.1.2 Computer architecture

In this section, we will consider the hardware, operating system, symmetric multiprocessing (SMP), and threads that are most helpful for working out the best design solution.

Before reading this section, we suggest that you read the parallel computing store example at:

<http://www.linuxdoc.org/HOWTO/Beowulf-HOWTO-4.html>

It is a very good example on how a machine can approach a real parallel problem. Table 2-1 gives a summary of the possible scenarios.

Table 2-1 Parallel computing store summary

| OS | Multi-Tasking | Thread | SMP | Message Passing | |
|-----------|---------------|--------|-----|-----------------|--|
| MS DOS | NO | NO | NO | NO | The CPU processes all the programs in the queue one at a time. |
| UNIX | YES | NO | NO | NO | The CPU dedicates an amount of time to each process in the queue to execute part of the program. You have the impression that all programs are running at the same time, but the CPU can only perform one operation at a time. |
| UNIX | YES | NO | YES | NO | Similar to the previous description, but more jobs benefit from the CPU time because there are more CPUs available. |
| UNIX | YES | YES | YES | NO | A job can run faster because it is divided across more CPUs simultaneously, hence it has more CPU time. Amdal's law limits the application speed to the slowest sequential portion of the program. |

| OS | Multi-Tasking | Thread | SMP | Message Passing | |
|-------------------|---------------|--------|-----|-----------------|--|
| UNIX ^a | YES | NO | YES | YES | Similar to the previous description, but the advantage is scalability. The disadvantage is the extra overhead added by communications. |

a. The CPU may not necessarily be on the same machine.

We now know that parallel computing is performed on machines with either single or multiple CPUs and the hardware may be configured in several ways:

- ▶ Distributed clusters of machines: Each node is an autonomous computer. They perform parallel programming using message passing, sending data from a program running on one node to a program running on a different node by using Single Instruction Multiple Data (SIMD) or Multiple Instruction Multiple Data (MIMD).
- ▶ Massively Parallel Processor machines (MPP): This has hundreds of processors programmed on the SIMD model. It includes a large array of processors with one instruction unit that controls the entire array by fetching each instruction and commanding all the processors in the array to carry the instruction out in parallel, each with its own data.
- ▶ Shared Memory Processors (SMP): These are computers with more than one CPU. You can run separate programs on each of the processors simultaneously or a single program in parallel across the processors.
- ▶ Distributed clusters of local memory machines: This is a combination of the single CPU and SMP machines via shared memory.

Beowulf may be considered a cluster of the SMP or local memory machines based on the MIMD model.

Since SMP machines communicate via shared memory, you may think that this solution is better because an overhead introduced by a network does not occur. That is almost true, but you also have limited scalability because of memory limitations; the machine cannot have infinite memory.

This does not mean that you cannot have the Beowulf compute nodes with an SMP machine. We discuss this topic in more depth in Section 2.2.7, “Compute” on page 24.

2.1.3 Software application program interface (API) architecture

The API is the interface by which an application program accesses the operating system and other services. An API is defined at source code level and provides a level of abstraction between the application and the kernel to ensure the portability of the code.

An API can also provide an interface between a high level language and lower level utilities and services which were written without consideration for the calling conventions supported by compiled languages. In this case, the API's main task may be the translation of parameter lists from one format to another and the interpretation of call-by-value and call-by-reference arguments in one or both directions.

Parallel architecture may have:

- ▶ Messages: May be implemented both for SMP and cluster machines; they require copying of data between nodes. Latency and the speed of this operation are the limiting factors. The advantage of writing code using messages on an SMP machine is that if you decide to move to a cluster, you can easily add a machine without have to recreate your application from scratch.
- ▶ Threads: They are designed to work fast on SMP machines because of shared memory. They do not need copying as much as messages do. The disadvantage is that you cannot extend them beyond one SMP machine because data is shared between CPUs. NUMA may allow that, but we are not considering NUMA architecture at present due to its expense and because Linux is not yet mature on that platform.

2.1.4 Application architecture

Up to this point, we have considered machine hardware and APIs. Now, we discuss how to design the application. We show the logical sequence you should follow to develop a good application and we explain how to resolve the bottlenecks.

Batch and parallel jobs

You can have two kinds of jobs: a batch job and a parallel job.

A batch job is an application that runs independently on each node of the cluster without the need to interact deeply with the other nodes. This kind of application usually performs calculations on a set of data and at the end of the job passes the result to a control node responsible for collecting the results from all the nodes.

A parallel job is a job that needs to talk with the other compute nodes of the cluster. This operation needs message passing and network activity.

The reason for using a cluster is to speed up the execution of a program, but sometimes SMP machines that are not in a cluster can be fast enough.

With this in mind, we should consider using a cluster in the following circumstances:

- ▶ Problem size: If an application requires a lot of memory, a workstation may not be enough; a cluster might be a good solution.
- ▶ Real time answer: Problems that need a faster response. In this case, you should also consider the number of users who access the cluster at the same time. If too many users share the cluster resources at the same time, the cluster performance will slow down and maybe a workstation could be an adequate solution.
- ▶ Is the application suitable for a cluster or do I need to port it?
- ▶ How long does the application take to run?
- ▶ How large is the application? Can I fit it onto any machine?

If at this point we have decided to use a cluster and to use parallel computing, so we need to create our application.

2.1.5 Creating the application

We need to clarify the concepts of concurrency and parallelism, because we have to be clear about which part of the program can be parallelized.

- ▶ Concurrency is a property of the application and defines the parts of the program that can be computed independently.
- ▶ Parallelism is the ability to run concurrent parts of a program that are executed on separate processes at the same time. Parallelism is also a property of the hardware.

2.1.6 Bottlenecks

We have to consider possible bottlenecks, as they actually drive both application design and hardware choice.

We need to figure out if the application is:

- ▶ Processor bound: CPU limited
- ▶ I/O bound: Hard disk or I/O network limited

Communication speed and latency between the nodes are the limiting factors of parallel performance, so it is very important that your application compensate for those factors. The cost of communication time must be worth the improved performance in computation. It is important to realize that fast processors do not equal better performance if communication is a bottleneck.

I/O intensive applications, such as databases, need to transfer large amounts of data, so both CPU and I/O performance are required. This is unlike the scientific field where just high performance indexing based on the combination of CPU and the network are considered to be important.

Anatholy F. Dedkov and Douglas J. Eadline, in *Performance Consideration for I/O-Dominant Applications Parallel Computers*, formulated this rule regarding processing speed:

“For two given parallel computers with the same cumulative CPU performance index, the one which has slower processors (and a probably correspondingly slower inter processor communication network) has better performance for the I/O dominant applications (while all other conditions are the same except the number of processors and their speed).”

Writing and porting

At this point, we should have defined:

- ▶ Which part of the program can be concurrent.
- ▶ What are the bottlenecks and how to estimate the parallel efficiency.

We have not yet described the concurrent parts of the program.

We have two options to consider: explicit and implicit parallel execution.

Explicit

Explicit methods are determined by you or the programmer, and you have to add messages using either MPI or POSIX threads.

Note: Threads cannot move between machines.

The explicit method is very difficult to implement and debug, so programmers usually use programming languages, such as C and Fortran, and MPI libraries, to implement applications. This approach is the most general and efficient, and the resulting code looks to be easily portable.

Note: FORTRAN has the largest amount of support, in terms of libraries and tools, for parallel computing.

Implicit

Implicit methods are created when you or the programmer provide some information about the concurrency of the application. Then specific compiler directives, such as High-Performance Fortran (HPF) or parallelization tools, analyze the program and perform automatic detection of parallelism, and decide how to execute the concurrency part of the program.

Program portability can be achieved using standards such as Message Passing for FORTRAN and C/C++ programs, and OpenMP for shared memory, which is implemented by compiler directives.

2.2 Hardware architecture

This section shows which components are in a Beowulf cluster, from each node function to the network architecture. We show an example of how a Beowulf cluster can be designed in Figure 2-2 on page 20.

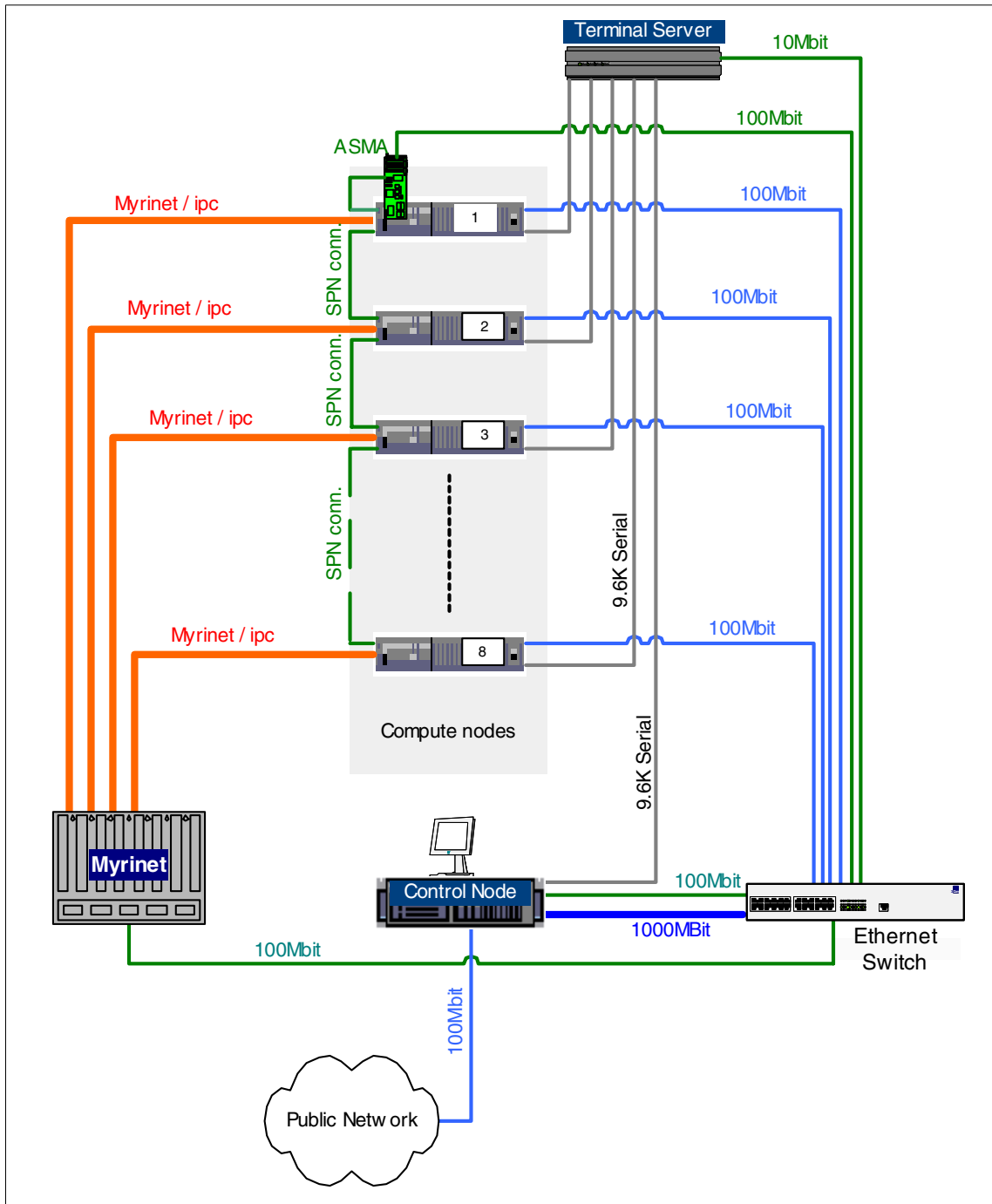


Figure 2-2 Beowulf Linux cluster example

2.2.1 Cluster components

Previously, we discussed parallelism and how applications should be designed for maximum efficiency in this environment. Now we will focus on the functions that make up the cluster.

We will divide the cluster nodes based on their function. It is important to note that these functions may reside on one or more machines. In a small cluster, all of the functions will likely be on one machine. In a larger cluster, you will likely divide the functions across many units. In very large clusters, a single function might be spread across several similar nodes. The following lists the node functions and structure:

- ▶ User
- ▶ Control
- ▶ Management
- ▶ Storage
- ▶ Installation
- ▶ Compute Node

Later, we create the nodes and show you how those functions can be on the same node, and therefore use the same server or hardware.

It is not essential that each function be on a separate machine; this is up to you. Based on budget and performance, you may decide to install either one function per server (or more), or to consolidate function onto just a few servers.

2.2.2 User

The user node is the gateway for the outside world to access to the cluster. Users usually login to the user node machine and compile or run their jobs.

You may want to have hardware redundancy on this machine node. If it fails, users may not access the cluster and use its powerful calculus capacity.

The user node is an important part of the cluster and we recommend having hardware redundancy. If you do not have the user node up and running, your cluster cannot function, because there is no opportunity to run a job.

The user node can also be on the same server that is used for management or installation.

Important: We suggest you use redundant array of independent disks (RAID) adapters to protect the data on the user node.

2.2.3 Control

The control node can be on the same machine that takes care of management and installation functions. It is responsible for controlling the cue or batch jobs. The control function may be located on one of the compute nodes or on any other machine responsible for controlling or managing the cluster.

The control node provides services such as Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and Network File System (NFS). In addition, the portable batch system (PBS) and the scheduler are usually installed on this machine.

You need redundancy on hardware and data because, unlike in the case of compute nodes, the failure of the control node may affect the entire availability of the cluster. We suggest using:

- ▶ Redundant fans
- ▶ Redundant power supply
- ▶ RAID (to protect the data)

2.2.4 Management

Management is the function that manages and controls the cluster and its components, for example, switches. It is either on dedicated machines called management nodes, or, for budget reasons, shares the machine with other functions, such as the master, installation, and compute nodes. It may also be found on a dedicated virtual local area network (VLAN), defined as management, and accessible only by the administrator (for security reasons).

There might be more than one management node in a cluster, such as in large clusters; it depends on how many nodes and components you need to manage. As the cluster manager, it takes control of the nodes and collects Simple Network Management Protocol (SNMP) alarms.

You need redundancy on hardware and data because, unlike in the case of compute nodes, the failure of the control node may affect the entire availability of the cluster. We suggest using:

- ▶ Redundant fans
- ▶ Redundant power supply

- ▶ RAID; to protect the data

Possible configurations are:

- ▶ Advanced System Manager Adapter (ASMA) or Wiseman card to allow a daisy chain connection of service processors to receive alarms from the compute nodes and for sending SNMP traps.
- ▶ On-board or peripheral computer interconnect (PCI) ethernet adapter, more than one in some cases, to allow the management of several VLAN or networks.
- ▶ If it performs the installation function as well, a gigabit ethernet adapter may be helpful in giving high-bandwidth performance and reduce the node installation time.

2.2.5 Storage

The storage function is performed by dedicated machines and is responsible for storing large amounts of data needed by the applications running on compute nodes. Nodes that perform this function are usually called storage nodes. The storage node is the answer to the question, “How do we feed large amounts of data to the cluster?”

Since it is difficult to store a TB on a single server or on one of the cluster nodes, there is a need for SAN, or dedicated servers; the network may be a bottleneck at this point.

We suggest equipping storage nodes with:

- ▶ ServeRAID controllers that give data protection using one of the different RAID levels available. See the redbook *Netfinity Server Disk Subsystem*, SG24-2098 for more information.
- ▶ Fibre Channel, for better speed and data transfer rate.
- ▶ Gigabit Ethernet or Myrinet, for high-bandwidth performance.
- ▶ 10/100Mbit Ethernet, for eventually connecting to a management VLAN or network.

2.2.6 Installation

The installation node is responsible for the installation of the compute nodes and is sometimes referred to as a *staging* server or node. It is usually a server that exports a file system containing the operating system, libraries and all the necessary software for the compute nodes. It usually has a Gigabit Ethernet PCI adapter to allow more bandwidth and to speed up the cluster installation.

An installation server usually shares a file system that can be accessed via NFS, Web or file transfer protocol (FTP). This is where the operating system and relevant software needed for each compute node to work are installed.

The installation node can also be configured as a compute node since they carry out computations. It is the only node on the public network, since users need to access it to be able to run their jobs.

2.2.7 Compute

The compute function or node is the computational heart of the cluster. Its main activity is just to perform calculus.

The compute function is usually called the compute node, especially for large clusters, because this function is assigned to dedicated machines.

The choice of hardware and configuration parameters of the compute node should be based on the applications that will be run on the system. The following characteristics should be considered:

- ▶ Processor Type
- ▶ Size and speed of L2 cache
- ▶ Number of processors per node
- ▶ Speed of front-side bus
- ▶ Memory subsystem scalability
- ▶ PCI bus speed

As we said, the application drives the decision on how to build the compute nodes. For example, if the compute node is required to access the cache frequently, the size of the L2 cache and memory subsystem should be considered; a large cache may enhance the performance. However, some experiments show that if your application can fit on the L2 cache and does not need to frequently access the memory subsystem, the performance of the compute node increases.

On the other hand, applications that use intensive amounts of memory will benefit from a faster CPU, system bus, and memory subsystem.

From a processor performance point of view, the ideal is one processor per compute node. If you have budget restrictions, we suggest using an SMP machine. In terms of price performance ratio, an SMP machine with two processors is better than a machine with three or four.

The Linux internal scheduler determines how these CPUs get shared. You cannot assign a specific task to a specific SMP processor. You can, however, start two independent processes or a threaded process and expect to see a performance increase over a single CPU system.

Hardware high-availability is not as important as other parts of the cluster. If there is a node failure, it just loses a part of the job that can be run later on another node.

If your application uses message passing (we assume the necessity for high network activity and low-latency), or if the application requires the transfer of large pieces of data across the network, you may want to consider Gigabit or Myrinet PCI adapters. In this case, we recommend choosing machines with 64-bit, 66MHz PCI busses to get the benefit of full bus performance from those adapters.

You should look at the specifications of the Ethernet card you are using, since we know that, currently, Gigabit does not support Pre-Execution Environment (PXE) capability to boot the machine via the network.

It is possible that a compute node can take control of other functions too, such as:

- ▶ Management
- ▶ Installation
- ▶ Control

It is desirable to condense multiple functions on the same node if your budget is an important factor, or for small clusters

2.3 Network architecture

Networking is the conversion of a group of machines into a single system. It also allows remote access to a machine and both hardware and software are necessary for the network to work.

Networking in Beowulf clusters is very important. The most demanding communication occurs between the compute nodes.

Because of the communication demands between compute nodes, network performance is a very important topic to be considered in a Beowulf type of cluster. Depending on the characteristics of the programs being executed, high-bandwidth, speed and low-latency networks may be required. The network communication will determine the Beowulf-class system and the degree of complexity in building efficient programs.

The primary bottlenecks of Beowulf are network bisection bandwidth, latency, and global synchronization.

Technologies

Let us look at some of the hardware technologies and protocols useful for fulfilling the needs of a cluster inter process communication (IPC).

There are several technologies and protocols usable for Beowulf communication:

- ▶ Fast Ethernet
- ▶ Gigabit Ethernet
- ▶ Myrinet

Switch

The demand for high bandwidths and the growing number of nodes on the same network are the most important reasons to use a switch. A switch accepts packets on twisted-pair wires from nodes and does not broadcast to all the connected nodes to determine who the receiver is, as a hub does.

Instead, a switch uses the destination address field of the packet to determine who the recipient is and sends the packet to it alone. However, as the number of nodes in a cluster increase, one switch may not be enough.

We must guarantee *full bisection bandwidth*, because each node must be able to talk to any other node of the cluster on the same network without having routers in the way, because routers can affect the performance. If you need to build a $2N$ -port full-bisection switch using N -port full-bisection switches (for example layer 2 switches or Myrinet), you should use 6 N -port switches and $2N$ cables. We show how to interconnect switches to guarantee the full bisection bandwidth in Figure 2-3 on page 27.

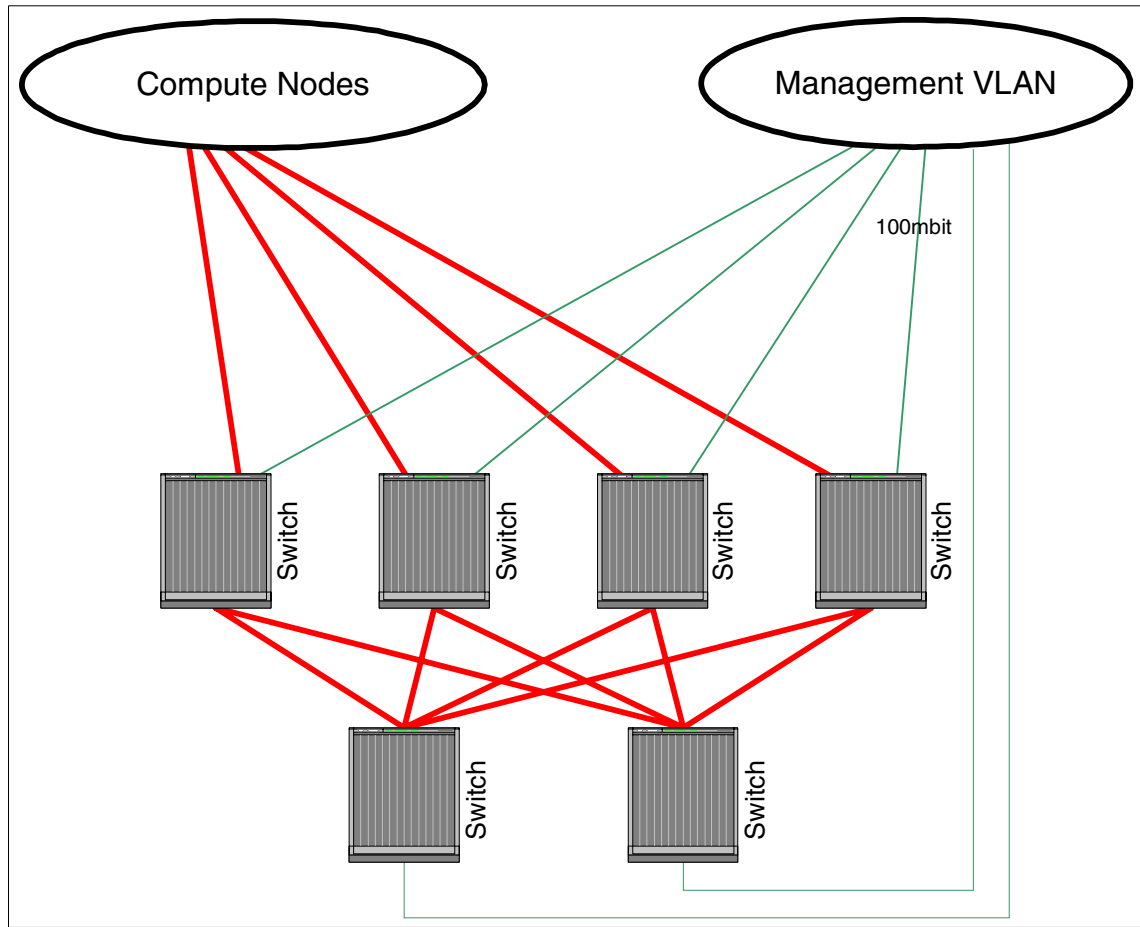


Figure 2-3 Full bisection bandwidth example

2.3.1 How to design the cluster network

We have looked at some of the technologies and protocols used for Beowulf IPC. Now we can explore how to design the network components of our cluster, because the cluster needs a network topology in order to be able to communicate between the nodes (as well as with the rest of the world).

The components of the cluster network can be summarized as:

- ▶ IPC: Needed to allow nodes to pass messages and chunks of data.
- ▶ Management: To satisfy all the needs of cluster administrators.
- ▶ Cluster: Used for I/O and job traffic only.

- ▶ **Public:** We need a way to talk to the LAN or wide area network (WAN) in order to have access to the cluster.

You have to decide what to do for your network:

- ▶ A single network topology, where the compute, master, installation, and storage nodes are all connected.
- ▶ Divide your network into several virtual local area networks (VLANs).

We usually create several VLANs. To do that, you need an intelligent or layer 2 switch capable of managing virtual networking, that is, the ability to configure logical topologies on top of the physical network infrastructure. VLAN offers benefits in terms of efficiency in the use of bandwidth, flexibility, performance, and security. From a logical point of view, VLAN technology is a segmentation of the network in different broadcast domains so that packages are only switched between ports designated to be part of the same VLAN.

We suggest creating the following VLANs, as represented in Figure 2-4 on page 29:

- ▶ **Management:** This VLAN is used to access Myrinet or another layer 2 switch used for IPC, Service Processor Network, and EquinoxTM (for management purposes only). Telnet and SNMP are used to manage the switch, Equinox and the Service Processor Network. This VLAN is usually isolated for security reasons.
- ▶ **Cluster:** The compute and storage nodes use this VLAN for I/O traffic only. The master node uses this network for cluster management and installation only.
- ▶ **Public:** The master node is connected to this network for user access and to launch jobs.
- ▶ **Service Processor Network:** An Ethernet link to the Service Processor; used for management purposes.

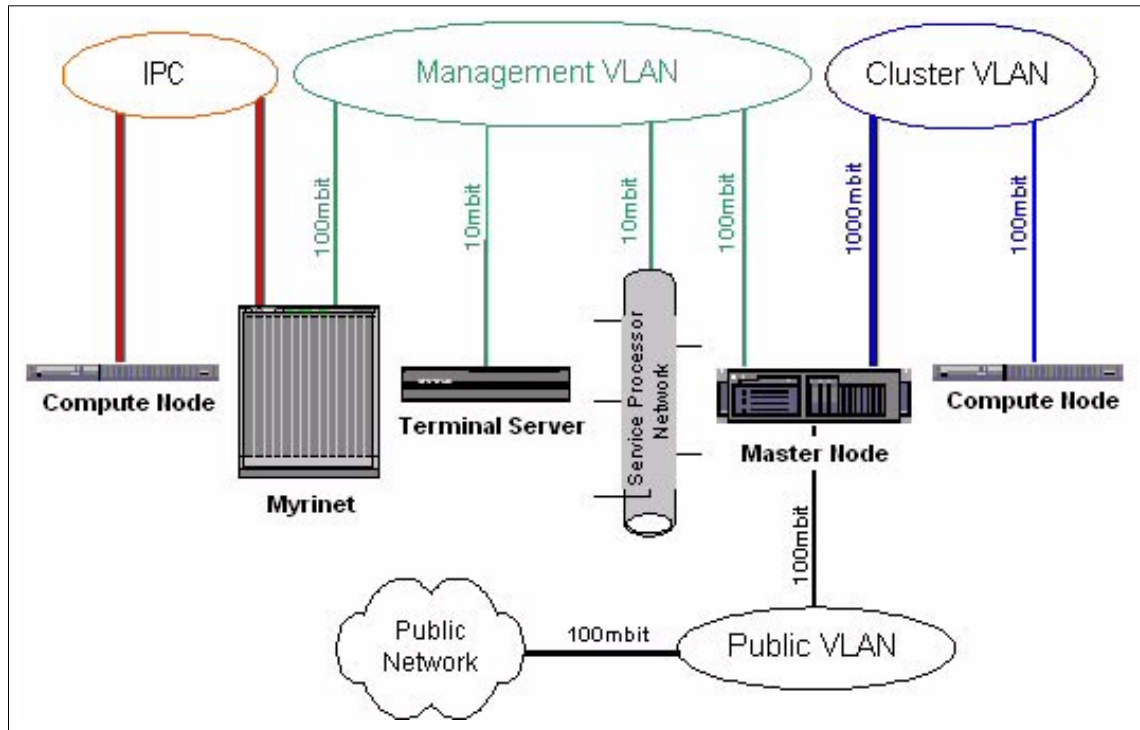


Figure 2-4 Beowulf cluster network scheme

2.3.2 Remote access/Equinox terminal server

The Linux operating system, like all UNIX-based operating systems before it, talks at a very low level to the serial port on the machine. An extremely high level of remote control is available through this serial port. Remote management of the operating system is facilitated by the use of a terminal server.

The terminal server connects to the serial port on each compute node and allows us to telnet, or otherwise share, this communications port. We can remotely access the operating system on any and all compute nodes by using the terminal server.

2.4 Software architecture

We have discussed the application and hardware design of a cluster. However, the cluster is not limited to those components alone. To develop and run applications on a cluster, you need several software components:

- ▶ Operating system
- ▶ Device drivers
- ▶ Libraries
- ▶ Development and debugger tools
- ▶ Compilers
- ▶ Resource management tools

See Figure 2-5 for an overview of the software components.

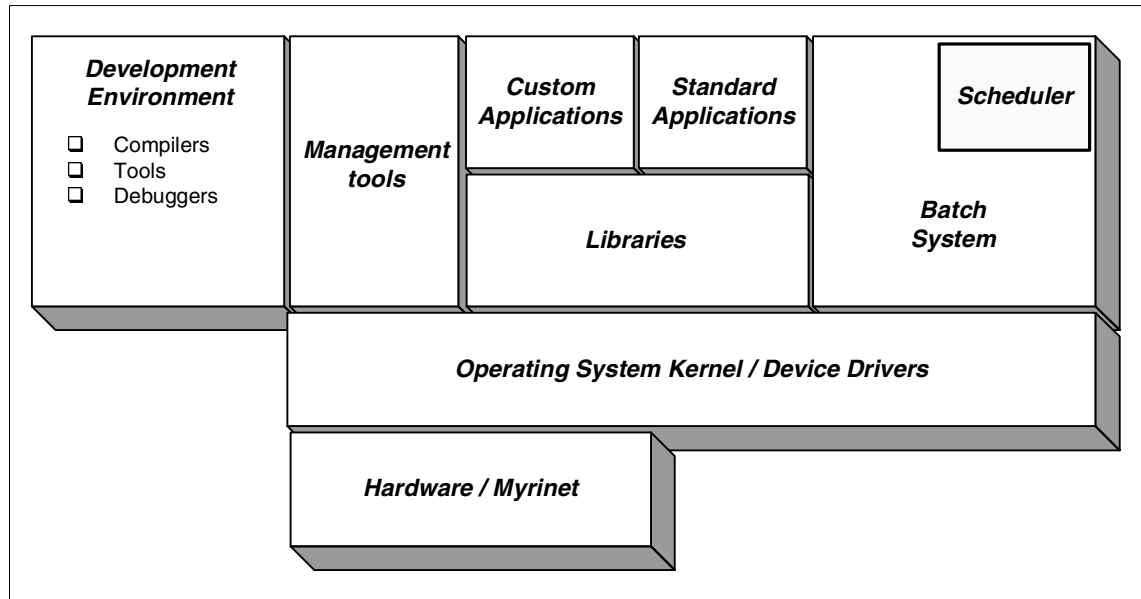


Figure 2-5 Beowulf software components

For additional information on some of the main software components and a detailed description of development and cluster management tools, please see Chapter 3, “Components overview” on page 37.

2.4.1 Operating system

An operating system performs several tasks, like recognizing input from the keyboard, sending output to the display panel, controlling peripheral devices, and keeping track of files and directories on the disk. Several operating systems can be used for creating a Beowulf cluster and Linux is one of them. We normally interact with the operating system through a set of commands, for example, **bash** on Linux.

2.4.2 File system

The file system is a major part of any operating system; it is a repository for programs and data. The file system has two separate meanings in Linux:

- ▶ It can be the concept of a root directory hierarchy, with different partitions mounted under the root directory. There is no letter, as with Microsoft® Windows, to identify the partitions, and everything is seen as a file.
- ▶ It can refer to the Specific formats for representing files or directories on a disk or memory, for example, EXT2 and JFS.

In a cluster environment, it makes sense to talk about Distributed File Systems. Every Beowulf node, if not a diskless client, has a local file system that may need to access processes running on other nodes. That is the basic reason for using a distributed file system in a Beowulf cluster.

A distributed file system has characteristics that distinguish it from a local file system:

- ▶ Network transparency: Remote and local files can be accessed using the same system calls.
- ▶ Location transparency: The file name is not bound to its location, because the server containing it is not part of the path.
- ▶ Location independence: If the physical location of the file changes, its name does not necessarily change, because the name of the server containing it is not part of the file path.

Parallel file systems

One critical area where Linux clusters were devoid of support was the parallel file system, which is essential for high performance I/O.

To begin discussing parallel file systems, we must look again at local file systems. Local file systems work by creating a map between the blocks found on the storage devices on one side and files and directories on the other. Because of this, they see the devices as local and not shared, so there is no need for the file system to enforce device sharing semantics. To improve the file system performance, we could improve caching and combine file system operations to reduce the number of disk accesses required for each file system operation.

Current technologies allow multiple machines to share storage devices. File systems that allow these machines to simultaneously mount and access files on these shared devices are called shared-disk file systems. This is in contrast to traditional distributed file systems where the server controls the devices and is the focus of all data sharing.

A shared-device file system approach offers several advantages over traditional client-server file system approaches:

- ▶ Increased files system availability: There is no single point of failure on the host side because the file system is still accessible to the other hosts.
- ▶ Load balancing: The client can access any set of data on any storage device.
- ▶ Scalability in capacity, connectivity, and bandwidth can be achieved without the limitations inherent in network file systems, such as NFS, which are designed around a centralized server.

Parallel file systems are vital for large I/O intensive parallel applications. Global file systems (GFS) and parallel virtual file systems (PVFS) are currently available, and IBM plans to implement general parallel file systems (GPFS) on the Linux platform.

2.4.3 Interprocess Communication (IPC)

Interprocess communication (IPC) is the exchange of data between one process and another, either over the network or on the same computer. The exchange of data implies a protocol that guarantees that a response occurs at each request. The technologies presented in “Technologies” on page 26 are suitable for IPC. We recommend using a high bandwidth and low latency technology, such as Myrinet or Giganet, with an application that uses message passing between processors on different motherboards. Fast Ethernet is adequate where networking is not a bottleneck. IPC is an element that fits into the definitions of the levels and classes of a Beowulf cluster.

Parallel Virtual Machine (PVM)

PVM is an application programming interface (API) that facilitates parallel computing. PVM is designed for use across clusters. If you code your application using the PVM API, then you have enabled it to run across a cluster. In this environment, one machine runs the PVM console. The PVM console is aware of all of the nodes available to it. A job is submitted to the PVM console where it is distributed across the PVM nodes. Results are returned to the PVM console and recompiled into a complete result.

PVM is very effective for jobs that can work in a loosely coupled environment, that is, those that can work well in individual pieces. There are other solutions that are probably better suited for jobs with more stringent communication requirements.

Message Passing Interface (MPI)

MPI specifications are commonly used where there is a demand for a logical structure layer to match a physical communication layer, for example when applications need to interact with several CPUs on different motherboards.

MPI is a standard specification of message passing libraries for writing parallel programs. MPI standard was first released in 1994 from MPI Forum (MPIF). It specifies a portable interface for writing message passing parallel programs. For details about MPI and MPIF, please visit:

<http://www.mpi-forum.org>

Message passing has been widely adopted because of its close association with the physical attributes of SMP architecture. Message passing supports an interaction between sequential processes. One or more processes run on each processor and they interact via messages conveyed across the physical network.

MPICH is the most widely used portable implementation of the full MPI specification for a wide variety of parallel and distributed computing environments. The MPICH implementation is particularly popular on heterogeneous workstation networks and Beowulf clusters, where no vendor implementation is available. MPICH is freely available and can be downloaded from the Web at:

<http://www.mcs.anl.gov/mpl/mpich>

The message-passing programming paradigm assumes that an application program is split into several tasks, each task having its own private data and addressing space. There is no shared global data between any tasks of the program. All synchronization and communication between the tasks is accomplished by explicit library calls between tasks, such as MPI_SEND or

MPI_RECEIVE. Because the programmer has total control over task interaction, better scalability can be achieved by properly designing synchronization and communication between tasks, more so than with any other parallelization technique today.

See <http://www-unix.mcs.anl.gov/mpi/mpich/index.html> for more information on MPICH and related topics.

OpenMP

PVM and MPI are specifically designed to work across a cluster. OpenMP is designed to work within a symmetric multi processor (SMP) machine. In the SMP model, the processors have a shared memory address space, which can be used to transport messages. OpenMP uses this shared memory address space and is therefore very effective in the SMP environment. However, it is not useful in a clustered environment. Yet in certain situations, OpenMP can be combined with MPI or PVM to utilize both SMP and clustered environments.

2.4.4 Resource management

From a user point of view, a cluster is like any other server or PC. The user logs in and runs interactive programs. This is true for any users that access the same cluster. However, you should discourage intense usage by individuals; if too many of the cluster's resources are used at the same time, the performance will slow down. The scheduler is responsible for resource management in the cluster. Although it is possible to run a job that uses all available resources, a job typically uses far fewer resources than that. With the scheduler, you specify how many nodes you need and how long your job is expected to run (along with various other information). The scheduler then arranges to run your job in the most expeditious manner, trying to keep the cluster as fully utilized as possible at all times.

Job management

Beowulf clusters require a mechanism to manage and schedule the jobs and workload from different users. The idea is to utilize any given time and hardware resources as efficiently as possible. This ensures optimized use of the cluster calculation time and will obtain results as quickly as possible.

Resource manager

It is essential that the user is able to request an appropriate computational node for a given task. To allow this to happen, the resource manager coordinates the actions of all other components in the batch system by maintaining a database of all resources, submitted requests, and running jobs.

Conversely, in a heterogeneous environment, it is important that the resource manager conserve its most powerful resources as long as possible, unless they have been specifically requested. It also needs to provide some level of redundancy to deal with computational node failure and scale hundreds of jobs on thousands of nodes. Additionally, it should support hooks for the accumulation of multiple machines at different sites.

Job scheduler/policy manager

The job scheduler takes the node and job information from the resource manager and produces a list. This list is sorted by the job scheduler and tells the resource manager when and where to run each job.

The policy manager must have the flexibility to arbitrate a potentially complex set of parameters required to define a fair share environment, and yet retain the simplicity of expression that will allow the system administrators to implement politically driven resource allocations.

For more information on the resource manager and the job scheduler/policy manager, see *Portable Batch Scheduler and the Maui Scheduler on Linux Clusters*, found at:

http://www.scl.ameslab.gov/Publications/HalsteadPubs/usenix_2k.pdf



Components overview

In Chapter 2, “General cluster architecture” on page 11, we discussed the theories and general architectures regarding Linux clusters. We now move on to listing and describing the components that make up an IBM *@server* xSeries Linux Cluster Solution.

It is worth spending time at this stage to select the right components to form your cluster according to your environment and demands. We have divided this part of the components overview into the following topics:

- ▶ Hardware
 - Nodes (xSeries)
 - Management (ASM/SPN, ELS, and KVM)
 - Network (FE, GE, and Myrinet)
- ▶ Software
 - Operating system (Linux, file systems, and drivers)
 - System management (xCAT and IBM Director)
 - Development (compilers, libraries and debugger)
 - Resource management (batch systems and scheduler)

3.1 Hardware

The current IBM Cluster offering is built on the xSeries x330 and x340 platforms. In this chapter, we discuss all the hardware components and technologies used to build an IBM @server xSeries Linux Cluster Solution, from node functions to network architecture and components.

3.1.1 Head node

Head node is a generic term which has been in use for a fair amount of time. We are actually trying to move away from the term, because it is not very descriptive. However, as a generic term, it is very appropriate.

As discussed in Chapter 2, “General cluster architecture” on page 11, the head nodes generally provide one or more of the following functions:

- ▶ User node and control node
- ▶ Management node
- ▶ Installation node
- ▶ Storage node

The head node or nodes are typically built using x340s, also known as a 4500R. In a small cluster, say eight compute nodes, all of the functions can be combined in one head node. In larger clusters, the functions will probably be split across multiple machines for security and performance reasons. The control of the cluster is performed from one or more of the head nodes.

The x340 is configured similarly to the compute node, along with additional high-availability features, such as hot-swap, redundant power, and ServerRAID adapter. We typically configure the x340 with a hardware RAID 5 array consisting of six 18 GB slim height drives for 90 GB of usable storage. The IBM ServerRAID adapter has full Linux support, including an IBM-written general public license (GPL) driver in the Linux kernel, a native Linux IBM ServerRAID Manager utility for creating and managing RAID arrays, an SNMP agent that monitors and sends alerts, a client-server remote management agent, and command line utilities that allows automated management through scripts.

The x340 takes up 3U in a rack. It has up to two CPUs and four dual inline memory modules (DIMMs). It has two PCI busses, one 64-bit with three slots and one 32-bit with two slots, making a total of five PCI slots. We can install up to six hard drives, three natively and three more with an expansion kit. These drives can be configured in a RAID array with the optional ServerRAID adapter.

3.1.2 Compute nodes

The compute nodes form the heart of the cluster. The user, control, management, and storage nodes are all designed in support of the compute nodes. It is on the compute nodes that all computations are actually performed. These will be logically grouped, depending on the needs of the job and as defined by the job scheduler. The compute nodes are typically built using the x330.

The x330 features two Pentium® III processors in a 1U form factor, a 133 Mhz Front Side Bus, two 64 bit PCI slots, and an onboard Service Processor. The 64-bit PCI slots allow the server to take advantage of the bandwidth capabilities of the Myrinet 2000.

The x330 takes up only one unit or 1U in a rack. In that 1U space, we fit two hard drives, two CPUs, four DIMM memory slots (allowing up to 4 GB of memory) and two PCI card slots. This is a typical compute node configuration.

3.1.3 Management

Before looking at the software elements of the cluster, we need to discuss some hardware components of the cluster that are used for its management.

Cluster management begins with our ability to control the machines in the cluster. Clearly, it is possible to walk around and push a power off switch in order to power off a unit. However, this is often not practical and, for certain tasks, a higher level of control is needed. There are three components to help us to deal with such issues:

- ▶ Terminal server
- ▶ ASM/SPN network
- ▶ KVM switch

Terminal Server

The terminal server is a device that connects many terminals to a local area network (LAN) through one network connection. It provides serial connectivity to the network for serial devices and printers.

We recommend using a terminal server in the cluster for two reasons:

- ▶ Access to the cluster machines in case you cannot get access via the LAN. For example, it may be down for maintenance.
- ▶ Medium access control (MAC) address collection. If you do not have a layer 2 switch capable of this, the Equinox terminal Server may allow you to get the MAC addresses for the installation node.

We use an Equinox ELS Terminal Server in our lab. It offers:

- ▶ Local area transport (LAT) and Transmission Control Protocol/Internet Protocol (TCP/IP) support
- ▶ ELS Terminal Server Features
- ▶ Support for TCP/IP and DEC LAT protocols
- ▶ Built-in parallel port
- ▶ Multi-session support
- ▶ Support for NetWare™ IPX print services

The terminal server is an optional component of the cluster. It is not vital for basic cluster function, but its presence is useful.

Please visit the Equinox web site <http://www.equinox.com/> for more information.

Service Processor Network

The onboard Service Processors (SP) of xSeries machines allow for remote power management, environmental monitoring, alerts and remote BIOS. They are linked together to form Service Processor Networks, which are created from two elements. These are:

Advanced system management (ASM)

Each managed node has a chip set called the advanced system management processor. This is sometimes referred to as the ranger chip set. Working with the hardware instrumentation and xSeries systems management software, ASM processors are the key to problem notification and resolution. They provide the system administrator with complete remote management of a system, independent of the server status. The processors simulate a computer within a computer, keeping the server up and available for your critical applications.

Advanced Systems Management Adapter (ASMA)

The ASMA card (sometimes called a Wiseman card) is a peripheral computer interconnect (PCI) adapter which allows you to connect through a local area network (LAN) or modem from virtually anywhere for remote management. Up to 11 ASM processors are daisy chained and then connected to the ASMA card. Each ASMA adapter is externally powered, so it is not dependant on the status of the host machine. The ASMA card is then connected to the management network so that the compute nodes can be controlled and alerts forwarded.

For more information about ASM/SPN, refer to the *IBM Netfinity Advanced Systems Management* white paper, found at:

<http://www.pc.ibm.com/us/eserver/xseries/>

Keyboard, video, and mouse (KVM) and the crash cart

The x330 has a built-in capability for a KVM switch. To use this function, you simply daisy chain the unit's KVM ports and then connect them to a central Apex KVM switch. This allows for a high level of control in displaying the panel of one node by pressing a button on the front of the x330. The head nodes will also be connected directly into the KVM switch. Pressing the Print Screen key on the master console will bring up a menu and allow you to select the desired unit, either one of the head nodes or the entire daisy chain of x330s. If you select the daisy chain, you can then press the NumLock key twice, followed by the number of the desired unit on the daisy chain; then press the Enter key to select that unit.

In the event the KVM switch is not used, the next alternative is to use a crash cart. In this environment, the keyboard, mouse, and video display are kept on a cart with wheels. When required, the cart is wheeled over to the desired node and connected to the respective ports to provide KVM functions. The need of a crash cart has largely been obviated by the development of the x330 with its built-in KVM features.

3.1.4 Network

Networking in clusters usually needs high bandwidth, high speed, and low latency. The most demanding communication occurs between the compute nodes. This section presents the protocol and technologies used to build a cluster solution.

Fast Ethernet

Fast Ethernet and TCP/IP are the two standards largely used for networking. They are simple and cost effective, and the technology is always improving.

Fast Ethernet works well with a 200 MHz Intel Pentium Pro based machine used for a Beowulf node, and can operate at either *half-duplex*, or *full-duplex*. With full-duplex, data can be sent and received at the same time.

Full-duplex transmission is deployed either between the ports on two switches, between a computer and a switch port, or between two computers. Full-duplex requires a switch; a hub will not work.

Gigabit Ethernet

Gigabit Ethernet uses a modified version of the American National Standards Institute (ANSI) X3T11 Fibre Channel standard physical layer (FC-0) to achieve 1 Gigabit per second raw bandwidth.

Gigabit Ethernet supports multi and single mode optical fiber and short haul copper cabling. Fibre is ideal for connectivity between switches and servers and can reach a greater distance than copper.

Researchers have observed that the most significant overhead is the time required to communicate between CPUs, memory, and I/O subsystems that are directly connected to the network, and that this communication time may increase exponentially in a Beowulf cluster.

With the increase of PCI bus capability, the Gigabit Ethernet media is usable in a Beowulf cluster, as it can provide more bandwidth. At this point, however, the bottleneck becomes the protocol used for communication.

Some Parallel libraries, such as local area multicomputing (LAM) and MPICH, use the Transmission Control Protocol/User Datagram Protocol (TCP/UDP) socket interface for communicating messages between cluster nodes. Even though there has been a great effort to reduce the overhead incurred in processing the Transmission Control Protocol/Internet Protocol (TCP/IP) stacks, this is still one of the factors affecting cluster performance.

For more information on this subject, see *The Failure of TCP in High - Performance Computing Grid*, found at:

<http://www.sc2000.org/techpaper/papers/pap.pap174.pdf>

So there is a need to develop something new. Currently, for example, we use the Virtual Interface Architecture (VIA) used by Giganet.

Virtual Interface Architecture (VIA)

VIA is a mechanism that bypasses the layers of the protocol stack and avoids intermediate copying of data; these are the two major impediments to network performance during the sending and receiving of messages. It also allows for lower CPU utilization by the communication subsystem.

VIA is based on two interfaces:

- ▶ Software
- ▶ Hardware

VIA is portable between computing platforms and network interface cards (NIC).

A new version of MPICH libraries has been created that performs well on Giganet adapters using VIA (MVICH).

Myrinet and GM

Myrinet is a high-performance, packet communication and switching technology designed around simple, low-latency blocking switches. One limitation is that the switches are incrementally blocking.

Myrinet has a customized protocol. It provides a second processor that does much of the protocol work, avoiding interruptions to the primary processor.

Some of Myrinet's characteristics are:

- ▶ Full-duplex 2+2 Gigabit/second links, switch ports, and interface ports.
- ▶ Flow control, error control, and heartbeat continuity monitoring on every link.
- ▶ Low-latency, cut-through, crossbar switches, with monitoring for high-availability applications.
- ▶ The network capability to scale to tens of thousands of hosts, with network-bisection data rates in Terabits per second and alternative communication paths between hosts.
- ▶ Packets may be of any length, and thus can encapsulate other types of packets, including IP packets, without an adaptation layer. Each packet is identified by type, so that a Myrinet, like an Ethernet, may carry packets of many types or protocols concurrently.

GM is a message passing system for Myrinet networks and has replaced the MyriAPI software. The GM package is based on several components, such as drivers, API, and libraries.

Myricom's Myrinet software support is distributed as open source. Please refer to the Myricom web site <http://www.myri.com/> for further information.

3.2 Software

In order to build an operational cluster, you will need several software components, including an operating system, drivers, libraries, compilers, and management tools. This section presents the software products normally used to build an IBM *@server* xSeries Linux Cluster Solution.

3.2.1 Operating system

Linux is the UNIX-based operating system used primarily in a Beowulf cluster. One of the reasons for its success in Beowulf clusters is because the Beowulf software and drivers are in continuous evolution, hence the need for an open source operating system.

We use Red Hat 6.2 in our cluster because many of the libraries and tools have been written and tested based on that release.

Installation

The installation software is responsible for facilitating the install operation of the compute nodes. Our solution is based on using the Pre-Execution Environment (PXE) for remote machine booting, and Kickstart for the installation. It may also be possible to use ALICE (Automatic Linux Installation and Configuration Environment) instead of Kickstart in the future.

PXELinux

PXELinux is a program for booting off a network server using a boot PROM compatible with the Intel Pre-Execution Environment (PXE) specification.

It allows for a level of control of the boot process prior to actually loading a kernel. Using PXE, or more specifically, PXELinux, we can control the operation of the boot process from a management node. If no changes are required, we simply tell PXELinux to boot from the hard drive. However, if we wish to reload the node from scratch, we can tell the node to boot from the network. If we wish to rebuild one or all of the nodes, we can use Red Hat Kickstart to provide an automated installation of the nodes.

PXELinux is part of syslinux, which is available at:

`ftp.kernel.org`

Red Hat Kickstart

Kickstart is a system installation and customization program developed by Red Hat. It allows clients, such as compute nodes, to be completely installed remotely off a master server without the need for manual interaction. xCAT configures and invokes KickStart during the installation of compute nodes.

For more information on KickStart please go to:

<http://www.linuxdoc.org/HOWTO/KickStart-HOWTO.html>

Advanced Linux Installation and Configuration Environment (ALICE)

ALICE is a tool based on scripts and programs to customize installations files and operations. ALICE works in combination with YaST (Yet another Setup Tool), the SuSE installation and administration utility. YaST will partition and format the hard disk and then install the selected packages. Afterwards, ALICE takes control and modifies the target's system configurations files: Add users and groups and enable/disable services.

File systems

Another important issue regarding operating systems is the available file systems. Here we have to consider different types of file systems, each suitable for different scenarios. First, we look at traditional file systems and afterwards at some distributed file systems.

Local file systems

Most of the current Linux installations use the traditional Linux standard file system ext2 since it proves very reliable performance for most purposes.

There are performance issues in ext2 file systems when dealing with a large amount of small files. Furthermore, there is a demand for handling files larger than 2 GB that is not provided by the regular ext2 file system. There is a kernel patch for ext2 that enables it to support files larger than 2 GB, but it requires your application to be rebuilt against certain large file functions providing libraries.

What are the new file systems? Currently, there is only one, more or less enterprise-ready, file system available. It is called ReiserFS and is based on balanced trees, thus giving a huge performance increase for most scenarios. Additionally, it supports journaling and allows a faster restart after a system crash or power outage. It also supports file sizes larger than 2 GB.

Other new file systems addressing the same issues are JFS from IBM, XFS from SGI, and ext3, an enhanced version of ext2 that supports journaling. As all of those file systems are not yet available for daily production use, we will not cover them in great detail.

Network File System (NFS)

NFS is the best known network file system. Beowulf clusters almost always use the NFS protocol to provide a distributed file system. NFS uses a client/server architecture and Remote Procedure Call (RPC) calls to communicate.

The server exports files to a client that accesses them as if they are local. NFS does not save information about the client that made the request, so every client request is considered independently. Therefore, they must contain all the information necessary for execution each time they make a call to the server. This causes large messages and consumes a lot of network bandwidth. The advantage, however, of this system is that if the client crashes, the server does not hang, but continues working and satisfying all the other requests.

NFS has some well-known performance and scalability issues. Some of those were solved in NFSv3, a newer version and implementation of NFS, also available for Linux (Kernel 2.4.X). Nevertheless, there is increasing demand for better network file systems.

Andrew File System (AFS)

AFS was originally developed by Carnegie-Mellon University and was later turned into a commercial product of Transarc, now part of IBM. In 2000, IBM made its AFS enterprise file system product available to the open source community under the IBM Public License (IPL).

AFS is a very scalable, highly availability secure file system with a very robust data management model. Lacking a stable Linux AFS server, this created the need for a second platform, such as AIX, running the AFS server.

The use of AFS makes the most sense if you already have an AFS cell as part of your environment that you want to integrate into your cluster.

For further information regarding AFS please visit:

<http://www.transarc.ibm.com/>

Besides the IBM AFS client, there is another free AFS client implementation available for Linux. It is called the ARLA Project, and can be found at:

<http://www.stacken.kth.se/projekt/arla/>

Global File System (GFS)

GFS is a shared disk cluster file system for Linux. It is designed to support multi-client journaling and rapid recovery from client failures. All nodes are equal, and no server may be a bottleneck or a single point of failure.

The GFS cluster nodes physically share the same storage, Fibre Channel, or shared small computer system interface (SCSI) devices. The file system appears to be local on each node and GFS synchronizes file access across the cluster.

GFS is available from http://www.globalfile_system.org/ under the GNU Public License (GPL). Further documentation and information can also be found there.

General parallel file system (GPFS)

Capable of scaling to nine TB, GPFS is being ported to Linux, where it will be known as Blue Hammer.

At the center of Blue Hammer is the SP supercomputer's parallel system support program (PSSP) cluster management software and GPFS.

For more information on this issue, please refer to the following IBM Redbooks:

- ▶ *GPFS: A Parallel File System*, SG24-5165
- ▶ *Sizing and Tuning GPFS*, SG24-5610
- ▶ *PSSP 2.4 Technical Presentation*, SG24-5173
- ▶ *PSSP 3.1 Announcement*, SG24-5332
- ▶ *PSSP 3.2: RS/6000 SP Software Enhancements*, SG24-5673

Parallel virtual file system (PVFS)

PVFS is designed as a client-server system with multiple servers (called I/O daemons):

- ▶ I/O daemons typically run on separate nodes (I/O nodes) in the cluster.
- ▶ PVFS files are striped across the disks on the I/O nodes.
- ▶ Application processes interact with PVFS through a client library.
- ▶ The PVFS manager daemon handles operations such as permissions, checking for file creation, and open, close, and remove operations.
- ▶ The client library and the I/O daemons handle all file I/O without the intervention of the manager.
- ▶ The I/O daemons and the manager do not need to be run on different machines. In fact, you may achieve better performance by running them on the same machine.

The purpose of PVFS is to provide:

- ▶ High-speed access to file data for parallel applications
- ▶ Cluster-wide consistent name space
- ▶ User-controlled striping of data across disks on different I/O nodes
- ▶ Existing binaries that operate on PVFS files without needing to be recompiled

3.2.2 System management

Due to the large amount of management tasks that are executed in cluster systems, system management is one of the hotly discussed components. Although the current solution includes the use of xCAT, this section also provides a brief explanation about cluster system management (CSM) and IBM Director.

xSeries Cluster Administration Tool (xCAT)

xCAT was developed by Egan Ford as a means of simplifying the cluster installation process. It has proven to be a very effective tool for this purpose. It allows for a simplified approach to loading or reloading the nodes as well as facilities for managing a single machine or specified groups or ranges of machines. xCAT uses the special management capabilities of the x330s and x340s to provide a tremendous level of control of the nodes, including:

- ▶ Remote power control
- ▶ Remote hardware reset
- ▶ Remote software reset
- ▶ Remote OS console
- ▶ Remote POST/BIOS console
- ▶ Remote vitals (fan speed, temperature, etc.)
- ▶ Remote hardware event logs
- ▶ Remote hardware inventory
- ▶ Parallel remote shell
- ▶ Parallel ping
- ▶ Command line interface (no GUI)
- ▶ Single operations can be applied in parallel to multiple nodes
- ▶ Network installation (KickStart)
- ▶ Support for various user defined node types
- ▶ SNMP alert logging

Since it is written in shell scripts, it can be easily modified if the need arises. For more information about these tools, refer to Appendix A, “xCAT help” on page 137.

Attention: xCluster Administration Tools (xCAT) is a set of shell scripts that automate some processes. THE SCRIPTS ARE PROVIDED AS-IS. IBM HAS NO OBLIGATION TO PROVIDE ANY ERROR CORRECTION OR ENHANCEMENTS. FURTHER, IBM MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE SCRIPTS OR THEIR USE, NOR SHALL IBM HAVE ANY LIABILITY IN RESPECT TO ANY INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OF THIRD PARTIES DUE TO CUSTOMER'S OPERATION UNDER THE LICENSES OR RIGHTS HEREIN GRANTED.

Attention: Linux and other open source programs (OSP), when packaged with or preloaded by IBM on any computer system, are distributed and licensed to you by Caldera Inc., Red Hat Inc., SuSE GMBH, TurboLinux Inc., or other distributors of OSP, not IBM.

No express or implied patent, copyright, or other license is granted to you by IBM for the use of Linux or other OSP.

IBM delivers Linux and other OSP as-is without any warranty.

IBM is not responsible for any claim that Linux or other OSP infringe a third party's patent or copyright.

Cluster Systems Management (CSM)

Part of the IBM Linux clustering strategy involves the port of its PSSP software, formerly only available for the RS/6000 SP platform, to xSeries based Linux clusters. It is not actually a port, but an enhanced version based on the SP software, open source, WebSM 2000, xSeries development, and other technology sources.

An initial version of CSM will become available by the middle of 2001, and will contain the following features:

- ▶ **dsh**, **fping**, and hardware control commands
- ▶ Remote HW console
- ▶ Parallel network install (using LUI)
- ▶ Config file manager
- ▶ Event monitoring and resource management (RMC)
- ▶ Distributed automated event response (ERRM)
- ▶ Predefined event conditions and responses

- ▶ Distributed management services and database

By the end of 2001, there will be an enhanced version of CSM featuring better scalability and ease of use.

Time will tell whether CSM will replace xCAT entirely or both will exist in parallel, satisfying different customer demands.

IBM Director

IBM Director is a powerful, highly integrated, systems management software solution built upon industry standards and designed for ease of use. With an intuitive, Java-based graphical user interface (GUI), an administrator can easily manage individual or large groups of IBM and Wired for Management 2.0 compliant non-IBM PC-based servers, desktops, workstations, and notebooks on a variety of platforms.

IBM Director embraces systems management industry standards from the Distributed Management Task Force (DMTF), which include the Common Information Model (CIM), Web-Based Enterprise Management (WBEM), and the Extensible Markup Language (XML). It also supports the Simple Network Management Protocol (SNMP). This product includes key functions, such as robust group management from a central location, system discovery, hardware and software inventory, automated event responses, and monitoring and event alerting.

IBM Director can only be used for monitoring messages coming from the ASM. It cannot be used during the installation phase in the same way that xCAT can.

3.2.3 Development

From a development point of view, a High-Performance Computing (HPC) cluster solution differs from other types of systems. For example, an HPC cluster requires special software components. This section provides a brief explanation about compilers, libraries, and debuggers, all of which are required to build an HPC cluster solution.

Compilers

The Portland Group is the premier independent supplier of high performance scalar and parallel FORTRAN, C, and C++ compilers for High-Performance Computing (HPC) systems and Linux workstations, servers, and clusters.

With the IBM Linux Cluster, you receive Parallel HPF/F90/F77/C/C++ compilers, and debugging and profiling tools for 32-bit Linux. The terms allow for an unlimited user license for use within the confines of the cluster.

PGI key technologies include the following:

- ▶ Native OpenMP and auto-parallel F77/F90/C/C++
- ▶ Native High-Performance Fortran (HPF) compilers
- ▶ Legacy SGI-style parallel directives and pragmas
- ▶ Graphical debugging and profiling
- ▶ Pentium II/III optimizations including SSE and prefetch
- ▶ Function inlining
- ▶ Loop unrolling
- ▶ Cache tiling
- ▶ C++ class member templates
- ▶ Partial specialization and ordering
- ▶ Class member templates
- ▶ C++ STL and full interoperability with GNU C compiler (gcc), GNU database (gdb) and g77.

For more details on the Portland Group and its products and technologies, visit:

<http://www.pgroup.com/>

Libraries

The ScaLAPACK (or Scalable LAPACK) library includes a subset of LAPACK routines redesigned for distributed memory MIMD parallel computers. It is currently written in a Single-Program-Multiple-Data style using explicit message passing for interprocessor communication. It assumes matrices are laid out in a two-dimensional block cyclic decomposition.

Like LAPACK, the ScaLAPACK routines are based on block-partitioned algorithms in order to minimize the frequency of data movement between different levels of the memory hierarchy. (For such machines, the memory hierarchy includes the off-processor memory of other processors in addition to the hierarchy of registers, cache, and local memory on each processor.) The fundamental building blocks of the ScaLAPACK library are distributed memory versions Parallel Basic Linear Algebra Subroutines (PBLAS) of the Level 1, 2, and 3 Basic Linear Algebra Subroutines (BLAS), and a set of Basic Linear Algebra Communication Subprograms (BLACS) for communication tasks that arise frequently in parallel linear algebra computations. In the ScaLAPACK routines, all inter-processor communication occurs within the PBLAS and the BLACS.

For more information on ScaLAPACK, see:

http://www.netlib.org/scalapack/scalapack_home.html

ATLAS is an approach for the automatic generation and optimization of numerical software for processors with deep memory hierarchies and pipelined functional units. The production of such software for machines ranging from desktop workstations to embedded processors can be a tedious and time consuming task. ATLAS has been designed to automate most of this process.

We concentrate our efforts on the widely used linear algebra kernels called BLAS. In particular, our initial work is for general matrix multiply (DGEMM). However, much of the technology and approach developed here can be applied to the other Level 3 BLAS. The strategy can have an impact on basic linear algebra operations in general and may be extended to other important kernel operations, such as sparse operations.

For more information on ATLAS, refer to:

<http://www.netlib.org/atlas/>

Debugger

Among the tools available with the IBM Linux Cluster is a time-limited version of Etnus's industry-leading TotalView Multiprocess Debugger. TotalView V4.1 is a full-featured, source-level graphical debugger for C, C++, FORTRAN 77 and 90, assembler, and mixed source/assembler codes. TotalView has an intuitive, easy-to-learn interface with a user-friendly graphical user interface (GUI) that enables software developers to debug complex multiprocess, multithreaded, and clustered programs from a single session. For those times when a GUI is impractical, TotalView offers a command line interface (CLI) as well. The CLI also enables you to write debugging scripts and macros to automate tedious debugging tasks.

TotalView lets you temporarily insert compiled machine code fragments right into the program while you are debugging. Inserting compiled code fragments (and thereby performing hundreds or thousands of times faster) makes it less likely these changes will mask any synchronization problems. You can debug remote programs over the network because of TotalView's distributed architecture. This enables you to fix programs running on machines to which you do not have physical access. In addition, you can attach to running processes, so that you can debug processes that were not started under TotalView.

The special time-limited version of TotalView is available for unlimited use for 45 days from the installation of the license key.

For more information, see:

<http://www.etnus.com/Products/TotalView/index.html>

Parallel applications

Many applications in a cluster will be custom built and designed. However, there are many standard applications available to do standard tasks. Examples of these are:

- ▶ Blast: Typically used in genomics research to do pattern matching
- ▶ Star-CD: Typically used in fluid dynamics modeling
- ▶ LS-Dyna: Typically used in crash simulations

3.2.4 Resource management

To provide a mechanism to schedule and manage batch job workloads, the IBM Linux Cluster comes with the Portable Batch System (PBS) from MRJ Technology Solutions. PBS is a flexible batch software processing system developed at the NASA Ames Research Center. PBS operates on heterogeneous clusters of workstations, supercomputers, and massively parallel systems. The benefits of using PBS include increased utilization of costly resources, a unified interface to all computing resources, reduced burden on system administrators and operators, and an elevated quality of service to the user community. Other key features of PBS include:

- ▶ Portability: PBS complies with the POSIX 1003.2d standards for shells, utilities, and batch environments.
- ▶ Configurability: PBS is easy to configure to match the requirements of individual sites. The flexible job scheduler allows sites to establish their own scheduling policies for running jobs in both time-shared and space-shared (dedicated) environments.
- ▶ Adaptability: PBS is adaptable to a wide variety of administrative policies and provides an extensible authentication and security model.
- ▶ Expand ability: PBS supports the dynamic distribution of production workloads across wide area networks and the creation of a logical organization from physically separate entities.
- ▶ Flexibility: PBS supports both interactive and batch jobs.
- ▶ Usability: PBS provides a graphical user interface (GUI) for job submission, tracking, and administration.

For more information about PBS, see:

<http://www.pbspro.com>

Job management

There are many options for job resource management and scheduling. Here we describe the PBS resource manager and the Maui Scheduler; these are best suited to our clustering needs. Maui schedules and sorts jobs to best fit into available time and machine resources by using intelligent algorithms. These algorithms are applied to the data by PBS and PBS is informed when and where to start the actual jobs.

Maui

The Maui Scheduler is a software tool that provides administrators with control over and information about advanced job scheduling, workload throttling policies and features, node allocation policies, consumable resource handling, administration, analytical simulation to evaluate different allocation and prioritization schemes, node allocation policies, reservation policies, fairness, fairshare, job prioritization (high priority jobs run first, but avoid long-term starvation of low priority jobs), advance reservations, standing reservations, Quality of Service (QOS) management, QOS overview, QOS configuration, and statistics and logging.

The Maui Scheduler was mainly written by David Jackson for the Maui High Performance Computing Center, which was running IBM SPs and using IBM's LoadLeveler job management software.

Maui installation prerequisites include a pre-installed, configured, and functional resource manager; in this case, PBS. Maui can be thought of as a scheduler plug-in for resource managers.

You can download Maui and find very good online documentation at:

<http://www.supercluster.org>

Portable batch system (PBS)

PBS is a flexible batch software processing system developed at the NASA Ames Research Center. It operates on networked, multiplatform UNIX environments, including heterogeneous clusters of workstations, supercomputers, and massively parallel systems.

PBS is a leading workload management solution for Linux clusters. PBS maximizes efficient use of resources, enforces policies, and provides detailed usage and accounting data. The benefits of using PBS include increased utilization of compute resources, a unified interface to all computing resources, reduced burden on system administrators and operators, and an elevated quality of service to the user community. OpenPBS is a free source code release of PBS from Veridian Systems.

For more information, visit:

<http://www.OpenPBS.org/>

For the commercial release, which goes by the name of PBS Pro, visit:

<http://www.pbspro.com/>

Load sharing facility (LSF)

LSF (Load Sharing Facility) is a distributed queueing system from Platform Computing Corporation that unites a cluster of computers into a single virtual system to make better use of the resources on the network. It has the capability to automatically select hosts in a heterogeneous environment based on the current load conditions and the resource requirements of the applications. You can run jobs remotely as if they are jobs being run on the local host. LSF harnesses all available computing resources, including the cumulative processing power of workstations, to process jobs efficiently, complete workloads faster, and increase user productivity. It supports both sequential and parallel applications running as interactive and batch jobs. It is based on a suite of applications. For more information, please refer to:

<http://www.platform.com>



Solution guide

This chapter describes the considerations for planning and installing the cluster.

We talk about:

- ▶ General considerations
- ▶ Configuration aids
- ▶ Configuration schemes
- ▶ Verification activities to put in place
- ▶ What we used in the cluster in our lab

4.1 General considerations

It is important to thoroughly plan your cluster prior to assembly. Plan and build your cluster on paper and in your head, thinking through every possible aspect and problem. Walk through the location where the cluster will be placed and make sure you understand the placement. Discuss the cluster with facilities and make sure you will have plenty of space, power, and cooling, and that the floor can hold the cluster. Make sure you understand any facility issues with scheduling to allow sufficient time and to ensure you do not disrupt normal operations.

Next, think through the hardware and its configuration. Make sure you have discussed the cluster thoroughly with your network people. This should include security considerations, IP addresses, and the specific equipment that will be used. Also, consider what, if any, impact or expectations this will have on your production network.

Finally, review the software you expect to install on your cluster. If you are using a standard configuration, this should be fairly routine. However, it is very common to include additional software that may have specific version requirements. Make sure you have reviewed all the software needed, noted what specific versions might be needed and if there are any dependencies.

In Section 4.4, “Cluster questionnaire” on page 66, we have assembled all of the questions which you should be asking or thinking about for building the cluster. Appendix F, “Cluster questionnaire” on page 205 includes blank sheets where you can record your answers. If this is an IBM Cluster Solution, many or all of these questions will be asked during the Solutions Assurance Review process.

Environment

- ▶ Does your lab provide enough space for the whole cluster equipment?
- ▶ Does its floor support the equipment weight?
- ▶ Is there enough support for cable management (raised floor)?
- ▶ Does your environment supply enough power?
- ▶ Do you have enough power sockets supported by a matching fuse?
- ▶ Is the air conditioning able to keep the temperature in the allowed range?
- ▶ Did you think about security mechanism to control access to your cluster?

Hardware

- ▶ Did you come up with a color, naming, and labeling scheme?
- ▶ Do you have enough switches for supporting full bisection bandwidth?

- ▶ Do you plan to connect the cluster to the LAN?
- ▶ Do you have the right configuration of machines according to their usage?
- ▶ Did you choose the right kind or kinds of network to support your expected throughput and latency?
- ▶ Does your cluster design reflect your application design and requirements?

Software

- ▶ Have you check the general compatibility of the software with the hardware you are installing?
- ▶ Are all the necessary drivers for the components available as stable versions?
- ▶ Does the operating system support all the planned features?
- ▶ Are the tools and libraries supported by the operating system distribution?
- ▶ Did you check for version dependencies?

The answers to the above questions should be written down and kept through the whole cluster installation for reference.

Additionally, it is a good idea to visualize your cluster design; machines and virtual local area networks (VLANs) should be specified and highlighted with different colors for easy identification.

4.2 Configuration aids

In this section we will name and point to some tools to help you with the questions asked in Section 4.1, “General considerations” on page 58, especially concerning the hardware architecture.

4.2.1 Rack configurator

To help set up the hardware, a useful tool is the IBM *@server* xSeries Rack Configurator. As the name implies, this configurator helps you put together the rack design and place nodes and other equipment. It might even help you with the basic cabling required by the configured rack setup.

The Rack Configurator generates a lot of useful information. This includes a printable list of parts that you need, where everything goes in the rack, recommendations on clearance, ventilation, and other relevant details. It also gives you a chart detailing the configuration resources, including the total weight and the amount of power needed to run the rack. For an example of this, see Figure 4-1 on page 61.

The Rack Configurator can be downloaded from:

<http://www.pc.ibm.com/support>

4.2.2 PC configurator

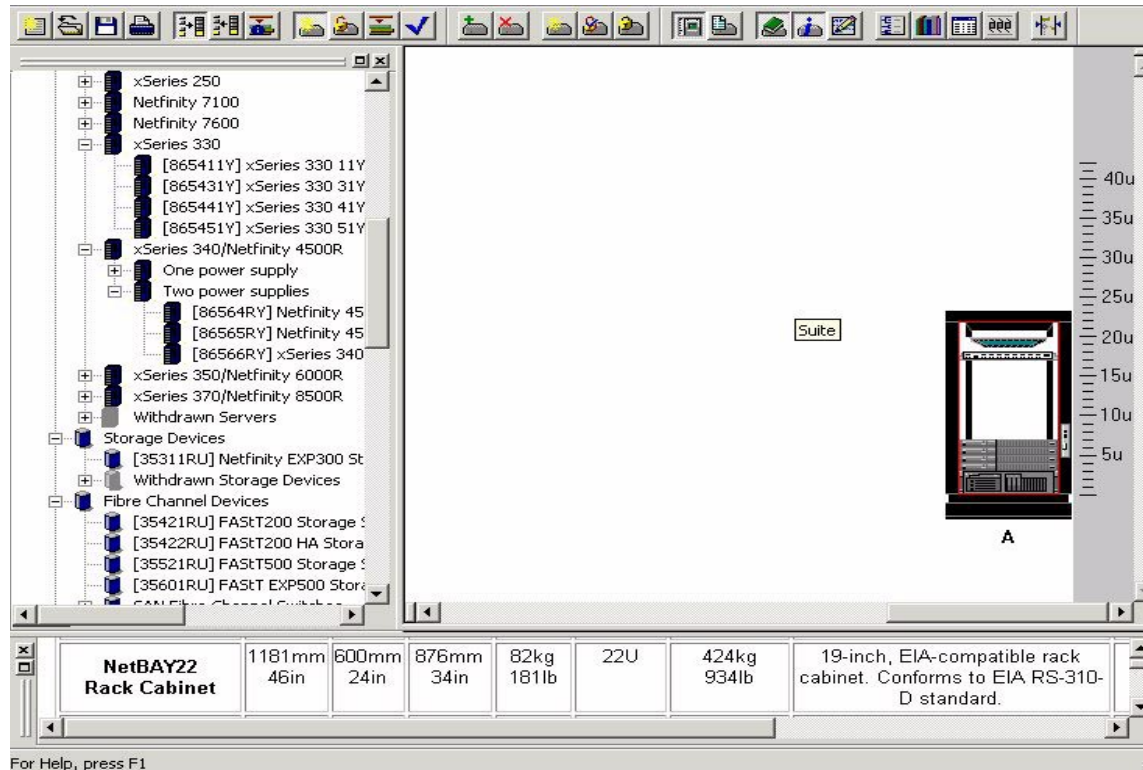
As the rack configurator helps you with your rack configuration, the PC configurator guides you through the configuration of individual boxes and racks. This configurator can validate configurations and finally generate reports to be used when ordering the equipment. This tool is a useful resource for pricing information. See Figure 4-2 on page 62.

You can download the PC configurator from:

<http://www.can.ibm.com/wwconfig/>

Please ensure that you always use up-to-date data files, which are also available from:

<http://www.can.ibm.com/wwconfig/>



For Help, press F1
 Figure 4-1 Rack configurator tool

4.2.3 Other useful resource and tips

Cabling

When putting your configuration together, do not forget to think about cabling. There are basically three issues:

- ▶ Cable length and volume (space and distance of components)
- ▶ Cable weight to be taken into account for environment check
- ▶ Cable management (raised floor or cable rails/tracks)

Compatibility

If you need to check hardware compatibility, for example, of racks and servers, go to:

<http://www.pc.ibm.com/us/compat/>

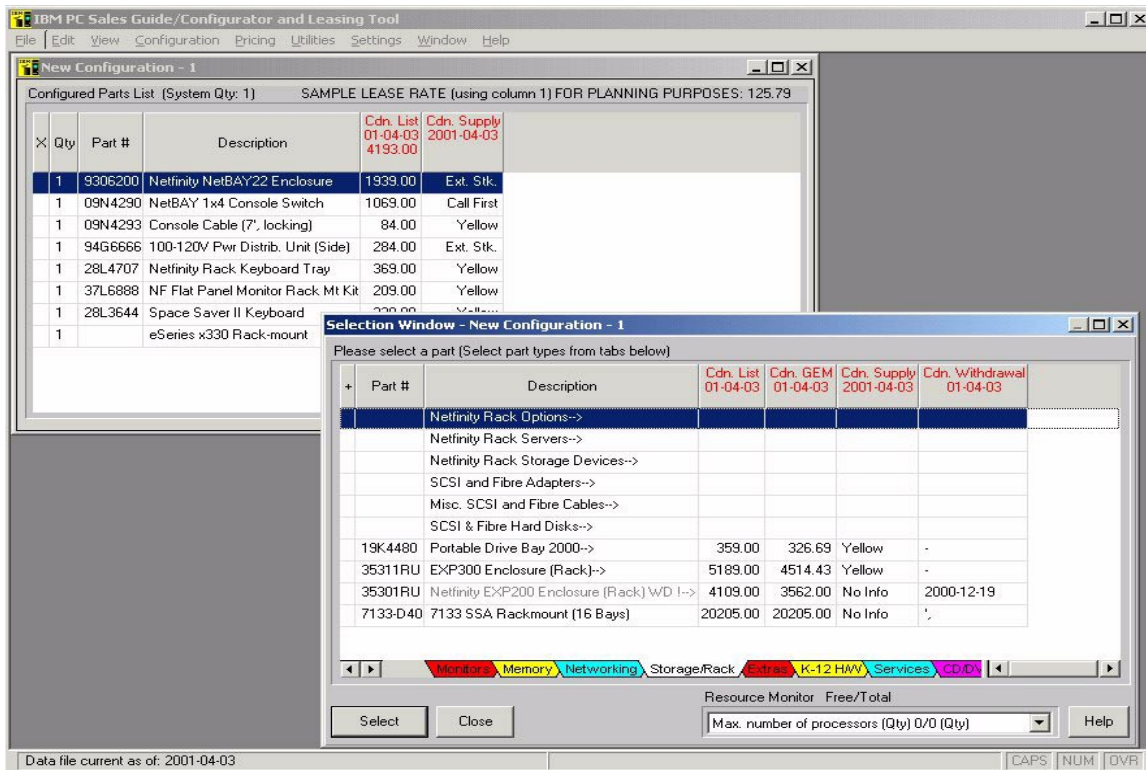


Figure 4-2 PC Configurator tool

4.3 Configuration schemes

Accounts

You need to set up user accounts on the user node. Use a scheme that helps to make the account names memorable.

Once the user accounts have been set up, the `/etc/passwd` file on the user node needs to be copied across to the control and compute nodes. This will enable other nodes to recognize the user when a job is submitted.

With kernel 2.2, there are some limitations regarding the maximum number of users, but these limitations have been overcome in kernel 2.4.

Naming conventions

Here is a list of the networks in the cluster:

- ▶ Cluster domain
 - cluster.ibm.com
- ▶ Cluster vlan
 - Network class C
 - 192.168.0/255.255.255.0
 - IPC (Myrinet)
 - Network class C
 - 192.168.1/255.255.255.0
- ▶ Management VLAN
 - Network class C
 - 192.168.2/255.255.255.0

Next, we look at the naming conventions for parts of the cluster:

- ▶ Node name convention
 - NodeX (X represents the node number)
- ▶ Myrinet name convention
 - MyriX (X represents the node number)
- ▶ Management name convention
 - ASMA
 - mgasmaX (X represents the ASMA card number)
 - Myrinet switch

- mgmyriX (X represents the Myrinet switch number)
- Equinox switch
 - mgelsX (X represents the Equinox switch number)
- Management node Ethernet card
 - mgtX (X represents the management machine number)

The following tables are examples of naming conventions in our Linux cluster. We strongly recommend that you adopt such a convention, as it is clear and will help you to keep track of everything.

Table 4-1 shows the standard localhost address.

Table 4-1 Localhost

| IP Address | Name |
|------------|---------------------------------|
| 127.0.0.1 | localhost.localdomain localhost |

Table 4-2 refers to the IP address for the management node. Notice that this is an IP address that connects beyond the cluster itself.

Table 4-2 eth0 on the management node

| IP Address | Name |
|-------------|--------------------------------|
| 10.12.42.56 | cluster.austin.ibm.com cluster |

Table 4-3 has the addresses for the individual nodes. These are local Class C addresses.

Table 4-3 eth1 on the management node

| IP Address | Name |
|-------------|------------------------------------|
| 192.168.0.1 | node1.cluster.austin.ibm.com node1 |
| 192.168.0.2 | node2.cluster.austin.ibm.com node2 |
| 192.168.0.3 | node3.cluster.austin.ibm.com node3 |
| 192.168.0.4 | node4.cluster.austin.ibm.com node4 |

Table 4-4 is also a set of Class C addresses for the Myrinet and for a different network.

Table 4-4 myri0 on the compute node

| IP Address | Name |
|-------------|------------------------------------|
| 192.168.1.1 | myri1.cluster.austin.ibm.com myri1 |

| IP Address | Name |
|-------------|------------------------------------|
| 192.168.1.2 | myri2.cluster.austin.ibm.com myri2 |
| 192.168.1.3 | myri3.cluster.austin.ibm.com myri3 |
| 192.168.1.4 | myri4.cluster.austin.ibm.com myri4 |

Finally, Table 4-5 gives a set of addresses for management functions, again on its own local network. This includes the settings for the ASMA adapter, the Myrinet switch, the Equinox switch, and the management ethernet adapter.

Table 4-5 eth2 on the management node (management only network)

| IP Address | Name |
|---------------|--|
| 192.168.2.1 | mgasma1.cluster.austin.ibm.com mgasma1 |
| 192.168.2.252 | mgmyri1.cluster.austin.ibm.com mgmyri1 |
| 192.168.2.253 | mgels1.cluster.austin.ibm.com mgels1 |
| 192.168.2.254 | mgt1.cluster.austin.ibm.com mgt1 |

4.4 Cluster questionnaire

This section will detail all of the questions that should be asked when building a cluster. The questions have been compiled from a variety of sources, including our Solutions Assurance Review process and from our own experiences in building other clusters. We have tried to group them in a logical manner. We suggest you make a copy or two of Appendix F, “Cluster questionnaire” on page 205 and record your answers. It is important that you discuss them thoroughly with the various groups involved.

Establishing expectations

This section discusses those issues which deal specifically with cluster expectations. The success or failure of a project is often not measured as much by achievements but by meeting expectations. These questions will try to establish those expectation:

- ▶ Have the cluster's performance requirements been documented? If so, please specify any required measurements.
- ▶ Has the cluster configuration been sized and verified to match the requirements?
 - Application type
 - Processor(s)
 - Memory
 - Cache
 - Disk Storage / RAID (local)
 - Tape (backup)
 - Adapters
- ▶ Have disk storage requirements and disk layout been established? In particular, this should apply to the cluster's need for connection(s) and the method of connection(s), to external data / storage.
- ▶ Will this cluster require high speed, low latency IPC (Myrinet)?
- ▶ What will the cluster be used for?
- ▶ What are your expectations for this cluster in 10 words or less?
- ▶ What cool name did you assign your cluster?
- ▶ Are you aware of any conditions or issues which could impair your ability to implement this cluster?

Environmental questions

This section discusses issues related to the environment where the equipment is to be installed. The questions should be answered before beginning the cluster assembly.

- ▶ Will the cluster be housed in a computer room and is there room?
- ▶ Will there be a raised floor? If so, will it provide sufficient room for cabling?
- ▶ Will the floor support the weight of the cluster?
- ▶ Is there, or will there be, sufficient power of the proper voltage and with the specified outlets?
- ▶ Is there adequate cooling for the additional equipment?
- ▶ Is there an established facilities schedule for making changes or adding/powering on equipment? If so, how will it effect implementation?

Hardware/configuration questions

This section discusses issues related to the hardware and configuration. The questions should be answered before beginning cluster installation.

- ▶ What hardware components besides Netfinity/eServer xSeries are included in the solution (RS/6000/pSeries, AS/400/iSeries, S390/zSeries, Storage Subsystems, and so on)?
- ▶ What are the IP address ranges (for compute nodes, head nodes, Ethernet switches, ASMA adapters, terminal servers, and so on)? Determine if a private address space will be used or if it will be a specified range provided by the Network group.
- ▶ What are your naming conventions (node, rack, management, and so on)? Do you start with zero or one? Are there any prefix or padding requirements? The naming *must* be prefix/suffix, where prefix is [A-Z a-z \-] and suffix is [0-9].
- ▶ Do you have any specific labeling requirements? This is similar to the question on naming conventions, but refers to specific labels used to identify equipment.
- ▶ Will you be using VLANs? If so, how many? How will they be partitioned?
- ▶ What security/authentication method do you require (global NIS, local NIS, local users only, rcp /etc/password, and so on)? We do not recommend using NIS on the compute nodes.
- ▶ How is external storage to be connected?
- ▶ What throughput is required from the cluster to your enterprise?
- ▶ Does your network group have any fixed schedule for making changes or adding equipment to the network? If so, how will it effect implementation?

- ▶ Does your network group have any specific equipment requirements for connectivity (brand, specific equipment, copper, fiber, and so on)?
- ▶ Are production level Linux software and drivers available for your special equipment/peripheral requirements?

Software questions

This section discusses issues related to the software and configuration. The questions should be answered before beginning the cluster installation.

- ▶ What other open source software will be installed?
- ▶ What commercial software will be installed?
- ▶ Do you have any existing Linux cluster(s) running the aforementioned applications?
- ▶ What do you use as a resource manager? It is assumed you will be using PBS, but are there others (Maui, LSF, and so on).
- ▶ Do you require special/commercial compilers (the IBM Linux Cluster provides the PGI compilers)?
- ▶ Does the operating system support all planned features and are any additional tools or libraries supported by the recommended level of the operating system (for example, Red Hat 6.2)?
- ▶ Are there any version dependencies for any additional software components?
- ▶ Is there a backup/recovery plan in place? Describe and define the backup requirements, frequency, and quantity.

4.5 Our cluster configuration

The cluster configuration that we use in our examples consists of one management node and four compute nodes. It will be the reference for all following examples regarding installation and setup.

The next two tables, Table 4-6 and Table 4-7 on page 70, list the hardware and software configurations of our cluster, respectively.

Table 4-6 Our cluster configuration

| How Many | What | Used as |
|----------|---|---|
| 3 | xSeries 330 equipped with: <ul style="list-style-type: none"> ▶ 2 CPU 866 MHz, 256k cache ▶ 1 18.2 GB Ultra 160 hard disk drive (HDD) ▶ 1 GB of 133 MHz error checking and correcting (ECC) static dynamic random access memory (SDRAM) random dual in-line memory module (RDIMM) Memory ▶ Myrinet LAN PCI Adapter | <ul style="list-style-type: none"> ▶ Compute nodes |
| 1 | xSeries 330 equipped with: <ul style="list-style-type: none"> ▶ 2 CPU 866 MHz, 256 KB cache ▶ 1 18.2 GB HDD ▶ 1 GB of 133 MHz ECC SDRAM RDIMM Memory ▶ Myrinet LAN PCI Adapter ▶ Advanced systems management adapter (ASMA), also known as a Wiseman PCI Adapter | <ul style="list-style-type: none"> ▶ Compute node ▶ Forward Service Processor alerts to management node |
| 1 | xSeries 340 equipped with: <ul style="list-style-type: none"> ▶ 2 CPU 866 MHz, 256 KB cache ▶ 3 18.2 GB HDD ▶ 1 GB of 133 MHz ECC SDRAM RDIMM Memory ▶ 1 ServeRAID 4L Adapter ▶ 1 xSeries 10/100 Ethernet PCI Adapter ▶ 1 xSeries Gigabit Ethernet SX Adapter ▶ 1 redundant power supply | <ul style="list-style-type: none"> ▶ Management node ▶ Head node ▶ Control node |

| How Many | What | Used as |
|----------|--|---|
| 1 | 16-port Equinox terminal server and related Transceiver and adapters | ▶ Serial Port Interconnect to allow access to the machines and return medium access control (MAC) address at the installation process |
| 1 | Myricom Myrinet switch | ▶ Compute node Interconnect, IPC |
| 1 | Extreme Networks, 48-port 10/100 Mb Ethernet Switch and 2 Gb uplink ports | ▶ Fast interconnect, IPC |
| 1 | xSeries NetBay 22 Rack, equipped with flatpanel and spacesaver keyboard, console switch, and PDU | ▶ Rack |

Table 4-7 Cluster software table

| What | Version |
|------------|--|
| Kernel | linux-2.2.18 |
| PXE-Linux | syslinux-1.53 |
| RPM | <ul style="list-style-type: none"> ▶ rpm-3.0.5-9.6x ▶ rpm-build-3.0.5-9.6x ▶ rpm-python-3.0.5-9.6x |
| Myrinet GM | gm-1.4 |
| MPICH | <ul style="list-style-type: none"> ▶ mpich-1.2..5 (GM) ▶ mpich-1.2.1 (TCP/IP) |
| Conserver | conserver-7.0.0 |
| OpenSSH | <ul style="list-style-type: none"> ▶ openssh-clients-2.5.2p2-1 ▶ openssh-2.5.2p2-1 ▶ openssh-server-2.5.2p2-1 ▶ rpm-devel-3.0.5-9.6x ▶ openssh-askpass-2.5.2p2-1 ▶ openssh-askpass-gnome-2.5.2p2-1 |
| POPT | popt-1.5-9.6x |
| ATFTP | atftp-0.3 |

| What | Version |
|-------------|--|
| PAPI | papi-1.1.5 |
| FPing | fping-2.3b1 |
| POV-Ray | <ul style="list-style-type: none">▶ v.3.1g of povuni_d.tgz▶ v.3.1g of povuni_s.tgz▶ mpi-povray-1.0.patch |



Hardware preparation

This chapter describes the basic hardware setup, configuration and customization. We discuss, in detail, how to install the equipment, add additional components like the Myrinet cards, and populate and cable the rack(s). After completing this chapter, you can start installing the software to get your cluster up and running. It is a good idea at this stage to have all of your planning details for your cluster at hand, so that you can implement the steps as you work through the chapter.

In this redbook, we presume a maximum number of 64 nodes (two racks); we give detailed examples for an eight node cluster. As already explained in the previous chapters, there are multiple options on how to design and implement the cluster. You might want to use only one large Myrinet switch and one large FastEthernet switch for the whole cluster or use individual interconnected switches for each rack. Whenever we talk about switches, you need to remember that there might only be one switch in your setup.

The topics covered in this chapter include:

- ▶ Preparing the node hardware
- ▶ Populating the rack(s)
- ▶ Connecting the cables
- ▶ Setting up the advanced systems management adapter (ASMA), if necessary

5.1 Node hardware installation

The first thing to do is to prepare each individual node before it goes into the rack. This means installing the usual additional options, such as a second CPU, memory, and hard disks.

Head node (x340)

We use the built-in Ethernet port for connecting to the customer network, so we add another FastEthernet card for the Service Processor Network (SPN) link. For better availability, a ServeRAID controller is installed. Finally, we need a Gigabit Ethernet adapter for the uplink to the FastEthernet switch to achieve a really fast compute node network installation.

For performance reasons, you may want to place the I/O intensive adapters (mainly ServeRAID and Gigabit Ethernet) into specific slots so that they are on different peripheral computer interface (PCI) busses. For details, please refer to your system documentation or the labels on the back of your server.

Compute nodes (x330)

We start by changing the serial port cable connection jumper from its default position COM port A/Mgmt to COM port B, as shown in Figure 5-1 and Figure 5-2 on page 75.



Figure 5-1 x330 as shipped (the serial port connection to be changed is circled)

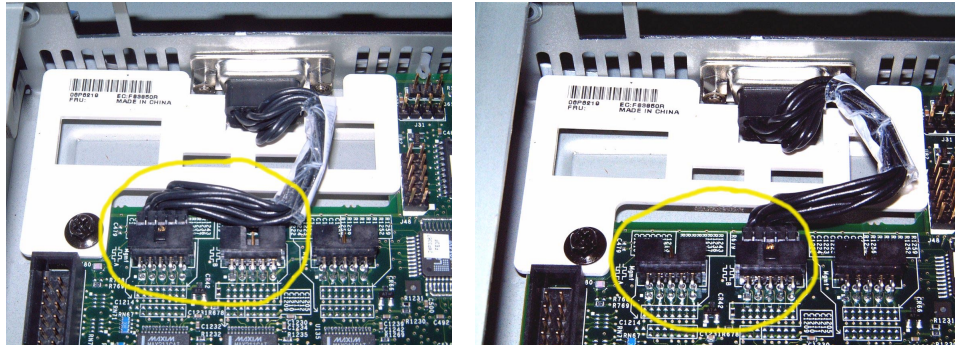


Figure 5-2 Serial port jumper before (port A) and after the change (port B)

We need to change the serial port connection because we plan to use a serial console redirection with the terminal server. Therefore, we need a reliable serial connection. With the default setup connecting to COM port A, we would share our connection with the systems management chip set, which would lead to lockups and time-outs related to its use of the port. By using COM port B for our serial connection, we do not interfere with the systems management functionality.

Once you have changed the serial port, you can install the Myrinet card into the short PCI slot. This is illustrated in Figure 5-3.



Figure 5-3 x330 with Myrinet card installed into short PCI slot

Finally, we have to install the Advanced Systems Management adapter (ASMA) card into the remaining long PCI slot (see Figure 5-4) for every eight (maximum of 11th) node.

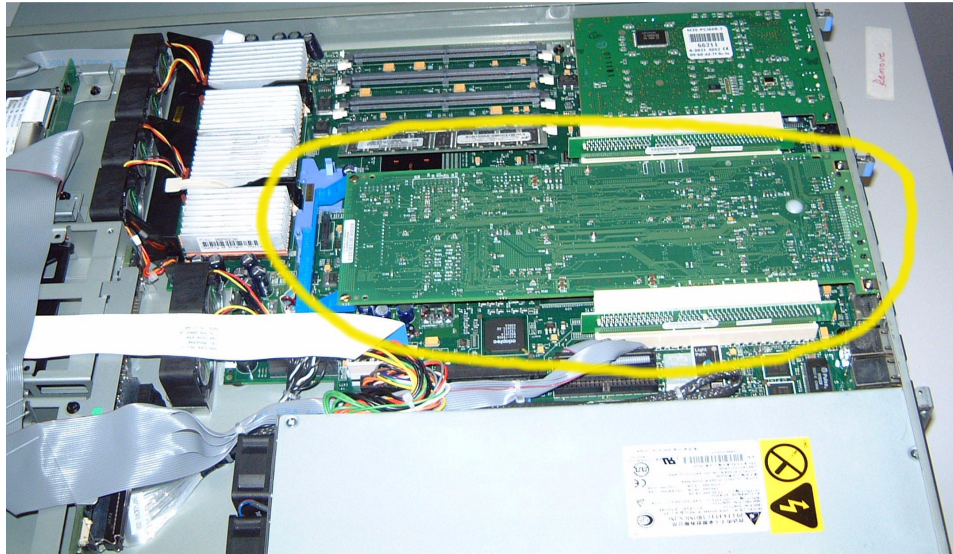


Figure 5-4 x330 with Myrinet and ASMA card installed

Flash updates

Now use the latest ServerGuide CD to update the BIOS of the motherboard and all the other components in all your nodes, such as the ServeRAID adapter, the ASMA card and so on. You can order or download the latest ServerGuide CD for your servers from:

<http://www.pc.ibm.com/support/>

There may be a newer version of the ASMA card firmware available on floppy disks. Please check for this as well.

After flashing your components, you have to restore the BIOS settings and ensure COM 2 is enabled, since it is connected to the external serial port. COM 2 should be on port 2F8 and IRQ3. Another good idea is to disable the virus warning/checking and the failed boot count option. Furthermore, you might want to disable NumLock for the management node, as it is most likely connected to a space-saver keyboard without separate NumBlock. Finally, you have to change the startup options to the following order:

- ▶ Floppy
- ▶ CD
- ▶ Network

- ▶ Disk

This ensures that your compute nodes can be installed and re-installed through the network without any flaws and need for interaction.

5.2 Populating the rack and cabling

Before placing the nodes into the racks, you should know the position of your additional equipment, such as network switches, monitor switches, terminal servers, and so on. It would be very helpful to have a plan for the rack that was derived from the Rack Configurator or other source for this step.

For most of the additional hardware, especially the switches, it is a good idea to place them in the middle of the rack, saving cable length and confusion. Heavy hardware, such as the management node, storage arrays, and the uninterruptable power supply (UPS) should be placed in the bottom. The power distribution units (PDUs) are installed on the right-hand side of the racks (seen from the front of the rack), as most power supplies are on this side, too. This leaves the left-hand side for the KVM (Keyboard, Video, and Mouse) switch.

Additionally, it is important to leave enough space for proper air flow and maintenance work.

Next, put your nodes into the rack and label them correctly (as illustrated in Figure 5-5 on page 78):

- ▶ One Node ID label on the front of the machine (for example, for x330s, on the CD-ROM drive)
- ▶ One Node ID label on the rear, preferably on the left-hand side (for x330s, you will need to label the mounting rail, as there is not enough space on the rear of the box)

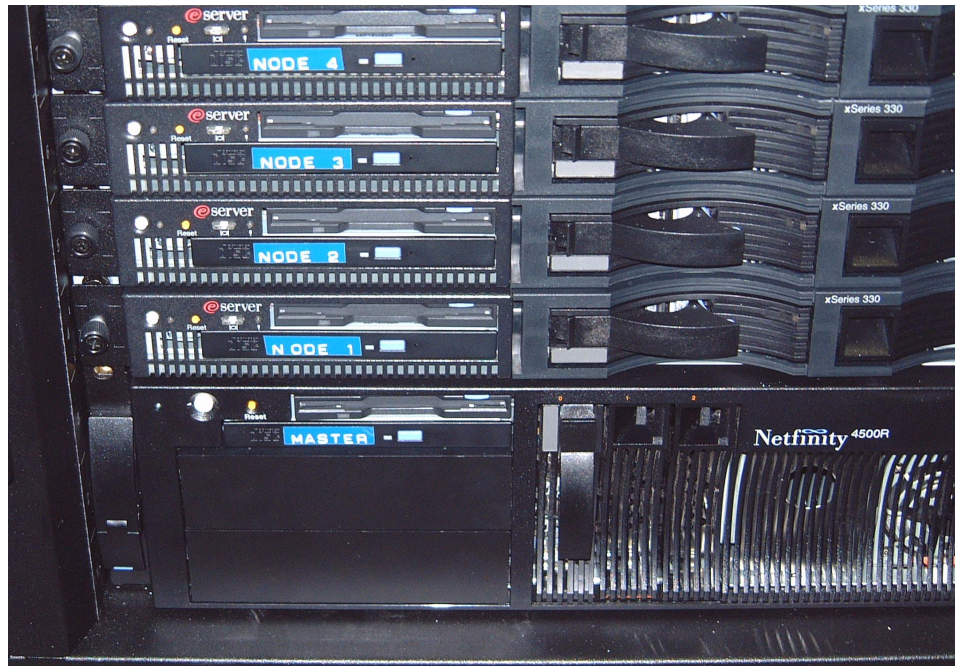


Figure 5-5 Node ID labeling on the front

Once you have finished labeling, you can start to cable the rack. This includes labeling each cable that is longer than 1 foot or 30 cm at each end and using different colors for different kinds of cables. Although the cable management of the x340 and the Netfinity racks (with rack extensions) is not demonstrated, we highly recommend that you use their cable management features to achieve less confusion regarding your cluster's cabling:

- ▶ Connect the power cables (including the external ASMA AC adapter and additional devices, for example, switches) to the PDUs or APC Master Switches (whichever your rack is equipped with).
- ▶ Connect your rack mounted monitor, mouse, and keyboard to the KVM switch of the rack.
- ▶ Connect the C2T cables from node to node, daisy chaining them, and connect the C2T KVM splitting cable to the last C2T OUT port, which is then connected to the KVM switch of your rack using the KVM (NetBAY console) cable.
- ▶ Connect the RS485 cables in daisy chains of eight nodes, plugging the last cable of each chain into the ASMA card dongle, which is directly connected to the ASMA card of that group.

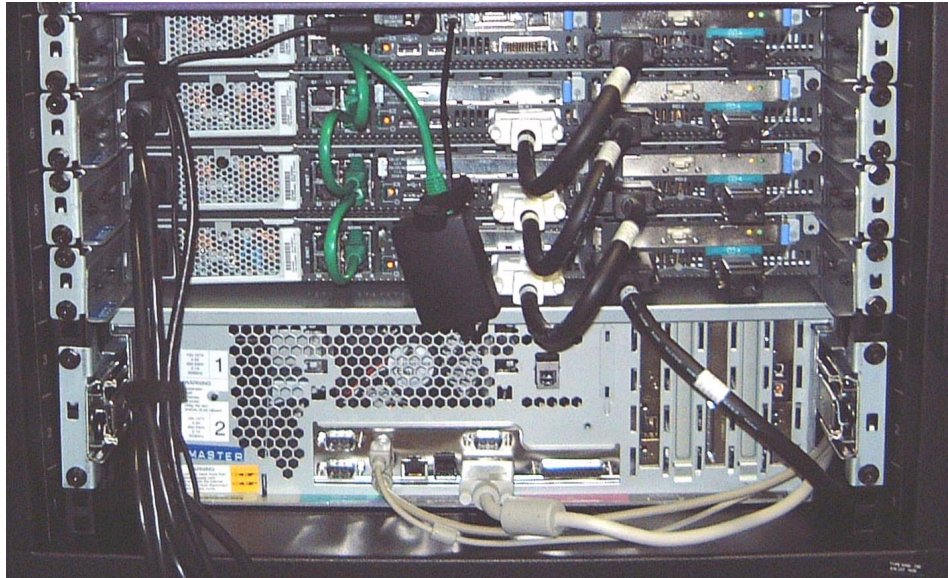


Figure 5-6 C2T SPN/ASMA cabling

Your rack should now look like Figure 5-6. After powering on the nodes (especially the ones with the ASMA cards inside), you can switch to their screens through C2T and the KVM switches:

- ▶ Connect the terminal servers (Equinox ELS) to the FastEthernet switches.
- ▶ Connect the ASMA card's Ethernet interfaces to the FastEthernet switches.
- ▶ Connect the nodes to the terminal servers (ELS) using the serial-to-CAT5 converters and CAT5 cables (see the left-hand side of Figure 5-7 on page 80).
- ▶ Connect the FastEthernet cables from the nodes to the switches and the FastEthernet cable from the Myrinet switch management port to the FastEthernet switch (see right side of Figure 5-7 on page 80).

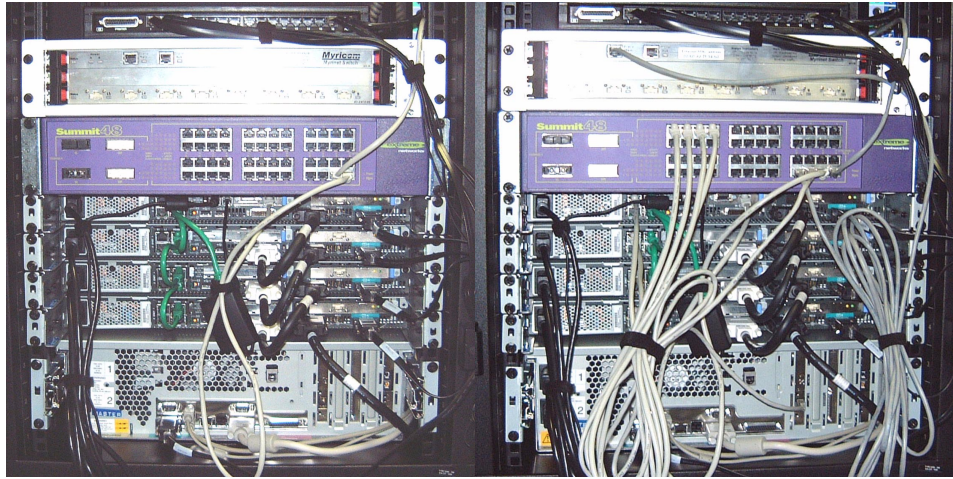


Figure 5-7 Terminal server cables (left) and “regular” FastEthernet cabling (right)

- ▶ Connect the Myrinet cables from the nodes to the switches.
- ▶ Connect the Gigabit Ethernet uplink from the management node and from any additional installation server nodes, if present, to the FastEthernet switches.

At this time, your rack should look like the one shown in Figure 5-8 on page 81. The following steps are optional and depend on your individual configuration and setup:

- ▶ Connect the nodes which need external network connections (user and management nodes) to the external networks.
- ▶ Interconnect the Myrinet switches from rack to rack (if applicable), using half of the ports of each switch to obtain full bisection bandwidth.
- ▶ Interconnect the FastEthernet switches from rack to rack (if applicable), preferably through Gigabit Ethernet.

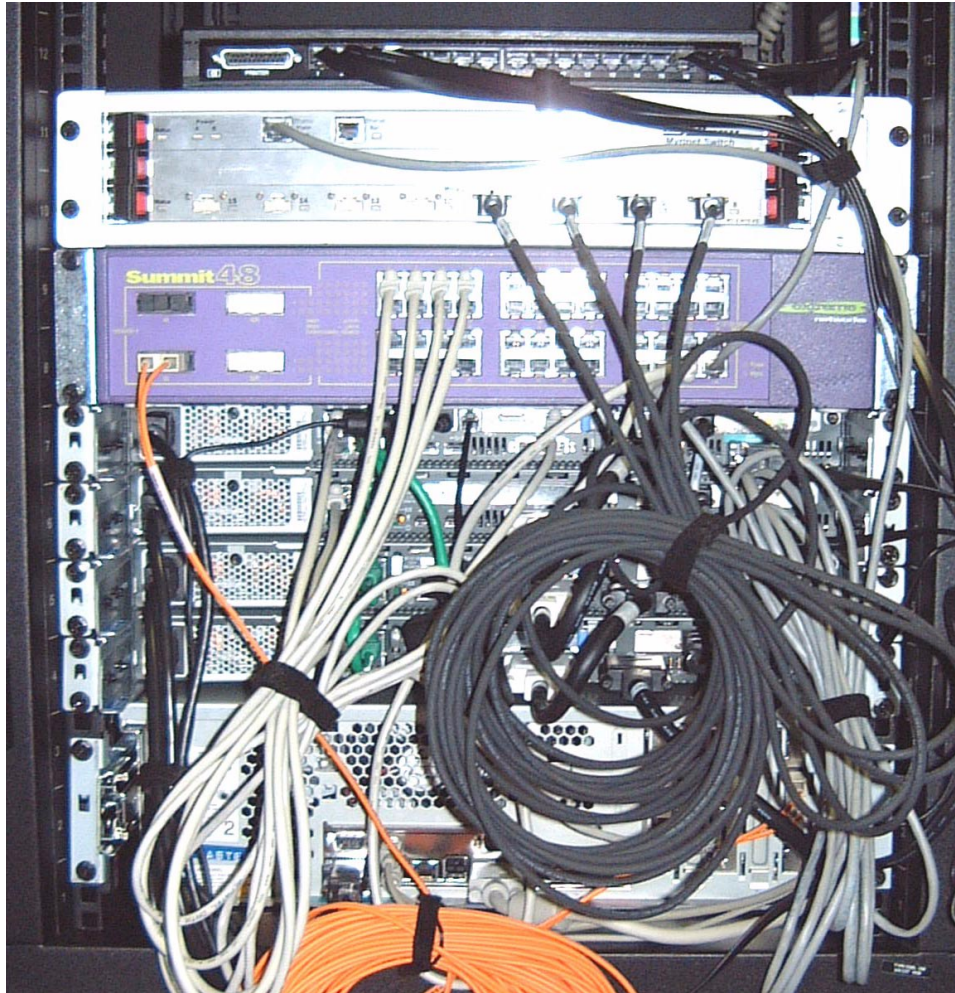


Figure 5-8 Rack cabled completely (including Myrinet)

Next, review the cabling. It is recommended that you find someone else to do this.

Finally, all the cables should be laid out neatly, keeping in mind that you might have to recheck and/or exchange cables later when setting up the cluster management software and applications due to cabling errors or faulty cables.

5.3 Cables in our cluster

The diagrams Figure 5-9 and Figure 5-10 on page 83 illustrate the back panel and cabling connections in the cluster used in the lab, configured with one x340 as head node and four x330 as compute nodes.

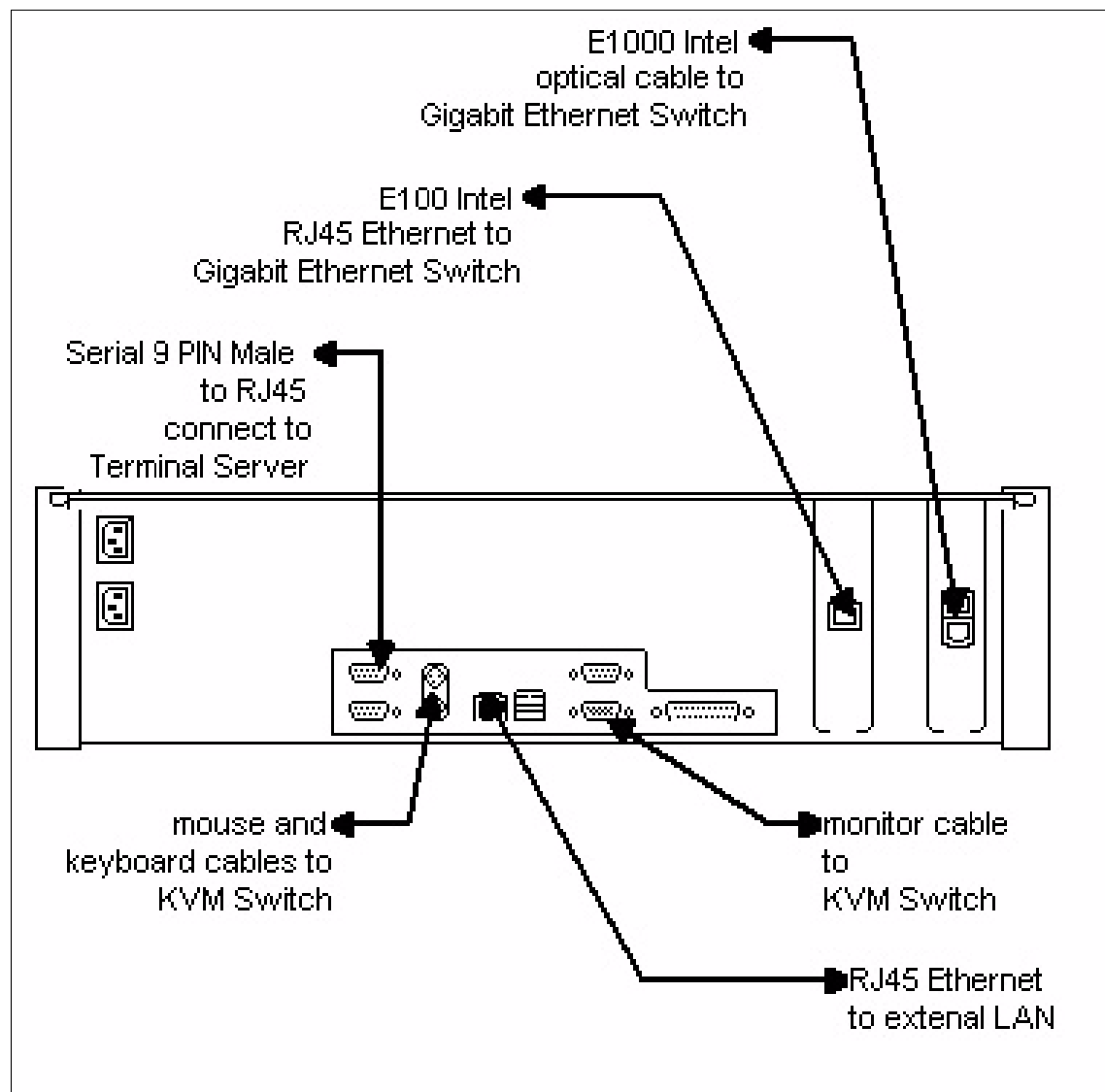


Figure 5-9 Cables on head node x340

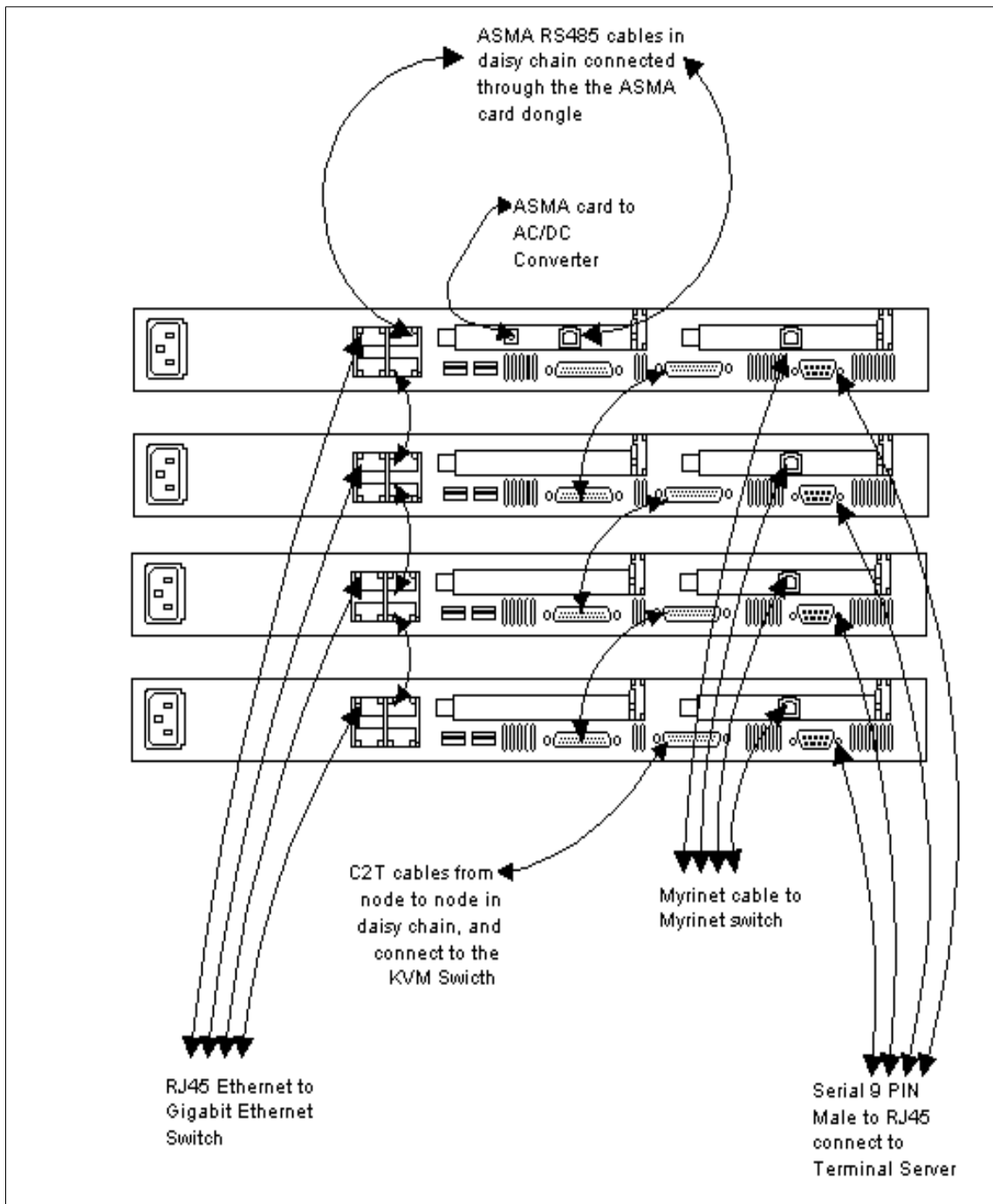


Figure 5-10 Cables on compute node x330

5.4 ASMA card setup

For the nodes that contain an ASMA card, you need to take the following steps, using the SPN setup floppy disk (the same as the firmware upgrade disk). Once you have booted from this floppy disk, select **Configuration Settings -> Systems Management Adapter** and apply the following changes:

- ▶ General Settings
 - Set name to mgasmaX (for example, mgasma1).
 - Set number to 1000 + X (for example, 1001 for the first node having an ASMA card inside).
 - Set systems management processor clock (time and date).
- ▶ SNMP Settings
 - Enable SNMP agent.
 - Enable SNMP traps.
 - Fill in system contact (for example, admin).
 - Fill in system location (for example, CR for computer room).
 - Fill in community name (for example, public).
 - Give community IP address (for example, the machine to send alerts to 192.168.2.254).
- ▶ Network Settings
 - Enable network interface.
 - Set host name to mgasmaX (for example, mgasma1).
 - Give local IP address for ASMA network interface (for example, 192.168.2.1).
 - Provide subnet mask (for example, 255.255.255.0).

Unless you need to integrate your Linux cluster into an already existing simple network management protocol (SNMP) environment, the contact, location, and community names are up to you. Otherwise, ask your SNMP administrator for the correct names that match your environment. In this case, you will be given a different and/or additional community IP address to enter for the ASMA card. Our default setup will just send all events through SNMP to the management node, which runs a simple SNMP catching daemon logging all SNMP events in `/var/log/messages`. For further details, please see Chapter 6, “Management node installation” on page 85.

Future versions of xCAT may include tools to automate the SPN and ASMA setup, requiring less manual interaction.



Management node installation

This chapter describes the management node installation process.

We talk about:

- ▶ RedHat installation
- ▶ System configuration
- ▶ Linux kernel customization
- ▶ xCAT download and installation
- ▶ Third party software download and installation for xCAT
 - conserver
 - fping
 - OpenSSH
- ▶ Network adapters configuration
 - Fast Ethernet
 - Gigabit
- ▶ Equinox setup

6.1 Before you start

A number of software packages will need to be downloaded before you start this installation. If you will be at a site where Internet access is slow or non-existent, it might be useful to download all the code and write it to a CD before you start.

Refer to Chapter 4, “Solution guide” on page 57 for more information about software packages.

6.2 Operating system installation

The initial installation of the management node is a simple Red Hat 6.2 CD-ROM installation; however, there are a few things that require special consideration:

ServeRAID driver For reasons of performance and/or reliability, the disks in the management node are often installed on a ServeRAID card. Unfortunately, the latest cards (4L and 4M) are not supported by the kernel on the Red Hat 6.2 CD-ROM, so a Red Hat driver floppy must be used.

File systems The management node stores a large amount of data as part of its administrative function, including a copy of the Red Hat CD-ROM and various logs. You must create at *least* the following file systems, and it is strongly recommended you create a /home file system if any users are to login to the management node:

| | |
|------------|-------------------|
| /boot | 50 MB |
| /install | 2 GB |
| /usr/local | 1 GB |
| /var | 1 GB per 128nodes |
| / | 2 GB |
| swap | 1 Gb |

Hardware clock The xSeries Advanced System Management hardware is not time zone aware; any alerts generated will be marked with the time from the hardware clock in the PC. This means that the hardware clocks in the compute nodes must be set to local time, not the more usual universal time coordinated (UTC). Since it is important that all the clocks in a cluster are kept in sync, the management node must also have its hardware clock set to local time.

kernel-smp package Due to a feature of the xSeries hardware, the Red Hat 6.2 installer does not automatically detect the second processor. In order to use a second processor, the

kernel-smp package must be manually specified as part of the install.

Tip: In our lab, we configured the ServerRAID using the ServerGuide 6.0.2A Setup and Installation CD provided by the IBM @server xSeries equipment (P/N 24P2785).

- ▶ Boot the ServerGuide CD.
- ▶ From the Main Menu, select:
 - Run setup programs and configure hardware
 - Red Hat Linux
- ▶ In the Configuration Method Selection, select:
 - Custom
- ▶ In the Custom Configuration Tasks, select:
 - Configure ServerRAID adapter
- ▶ Create the disk array in the ServerRAID Manager Window; we created a level 5 disk array.

6.2.1 Installing the Red Hat CD-ROM

The Red Hat installer includes support for automated installation using a process called Kickstart. Kickstart allows the questions that are asked as part of a normal install to be answered in advance, making the install completely automatic. This is the same process used by xCAT to install the nodes over the network.

We have created a Kickstart configuration file (known as ks.cfg) that can be used to install the management node so that it would be possible for you to exactly follow our installation procedure. If you wish to use this file to install the management node, download it from the IBM Redbooks additional materials FTP site at:

`ftp://www.redbooks.ibm.com/redbooks/SG246041/`

Kickstart configuration file

Before you use the ks.cfg file, you should review its contents. You may wish to modify a number of the settings according to your site requirements. At the very least, you should change the following:

network This stanza contains all the network configuration values for eth0. Modify this to suit your local environment based on the values decided upon earlier.

| | |
|-----------------|--|
| timezone | Set this based on your location. The values are composed of a path appended to /usr/share/zoneinfo. You can see all available values on an installed system. |
| rootpw | This is the password for the root user on the system. It is highly recommended you change this from the default. |

The ks.cfg we supply is designed to be copied onto the ServeRAID device driver disk. A self extracting executable (Windows only) of the disk image can be downloaded from:

<http://www.pc.ibm.com/support/>

To use this file, create the disk according to the README file and then copy the customized ks.cfg file onto it.

Use the following procedure to begin the install based on the ks.cfg:

- ▶ Create ServeRAID driver disk.
- ▶ Copy customized ks.cfg to floppy.
- ▶ Boot the system from a Red Hat 6.2 CD-ROM *without* the floppy in the drive.
- ▶ When you get the boot: prompt, insert the device driver floppy.
- ▶ Type in **linux ks=floppy** and press Enter.
- ▶ The install should start automatically. Remove the floppy when you see the Preparing to install dialog.

There is a small bug in the Red Hat Kickstart code that says something similar to: “a Kickstart installation will not proceed automatically when installing to a hard-disk with bad partitions.” This message usually occurs when information is being saved to a new disk. If this is the case, press Enter at the Initialize panel and the installation will proceed normally. Once the Preparing to install dialog box is displayed, the installation will proceed, and you can leave the system to install and reboot automatically to Linux.

6.2.2 System configuration

Now that the system has been installed, it must be configured. The first part of this configuration is similar to the configuration you might perform on any freshly installed Linux system, with an emphasis on security. All these configuration tasks must be performed as the *root* user.

Set root password

Only do this if you did not change the root password in the ks.cfg or if you would prefer a different password to that specified in the ks.cfg file.

Add login user

It is not possible to log in to Linux as root over the network, so at least one other user must be created for this purpose. It is often convenient to have more than one account if a number of people will be using the management node.

Create the user with Linuxconf or use the **useradd** command, as shown in Example 6-1.

Example 6-1 User creation with useradd command

```
[root@master /root]# useradd -c 'IBM User' ibm
[root@master /root]# passwd ibm
Changing password for user ibm
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Verify the network configuration

Kickstart should have setup your network for you, but it is advisable to verify this using **ping**. The simplest and most comprehensive test is to ping a *named* machine, preferably on a network that needs to go via the gateway, as shown in Example 6-2.

Example 6-2 Pinging a known machine through a gateway

```
[root@master /root]# ping -c 2 www.redbooks.ibm.com
PING www.redbooks.ibm.com (207.25.253.24) from 9.53.197.104 : 56(84) bytes of
data.
64 bytes from www.redbooks.ibm.com (207.25.253.24): icmp_seq=0 ttl=121
time=92.0 ms
64 bytes from www.redbooks.ibm.com (207.25.253.24): icmp_seq=0 ttl=121
time=92.0 ms
--- www.redbooks.ibm.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 92.0/95.9/99.9 ms
```

Verify your network settings if this test fails.

If the management node is to serve as a network gateway, you must enable IP forwarding. You should *not* enable this if you will have another gateway node in the cluster, as it may affect the security of your system. Open `/etc/sysctl.conf` in your favorite text editor and change the ipforwarding line to:

```
# net.ipv4.ip_forward = 1
```

Updating host table

You should already have decided on an addressing scheme, so it should be a simple matter to copy it into `/etc/hosts` using your favorite text editor. It is even possible to use `Linuxconf` for this task if you like, but it will take you much longer. Example 6-3 contains the contents of `/etc/hosts` from our four node cluster.

Example 6-3 /etc/hosts

```
127.0.0.1localhost.localdomain localhost

## eth0 on the mgmt node

9.53.197.104cluster.austin.ibm.com cluster

## eth1 on the mgmt node

192.168.0.1node1.cluster.austin.ibm.com node1
192.168.0.2node2.cluster.austin.ibm.com node2
192.168.0.3node3.cluster.austin.ibm.com node3
192.168.0.4node4.cluster.austin.ibm.com node4

192.168.0.254master.cluster.austin.ibm.com master

## myri0 on the compute nodes

192.168.1.1myri1.cluster.austin.ibm.com myri1
192.168.1.2myri2.cluster.austin.ibm.com myri2
192.168.1.3myri3.cluster.austin.ibm.com myri3
192.168.1.4myri4.cluster.austin.ibm.com myri4

## eth2 on the mgmt node (Management only network)

192.168.2.1mgasma1.cluster.austin.ibm.com mgasma1
192.168.2.252mgmyri1.cluster.austin.ibm.com mgmyri1
192.168.2.253mgels1.cluster.austin.ibm.com mgels1
192.168.2.254mgt1.cluster.austin.ibm.com mgt1
```

If your nodes are to have long and short names as in Example 6-3, be sure and specify them both in the file. If you wish, the domain you use can be fictitious and not visible outside the cluster.

Install Red Hat updates

Red Hat 6.2 has been around for some time and there have been a number of security fixes released for it. You may want to download the RPMs at <http://www.redhat.com/support/errata/> and install them on the management node. If you need the updates to be installed on the compute nodes as well, they can be copied to the post install directory later on.

Shut down all unnecessary services

Red Hat 6.2 enables a number of daemons that are not usually needed on the management node. For security and serviceability reasons, it is advisable to disable any daemons that are not actually needed. The following commands disable services that you would normally not need to run on a management node. You may choose to leave some of these services running, although we do not recommend it.

```
# /sbin/chkconfig apmd off
# /sbin/chkconfig autofs off
# /sbin/chkconfig httpd off
# /sbin/chkconfig identd off
# /sbin/chkconfig isdn off
# /sbin/chkconfig kudzu off
# /sbin/chkconfig lpd off
# /sbin/chkconfig pcmcia off
# /sbin/chkconfig reconfig off
# /etc/rc.d/rc `runlevel | cut -c3` <-- those are backticks
```

Note: In addition, you may need to make a careful analyze review of the services and proceed with the shutdown of the ones you do not need in your installation.

A large number of other services are started through the **inetd** command, all of which are unneeded. Use the following commands to disable those services by commenting out all the lines in the `inetd.conf` and then reloading it:

```
# cp /etc/inetd.conf /etc/inetd.conf.SAVE
# perl -pi -e 's/^\w/#$&/' /etc/inetd.conf
# /etc/rc.d/init.d/inet reload
```

Fix syslogd

The management node collects syslog information from all the other nodes in the cluster, but the default configuration for `syslogd` is to not listen for remote messages. To enable remote logging, `syslogd` must be started with the `-r` switch. Modify the daemon line in `/etc/rc.d/init.d/syslog` accordingly, as in Example 6-4.

Example 6-4 Extract from modified `/etc/rc.d/init.d/syslog`

```
...
start)
    echo -n "Starting system logger: "
    # we don't want the MARK ticks
    daemon syslogd -m 0 -r
...

```

To enable this change, restart the syslog daemon:

```
# /etc/rc.d/init.d/syslog restart
```

Copy Red Hat CD-ROM

Finally, copy the contents of the Red Hat 6.2 CD-ROM to the management node for part of the compute node installation process. You should have made plenty of space in `/install` for this, so just copy the files using the following commands:

```
# mkdir -p /install/rh62
# mount /mnt/cdrom
# tar cf - -C /mnt/cdrom . | tar xvf - -C /install/rh62
# umount /mnt/cdrom
```

6.3 xCAT installation

For licensing reasons, all non-IBM software are not part of the xCAT package. To install a fully working xCAT management node, you must install the xCAT software and then patch and/or install a number of standard open-source packages that it depends on. The first stage is to obtain xCAT and use the included patches and scripts to customize the Red Hat environment.

Download the xCAT install package from the Redbooks additional materials FTP site at:

```
ftp://www.redbooks.ibm.com/redbooks/SG246041/
```

Unpack it into `/usr/local/xcat` using the `tar` command:

```
# tar xzvf xCAT-distribution.tgz -C /usr/local
```

For the network installation, xCAT needs the *kernel* (`vmlinuz` file) and *initrd* (`initrd.img` file) from the Red Hat network boot floppy image. The `bootnet.img` file contains the boot floppy image. Copy these to `/usr/local/xcat/ks62` using the command:

```
# mkdir /mnt/loop
# mount -o loop,ro /install/rh62/images/bootnet.img /mnt/loop
# cp /mnt/loop/vmlinuz /usr/local/xcat/ks62/
# cp /mnt/loop/initrd.img /usr/local/xcat/ks62/initrd-network.img
# umount /mnt/loop
```

For the medium access control (MAC) collection and advanced system management (ASM) setup, xCAT uses a regular Red Hat kernel. Copy it to `/usr/local/xcat/stage/` using the command:

```
# cp /boot/vmlinuz-2.2.14-5.0 /usr/local/xcat/stage/vmlinuz
```

Once the archive has been unpacked, there are a few extra installation tasks to perform. Copy the `xcat.sh` file to `/etc/profile.d` directory. This will add `/usr/local/xcat/bin` and `/usr/local/xcat/sbin` to `PATH`. Use the command:

```
# cp /usr/local/xcat/samples/xcat.sh /etc/profile.d/  
# chmod 755 /etc/profile.d/xcat.sh
```

Next, patch the Red Hat Kickstart files that you copied from the CD-ROM. This fixes a number of issues with the installation, most importantly, the problem with brand new disks. Use the command:

```
# patch -p0 -d /install/rh62 < /usr/local/xcat/build/rh62/ks62.patch
```

To enable the disk fix, you must now compile and install the fixed version of `newtpyfsedit.so`. Use the command:

```
# cd /install/rh62/misc/src/anaconda/libfdisk  
# make newtpyfsedit.so  
# cp -f newtpyfsedit.so (DO NOT BREAK LINE!)  
/install/rh62/RedHat/instimage/usr/lib/python1.5/site-packages/
```

Copy the xCAT system init scripts for the local system and post installation files for the installed nodes using the command:

```
# cp -pf /usr/local/xcat/sys/* /etc/rc.d/init.d/  
# cp -r /usr/local/xcat/post /install/
```

Activate the SNMP manager and enable some TCP performance enhancements with the following commands:

```
# /sbin/chkconfig snmptrapd on  
# /usr/local/xcat/sbin/updaterclocal
```

Finally, if you downloaded any Red Hat updates, copy them to `/install/post/rpms` so they can be installed on the nodes.

6.4 Additional software installation

Although you have now configured all the Red Hat components required by Red Hat, there are still a number of other packages that must be installed and configured on the management node.

6.4.1 The Linux kernel

For a number of reasons, it is necessary to install a custom kernel on all the nodes in the cluster, including the management node. It is recommended that the same kernel be used on both the management nodes and the compute nodes, although this is not mandatory. Instructions on compiling a kernel are beyond the

scope of this redbook, so if you do not know how to do this task, it is important you read and understand the Kernel-HOWTO before you start. The Kernel-HOWTO and many other useful documents can be found in the Linux Documentation Project homepage at:

<http://www.linuxdoc.org/>

The Linux kernel can be downloaded from:

<ftp.kernel.org>

If you know what you are doing, you might want to upgrade to a later kernel. Note that the kstmp files in `/usr/local/xcat/ks62` refer to the 2.2.18smpx kernel and will need to be modified. This is only recommended for experts.

To generate a 2.2.18smpx kernel from a stock 2.2.18, you should download the 2.2.18 kernel source from

<ftp://ftp.kernel.org/pub/linux/kernel/v2.2/linux-2.2.18.tar.bz2> and unpack it to `/usr/src` using the following commands:

```
# rm -f /usr/src/linux
# tar xIvf linux-2.2.18.tar.bz2 -C /usr/src/
# cd /usr/src/
# mv linux linux-2.2.18smpx
# ln -s linux-2.2.18smpx linux
```

You are now ready to patch the kernel. The following patches must be installed:

- ▶ Transmission Control Protocol (TCP) Short Messages
- ▶ MagicSysReq Over Serial
- ▶ Perfctr (PAPI)

Note: If you do not know how to do this, consult the Kernel-HOWTO and any documentation that comes with the patches.

TCP short messages patch

Download and install the patch. The patch file can found at:

<http://www.icas.edu/coral/LinuxTCP2.html>

Tip: In our lab, we used the following commands to apply the patch:

```
(download the file tcp-patch-for-2.2.17-14.diff on /tmp)
# cd /usr/src/linux
# patch -p1 < /tmp/tcp-patch-for-2.2.17-14.diff
(it is OK if you receive Makefile fail message, such as: Hunk #1 FAILED)
```

MagicSysReq over serial patch

Download and install the patch. The patch file can be found at:

<ftp://ftp.cistron.nl/pub/people/miquels/kernel/v2.2/>

Tip: In our lab, we used the following commands to apply the patch:

```
(download the file linux-2.2.19-sercons-sysrq-patch /tmp)
# cd /usr/src/linux
# patch -p1 < /tmp/linux-2.2.19-sercons-sysrq-patch
```

The Linux kernel can be compiled to include support for various different hardware configurations. This is managed through the `/usr/src/linux/.config` file. A sample of the `.config` file for kernel 2.2.18 is included in the `/usr/local/xcat/build/kernel/2.2.18` directory. It is strongly recommended that you use this to build your kernel. The following patch installation procedure will modify this file, so it is necessary to copy the file before installing the patch:

```
# cp /usr/local/xcat/build/kernel/2.2.18/DOTconfig /usr/src/linux/.config
```

Perfctr (PAPI) patch

Download and install the patch. The patch file can be found at:

<http://www.csd.uu.se/~mikpe/linux/perfctr/>

Tip: In our lab, we used the following commands to install the patch:

```
(download the file papi-1.1.5.tar.gz to /tmp)
# cd /tmp
# tar zxvf papi-1.1.5.tar.gz
NOTE: The install file in /tmp/papi/src/perfctr provides good instructions.
# cd /usr/src/linux
# patch -p1 < /tmp/papi/src/perfctr/patches/patch-kernel-2.2.18pre22
# cd /tmp/papi/src/perfctr/linux
# tar cf - . | tar -C /usr/src/linux -xvf -
# cd /usr/src/linux
# make menuconfig
Select "General Setup"
Select "Performance monitoring counters support" (by pressing <Y>)
Exit
Save your new kernel configuration
# mknod /dev/perfctr c 10 182
# chmod 644 /dev/perfctr
# cd /tmp/papi/src/perfctr
# ./configure
enter /usr/src/linux/include
# make
```

Make the following changes by hand:

- ▶ It has been our experience that NR_TASKS in `/usr/src/linux/include/linux/tasks.h` should be changed to 4000. This is the max number of processes in the system. For large clusters running large jobs (for example, 1024 processor jobs), you need to start 1024 ssh sessions from a single node. With the default of 512 (only 256 for users), you will never launch a large job. Use the following command to make the changes:

```
#define NR_TASKS 4000
```

- ▶ To improve TCP performance over Myrinet change MAX_WINDOW to 262144 in `/usr/src/linux/include/net/tcp.h` with the command:

```
#define MAX_WINDOW 262144
```

- ▶ At the top of `/usr/src/linux/Makefile`, change EXTRAVERSION to smpx:
EXTRAVERSION = smpx

Building the kernel

The kernel is now ready to be compiled. The build procedure is outlined below; however, if you do not understand it, please refer to the Kernel-HOWTO. Start by using the command:

```
# cd /usr/src/linux
# make oldconfig
# make dep clean bzImage modules
```

Once you have successfully compiled your kernel, you should install the modules, the kernel, and map, and create a ramdisk image with the command:

```
# cd /usr/src/linux
# make modules_install
# cp arch/i386/boot/bzImage /boot/bzImage-2.2.18smpx
# cp System.map /boot/System.map-2.2.18smpx
# cd /boot
# mkinitrd initrd-2.2.18smpx.img 2.2.18smpx
# ln -sf bzImage-2.2.18smpx bzImage
# ln -sf System.map-2.2.18smpx System.map
# ln -sf initrd-2.2.18smpx.img initrd.img
```

Next, add the kernel stanza to `/etc/lilo.conf` and change the default parameter (see Example 6-5). Be sure and check that the `root=` stanza matches the other entries in `lilo.conf`. Be sure you know what you are doing before you change this file, and back it up before you start.

Example 6-5 Extract from `/etc/lilo.conf`

```
default=linux-smpx
```

```
image=/boot/bzImage
    initrd=/boot/initrd.img
```

```
label=linux-smpx
read-only
root=/dev/sda6
```

When you have changed `/etc/lilo.conf`, you must actually install the lilo loader by running the `lilo` command, as shown in Example 6-6.

Example 6-6 Installing the lilo loader with the lilo command

```
[root@master /root]# /sbin/lilo
Added linux-smpx *
Added linux-smp
Added linux
[root@master /root]#
```

Note that the asterisk (*) indicates the default kernel and should match the `label=` line you just added to `lilo.conf`.

Install Intel Ethernet drivers

To complete your kernel installation, download and compile the Intel Ethernet drivers. They can be downloaded from:

<http://support.intel.com/support/network/>

Follow the compilation instructions provided by the site.

Tip: In our lab, we used the following commands to install the drivers for the Fast Ethernet adapter and Gigabit adapter:

```
(installing Fast Ethernet)
# mkdir -p /usr/local/e100
# cd /usr/local/e100
(download the file e100-1.5.5a.tar.gz)
# tar zxvf e100-1.5.5a.tar.gz
# cd e100-1.5.5/src
# make install
(installing Gigabit adapter)
# mkdir -p /usr/local/e1000
# cd /usr/local/e1000
(download the file e1000-3.0.7.tar.gz)
# tar zxvf e1000-3.0.7.tar.gz
# cd e1000-3.0.7/src
# make install (take all the defaults)
```

Note: In order to verify the successful installation of the e100 and e1000 drivers, we need to *reboot* the system and probe the drivers by issuing the following commands:

(the modprobe command will attempt to load the e100 driver for the Fast Ethernet adapter - if the installation does not succeed, you will receive an error here)

```
# modprobe e100
```

(verify that the file e100 exists)

```
# ls /lib/modules/2.2.18smpx/net/e100.o
```

(the modprobe command will attempt to load the e1000 driver for the Gigabit adapter - if the installation does not succeed, you will receive an error here)

```
# modprobe e1000
```

(verify that the file e1000 exists)

```
# ls /lib/modules/2.2.18smpx/net/e1000.o
```

Remaining interface configuration

Before you reboot, you should configure your remaining ethernet adapters. You can add configurations for eth1 and eth2 through linuxconf or perform the updates manually. When selecting a driver for the 10/100 card, remember to use the Intel e100 driver and not the kernel eeepro100 driver.

The configured files used in our cluster lab are shown in Example 6-7, Example 6-8, and Example 6-9.

Example 6-7 /etc/conf.modules

```
alias scsi_hostadapter aic7xxx
```

```
alias scsi_hostadapter1 ips
```

```
alias parport_lowlevel parport_pc
```

```
alias eth0 pcnet32
```

```
alias eth1 e1000
```

```
alias eth2 e100
```

Example 6-8 /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=eth1
```

```
BOOTPROTO=static
```

```
ONBOOT=yes
```

```
IPADDR=192.168.0.254
```

```
NETMASK=255.255.255.0
```

Example 6-9 /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE=eth2
```



```
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.2.254
NETMASK=255.255.255.0
```

Your kernel installation and configuration should now be complete. Reboot the management node, making sure the system boots up without any problem. If a problem occurs, do not go to the next step before fixing it. Be sure to check that all the network interfaces came up OK by using **ping**. Use the command:

```
# shutdown -r now
```

Finally, you must **tar** the kernel for future installation onto the nodes. Use the following command:

Note: In the lab we found that the item with wildcard had to be first.

Note: The following is one long line and should not be broken:

```
# cd /install/post/
# tar czvf kernel.tgz /boot/*2.2.18smpx* /boot/bzImage /boot/System.map
/boot/initrd.img /lib/modules/2.2.18smpx /usr/src/linux
/usr/src/linux-2.2.18smpx
```

6.4.2 h2n

h2n is one of the examples from *DNS and Bind*, by Albitz, et al.. h2n automatically generates nameserver configuration files from `/etc/hosts` and is used by the xCAT `makedns` program. Download the book's examples from `ftp://ftp.ora.com/pub/examples/nutshell/dnsbind/` and extract the h2n script to `/usr/local/xcat/lib`. For example:

```
# tar xzvf dns.tar.Z -C /usr/local/xcat/lib h2n
# chmod 755 /usr/local/xcat/lib/h2n
```

6.4.3 PXELinux

To start network installation, xCAT uses the network boot support of the onboard ethernet in the x330s. This network boot support relies on a vendor independent API known as the Pre-Execution Environment or PXE. More information on PXE can be found at:

<http://developer.intel.com/>

PXE relies on fetching a small boot loader from the remote server through `tftp`. The current version of xCAT uses the PXELinux boot loader, which is part of the SysLinux package, and can be downloaded from:

<ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/>

It is important to use at least Version 1.53, since xCAT relies on a new feature introduced in this version. Although it is possible to rebuild PXELinux, this is not recommended. All that is required by xCAT is the pxelinux.0 file, which you can get with the command:

```
# tar xzvf syslinux-1.53.tar.gz
# cp syslinux-1.53/pxelinux.0 /usr/local/xcat/tftp/
# chmod 444 /usr/local/xcat/tftp/pxelinux.0
```

6.4.4 atftp

To support the TFTP for PXE, a tftp server must be installed on the management node. There are a number of different tftp servers available for Linux, including the one that comes on the Red Hat 6.2 CD-ROM. Unfortunately, many of these suffer from a number of issues, scalability being a particular concern. We have found the most reliable to be atftp which can be obtained from:

```
ftp://ftp.mamalinux.com/pub/atftp/
```

The atftp package comes with no installation instructions, but the installation is fairly straightforward. Use the command:

```
# tar xzvf atftp-0.3.tar.gz
# cd atftp-0.3
# make
# make install
```

Note that the **make install** command will overwrite the existing Red Hat tftpd. This is unlikely to be a problem.

Once you have installed the tftpd, certain options need to be configured in `/etc/inetd.conf`:

| | |
|------------------------|--|
| /tftpboot | Although the documentation indicates atftpd will default to using <code>/tftpboot</code> , this does not seem to work, and must be specified manually. |
| --no-blksize | Try adding this to older PXE clients if you experience tftp failures on boot. |
| --tftpd_timeout | By default, atftpd exits after 300 seconds, logging a lot of information to syslog as it dies. If this setting is unsatisfactory, you can set this parameter to 0. |

Accordingly, modify the tftpd line in `/etc/inetd.conf` to something similar to the following:

```
tftp dgram udp wait nobody /usr/sbin/in.tftpd in.tftpd --no-blksize
--tftpd_timeout=0 /tftpboot
```

Note that this line will previously have been commented out; you will need to uncomment it. If you want to enable tftp right away, you can reload inetd with the command:

```
# /etc/rc.d/init.d/inet reload
```

6.4.5 Equinox setup

The ELS-16 from Equinox is a powerful device that supports virtually any combination of serial access you can think of. The manual is large and comprehensive, but there is no need to be put off by the size of the manual; configuring, for our purposes (known as reverse telnet), is simple.

The configuration procedure comes in two stages: the first over a serial cable, and the second over the ethernet. If your cluster is over 16 nodes, you will have multiple ELSs, so you will need to perform this procedure multiple times.

Assigning an IP

Before you configure the ELS, it is a good idea to reset it to the factory defaults. Ensure the ELS is powered up, then locate the reset hole next the ethernet connector. Use a paperclip or similar item to hold the switch down until the LEDs on the ELS start to flash. When the LEDs stop flashing, the ELS has been reset to the factory defaults. For further information, please refer to the Equinox manual.

In order to assign an IP to an ELS, you must connect a system (your management node, a laptop, or any other computer) to one of the Equinox serial ports. If your ELS is fully populated with cables, you will need to temporarily unplug one from an x330 and plug it into the computer you are using for the configuration. Using the usual cable and adapter, connect the computer you will be using for the configuration to one of the serial ports on the ELS.

If you have a laptop handy, you might prefer to access the serial port using HyperTerminal for Windows. Alternatively, you can connect to the management node and use the cu program.

Start your communication software and press Enter to get a login prompt from the ELS. If you do not get a prompt, check the cable and make sure your software is configured for 9600 baud, 8 data bits, no parity and one stop bit (8-N-1).

Tip: If you are using a laptop, start HyperTerminal for Window and do the following:

- ▶ Fill out the connection description (for example, Equinox).
- ▶ Configure the Connect using parameter (for example, Direct COM1).
- ▶ Set the communication parameters properly:
 - 8-N-1
 - Flow control: Hardware
- ▶ Check for the connection.
- ▶ If the connection is up, press Enter to initiate the Equinox login.

When you get a login prompt, configure the IP address following the procedure shown in Example 6-10:

Example 6-10 Equinox startup

```
[root@master /root]# cu -l /dev/ttyS0 -s 9600
Connected

LAT_00807D22E94D::PORT_16
Equinox Systems Inc., ELS-16
Welcome to the Equinox Server Network

Enter username> root
Local> set priv
Password> system
Local>> change server ip 192.168.2.253
Local>> change server subnet mask 255.255.255.0
Local>> logout
Local -020- Logged out port 16
~.
Disconnected
[root@master /root]#
```

Once you have disconnected, verify the IP has been assigned correctly by pinging the ELS.

In order to proceed with the next step of the configuration, you should disconnect the serial cable from the configuration system. If you borrowed the cable from one of your nodes, do not forget to put it back again.

Configuring the ports

Once the ELS has its IP assigned, you can complete the setup (configure the ports) by running the command shown in Example 6-11.

Example 6-11 Output of setupels command

```
root@master /root]# /usr/local/xcat/sbin/setupels mgels1

Trying 192.168.2.253...
Connected to mgels1.cluster.austin.ibm.com (192.168.2.253).
Escape character is '^]'.

#>

LAT_00807D22414E::Remote Console
Equinix Systems Inc., ELS-16
Welcome to the Equinix Server Network

Local> set priv
Password>
Local>> define port 1-16 access remote
Local>> define port 1-16 flow control enabled
Local>> define port 1-16 speed 9600
Local>> change port 1-16 que disable
Local>> lo port 1-16
Local>> [root@master /root]#
```

Verify that you can telnet to port 3001 of the ELS; you will have to use the IP of the ELS unless you already added its name to /etc/hosts. You should get a connection, but since there is nothing active on the port you will see no activity. Close the telnet session using '^]', as shown in Example 6-12.

Example 6-12 Equinix verification test

```
[root@master /root]# telnet mgels1 3001
Trying 192.168.2.253...
Connected to localhost.
Escape character is '^]'.
^]
(to exit the telnet command press: control and ] )
telnet> close
Connection closed.
[root@master /root]#
```

Your ELS is now set up. Go back and repeat the procedure for any additional ELSs you may have.

6.4.6 Conserver

Conserver is a program that manages and logs accesses to serial ports. It can be downloaded from:

<http://www.conserver.com>

You should also download the INSTALL.txt file which gives detailed installation instructions.

In order for conserver to work correctly with xCAT, it should be configured using the following command:

```
# cd /tmp
# tar zxvf conserver-7.0.0.tar.gz
# cd conserver-7.0.0
# ./configure --prefix=/usr/local/xcat/ --with-maxgrp=256
# make
# make install
```

Complete the conserver installation according to the instructions.

Once you have installed conserver, edit the `conserver.cf` in `/usr/local/xcat/etc`. **setupe1s** configures an ELS such that connects to TCP ports 3001-3016 and to serial ports 1-16, respectively. You need to add a configuration stanza for each of your nodes; in a large cluster, it is important to refer to your documentation for which node is connected to which port on which ELS. Example 6-13 shows the configuration for our four node cluster.

Example 6-13 Example conserver.cf

```
#
# The character '&' in logfile names are substituted with the console
# name. Any logfile name that does not begin with a '/' has LOGDIR
# prepended to it. So, most consoles will just have a '&' as the logfile
# name which causes /var/consoles/<consolenam> to be used.
#
LOGDIR=/var/log/consoles
#
# list of consoles we serve
# name : tty[@host] : baud[parity] : logfile : mark-interval[m|h|d]
# name : !host : port : logfile : mark-interval[m|h|d]
# name : |command : : logfile : mark-interval[m|h|d]
#
node1:!mgels1:3001:&:
node2:!mgels1:3002:&:
node3:!mgels1:3003:&:
node4:!mgels1:3004:&:
%%
#
```

```
# list of clients we allow
# {trusted|allowed|rejected} : machines
#
trusted: 127.0.0.1
```

Note the LOGDIR specification; this directory may not exist by default and may need to be created. Use the command:

```
# mkdir /var/log/console
```

It is OK if this directory already exists.

If you try and start conserver at this point, it may complain about a missing service. If you have not already done so, you must manually add a line to /etc/services similar to:

```
console 782/tcp conserver # console server
```

You should now be able to start conserver using the command:

```
# /etc/rc.d/init.d/conserver start
```

If it starts, configure it to automatically start at system restart with the command:

```
# /sbin/chkconfig conserver on
```

Check that you can connect to conserver with the command:

```
# /usr/local/bin/xcat/console -Mlocalhost node1
```

Note: Since there are no compute nodes running at this moment, the console command tests the console service only. To exit the command, type the following:

```
^EC.
(Control-E, C, period)
```

6.4.7 fping

fping is a small utility for pinging a large number of hosts in parallel. xCAT uses it (through **pping**) to provide a quick and easy way to check to see if all the nodes in your cluster are up. Download fping from:

<http://www.fping.com>

We had problems building 2.4b1, so you might try another version instead. There is a small patch for the fping output in /usr/local/xcat/build/fping, which should be applied before you build fping. Use the command:

```
# tar zxvf fping-2.3b1.patch.tar.gz
# cd fping-2.3b1
# patch < /usr/local/xcat/build/fping/fping-2.3b1.patch
```

Configure and build fping according to this process:

```
# ./configure --prefix=/usr/local/xcat
# make
# make install
```

You may notice that fping is installed as setuid root. This is because it must access raw sockets. This is perfectly normal.

6.4.8 OpenSSH

Many of the tools used on the cluster depend on OpenSSH, which can be downloaded from:

<http://www.openssh.com/>

By far the easiest way to install OpenSSH is via RPMs. Unfortunately, the packages available on the OpenSSH site require a newer version of the rpm code than what is available on the Red Hat 6.2 CD. More information and updates can be downloaded from <http://www.redhat.com/support/errata/> (see enhancement RHEA-2000-051). If you prefer, the latest version is always available from <http://www.rpm.org/>, but only use a *version 3* release.

Download the OpenSSH and RPM packages and copy them to /install/post/rpms where they will be automatically installed on the nodes. From there, install them on the master node.

Tip: In our lab, we used the following commands:

```
# cd /install/post/rpms
# rpm -Uv rpm*
# rpm -Uv openssh*
# rpm -Uv popt*
```

We downloaded the following files to /install/post/rpms

```
rpm-3.0.5-9.6x.i386.rpm
rpm-build-3.0.5-9.6x.i386.rpm
rpm-devel-3.0.5-9.6x.i386.rpm
rpm-python-3.0.5-9.6x.i386.rpm
openssh-2.5.2p2-1.i386.rpm
openssh-askpass-2.5.2p2-1.i386.rpm
openssh-askpass-gnome-2.5.2p2-1.i386.rpm
openssh-clients-2.5.2p2-1.i386.rpm
openssh-server-2.5.2p2-1.i386.rpm
popt-1.5-9.6x.i386.rpm
```

If any of the commands fail for any reason, you must identify and fix the problem, or you may be unable to access the nodes once they have been installed!

An ssh keypair must be generated to allow access to the nodes. For the xCAT commands that rely on ssh (particularly psh), it is important that ssh not prompt for a password. If you know how, you may choose to use ssh-agent for this purpose. However, you can simply create the keypair without a password. To do this, use the command:

```
# /usr/bin/ssh-keygen -N ''-f ~/.ssh/identity
```

The characters after -N are space, single quote, single quote, space...

To enable the other nodes to connect back to the management node, authorize the key you just created for login to root with the command:

```
cat ~/.ssh/identity.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

The **chmod** is important, as this file contains sensitive information and ssh will not honor it unless its permissions are `-rw-----` (you can check this with `ls -l`).

Your keypair creation is now complete. Copy these files into `/install/post` where they will be copied to the remaining nodes at install time with the command:

```
# cd ~/.ssh
# cp -pf identity identity.pub authorized_keys /install/post/.ssh/
```

Finally, there are a few ssh options you need for xCAT. The simplest way of configuring these is to copy the config file from `/install` with the command:

```
# cp /install/post/.ssh/config ~/.ssh/
```

Now ssh has been installed, you need to configure it to automatically start and start it right now with the command:

```
# /etc/rc.d/init.d/sshd start
# /sbin/chkconfig sshd on
```

6.5 Reboot and make sure it all works

You have now completed the installation of the master node. It might be a good idea to reboot and make sure all the services start OK. Use the command:

```
# shutdown -r now
```



Compute node installation

This chapter describes the preparation and installation process of the first compute node and the installation of the remaining compute nodes.

We talk about:

- ▶ xCAT tables
- ▶ Cluster Services
 - xntpd
 - DNS
 - DHCP
- ▶ MAC Addresses collection
- ▶ ASMA Setup
- ▶ Installation of the first compute node of the cluster
- ▶ Installation of the remaining nodes of the cluster

7.1 Populate tables

xCAT is almost entirely configured through a number of tables located in `/usr/local/xcat/etc`. Each of these is a plain text file that may be edited with your favorite text editor. Here we only discuss the most common use of these tables. A full reference may be found in Appendix B, “xCAT configuration files” on page 167.

Before configuring the xCAT tables, make sure your `/etc/hosts` is up-to-date. This is used by a number of the xCAT commands and must contain the complete listing of all the hosts in your network.

7.1.1 Site table

The first xCAT table is called `site.tab`. This contains information about the environment that the cluster runs in. Example 7-1 shows the site file used in our lab.

Example 7-1 `site.tab`

| | |
|---------------------------|--------------------------------------|
| <code>rsh</code> | <code>/usr/bin/ssh</code> |
| <code>rscp</code> | <code>/usr/bin/scp</code> |
| <code>tftpd</code> | <code>/tftpboot</code> |
| <code>tftpxcatroot</code> | <code>xcat</code> |
| <code>domain</code> | <code>cluster.austin.ibm.com</code> |
| <code>nameservers</code> | <code>192.168.0.254</code> |
| <code>net</code> | <code>192.168.0:255.255.255.0</code> |
| <code>gkhfile</code> | <code>/usr/local/xcat/etc/gkh</code> |
| <code>master</code> | <code>master</code> |
| <code>pbshome</code> | <code>/var/spool/pbs</code> |
| <code>pbsprefix</code> | <code>/usr/local/pbs</code> |
| <code>pbsserver</code> | <code>mgt1</code> |
| <code>scheduler</code> | <code>maui</code> |
| <code>nisdomain</code> | <code>NA</code> |
| <code>nismaster</code> | <code>NA</code> |
| <code>xcatprefix</code> | <code>/usr/local/xcat</code> |
| <code>timezone</code> | <code>US/Central</code> |
| <code>offutc</code> | <code>-6</code> |
| <code>mapperhost</code> | <code>node1</code> |
| <code>maxtrys</code> | <code>10</code> |
| <code>clustervlan</code> | <code>eth1</code> |
| <code>serialmac</code> | <code>1</code> |
| <code>dynamicb</code> | <code>172.16</code> |

Most of the fields are self explanatory and many may be left at their default settings. Be sure and check any that you do not understand against Appendix B, “xCAT configuration files” on page 167, because it can be time-consuming to rectify mistakes made in this file.

Attention:

- ▶ net stanza must be 192.168.0:255... NOT 192.168.0.0:255...'
- ▶ dynamicb stanza means class B network

The domain stanza must match the one used in /etc/hosts. If you only want to use node shortnames, you must still fill in this field; cluster would be one option.

7.1.2 Node list table

The second xCAT table to edit is the node list. It contains a list of all the nodes in the cluster and the groups each node is a member of. As with many of the xCAT tables, the first field must be the short host name of the node. The second field is simply a separated list of groups that the node is a member of. Example 7-2 shows the nodelist table used in our lab.

Example 7-2 nodelist.tab

```
node1 rack1,r1n1,mgasma1,all
node2 rack1,r1n2,mgasma1,all
node3 rack1,r1n3,mgasma1,all
node4 rack1,r1n4,mgasma1,all
```

The groups are simply for management convenience and may be specified in any way which you feel will be useful. Here we have chosen to group the nodes on the basis of the rack they are in (useful for multiple rack systems), their physical rack location, and the advanced systems management adapter (ASMA) they are connected to. You may recall that due to the way xCAT works, it is not possible to assign host names to the nodes based on their physical location. This is a way of partially working around that problem.

Notice that all the nodes are a member of the all group. This group is configured in the same way as all the other groups were; it would be possible, yet confusing, to define the all group as a subset of nodes. The all group is convenient for management purposes and it is strongly recommended that it include the entire collection of nodes. A number of commands, defined later in this chapter, are specified based on this assumption.

7.1.3 Node resources table

The next step is to allocate boot and install resources to the nodes. This is done with the node resources table. Example 7-3 show the noderes table used in our lab.

Example 7-3 noderes.tab

```
#TFTP          = Where is my TFTP server?
#              Used by makedhcp to setup /etc/dhcpd.conf
#              Used by mkks to setup update flag location
#NFS_INSTALL   = Where do I get my files?
#NFS_DIR       = From what directory?
#POST_DIR      = From what directory?
#NIS_SERVER    = Where do I authencate?
#              NA=No NIS
#TIME_SERVER   = Where do I sync my clock?
#SERIAL        = Serial console port (0, 1, or NA).
#INSTALL_ROLL  = Am I also an installation server? (Y or N).
#
#node/group
TFTP,NFS_INSTALL,NFS_INSTALL_DIR,POST_DIR,NIS_SERVER,TIME_SERVER,SERIAL,INSTALL
_ROLL
#
all    master,master,/install/rh62,/install/post,NA,master,1,N
```

Here we have defined one set of resources for the entire cluster, although this may not be desirable for larger clusters or those with different node types. If your cluster is to have multiple resource allocations, it is usually simplest to do this on the basis of node groups. You should have decided on the answers to all these questions up-front and may well find that this file will only need very minimal changes.

7.1.4 Node type table

You must now specify the types of node you wish to install in the node type table. Example 7-4 show the nodetype table used in our lab.

Example 7-4 nodetype.tab

```
node1 compute62
node2 compute62
node3 compute62
node4 compute62
```

This file specifies the Kickstart file that will be used to install the node. Since the kickstart file contains the post-installation script, this will affect the nodes configuration. The kickstart file is generated from the template file `/usr/local/xcat/ks62/nodetype.kstmp` with a number of values being automatically generated from the xCAT tables. `kstmp` files for `compute62` and `user62` are supplied. If you have any special requirements you may wish to modify these or create your own.

7.1.5 Node hardware management table

One of the key features of xCAT is its hardware management support. The node hardware management table details which methods of hardware management are available to xCAT. Example 7-5 show the `nodehm` table used in our lab.

Example 7-5 `nodehm.tab`

```
#node power,reset,cad,vitals,inv,cons,bioscons,eventlogs,getmacs
#
node1 asma,asma,asma,asma,asma,conserver,asma,asma,rcons
node2 asma,asma,asma,asma,asma,conserver,asma,asma,rcons
node3 asma,asma,asma,asma,asma,conserver,asma,asma,rcons
node4 asma,asma,asma,asma,asma,conserver,asma,asma,rcons
```

The example shows a typical example for our cluster of four x330s and will likely not need to be changed in that environment. Please see Appendix B, “xCAT configuration files” on page 167 for information on all the possible values.

7.1.6 ASMA table

In order to utilize the advanced system management (ASM) network, xCAT needs to know its topology. This is configured in the ASM table. Example 7-6 shows the ASMA table used in our lab.

Example 7-6 `asma.tab`

```
node1 mgasma1,node1
node2 mgasma1,node2
node3 mgasma1,node3
node4 mgasma1,node4
```

This table links the node host name to its corresponding ASMA card and ASM name. The first field is the IP host name of the ASMA card and the second is the ASM name of the node, which will always match its host name.

7.2 Configure cluster services

Once you have configured xCAT, you must configure the services it relies upon to perform the installation. Most of these can now be automatically set up based on the information you just provided.

7.2.1 xntpd

In order to keep the clocks in sync on the cluster, we install a time server (xntpd) on the management node. If you have another time source, you might want to configure xntpd on the management node to sync to this. It is only important that the clocks are the same on all the nodes, not that they tell the correct time. If you have no external time source, the default Red Hat configuration should work fine. To start the service, use the command:

```
# /etc/rc.d/init.d/xntpd start
# /sbin/chkconfig xntpd on
```

7.2.2 Domain name system (DNS)

Name server setup is likely to be very site-specific. If you plan to use real names for all the nodes in the cluster, it is important that you discuss with your network administrator how the nameservers will be configured and whether you should run the nameserver on the management node or not. Such details are beyond the scope of this redbook.

xCAT requires that the nameserver be able to resolve the addresses of the nodes in the cluster. If you know how, you may choose to configure the nameserver to resolve addresses both within and without the cluster. Again, that is beyond the scope of this redbook. If you decide to configure your nameserver for all addresses, note that **makedns** generates the nameserver configuration file, `named.conf`. It is recommended you run **makedns** first and only then perform local customization.

If you have correctly setup your `/etc/hosts` and `site.tab`, configuring the nameserver for the cluster addresses should simply be a case of running **makedns**:

```
# /usr/local/xcat/sbin/makedns
```

makedns will restart your nameserver as part of its run. Verify that it worked correctly by querying your local nameserver for addresses that are part of your cluster both by name and IP address. In Example 7-7, we use the **host** command to query the local nameserver (0 is more efficient than connecting to localhost).

Example 7-7 Host command to query the local nameserver

```
[root@master /root]# host node1.cluster.austin.ibm.com 0
Using domain server 0.0.0.0:
```



```
node1.cluster.austin.ibm.com has address 192.168.0.1
node1.cluster.austin.ibm.com mail is handled (pri=10) by
node1.cluster.austin.ibm.com
[root@master /root]# host 192.168.0.1 0
Using domain server 0.0.0.0:
1.0.168.192.IN-ADDR.ARPA domain name pointer node1.cluster.austin.ibm.com
```

If the above test failed, go back and check syslog (`/var/log/messages`) for any errors from the nameserver and verify the `db.*` files `makedns` created in `/etc`. You may have an error in either `site.tab` or `/etc/hosts`.

7.2.3 User ID management/Network Information System (NIS)

For job scheduling purposes, it is important that the user IDs from any user nodes be replicated across the cluster. xCAT includes support for NIS, although this support has been shown to suffer from scaling problems in larger clusters. The recommended way to distribute user IDs across the cluster is to copy `/etc/passwd` and `/etc/group` from the user node to all the compute nodes. Note that `/etc/shadow` should *not* be copied, since this would allow the users to login to the compute nodes; only the user IDs need be present to run batch jobs.

If you wish to use NIS you should set it up now according to your site environment. Otherwise, you should use `pcp` to copy the files out once you have set up the user node.

Network file system (NFS)

NFS is used in the cluster for the Kickstart install and to distribute home directories. To make these processes work, we must add certain directories to the export list contained in `/etc/exports`. The configuration from our cluster is shown in Example 7-8; note that there must be *no* spaces between the address and the options.

Example 7-8 /etc/exports

```
/install/rh62 192.168.0.0/255.255.255.0(ro,no_root_squash)
/install/post 192.168.0.0/255.255.255.0(ro,no_root_squash)
/usr/local 192.168.0.0/255.255.255.0(ro,no_root_squash)
/home 192.168.0.0/255.255.255.0(rw)
```

You can also specify, by host name, if your network is non-contiguous, although this has a small performance penalty. For example:

```
# /home node*(rw)
```

Once you have modified `/etc/exports`, start NFS with the command:

```
# /etc/rc.d/init.d/nfs start
```

Verify the file systems have been correctly exported by querying the NFS server, as shown in Example 7-9.

Example 7-9 Output of showmount -e

```
[root@master /root]# showmount -e
Export list for master:
/home      192.168.0.0/255.255.255.0
/usr/local 192.168.0.0/255.255.255.0
/install/post 192.168.0.0/255.255.255.0
/install/rh62 192.168.0.0/255.255.255.0
```

If the list does not match your `/etc/exports`, there is probably an error in the file. If the command fails completely, NFS is probably not working.

Once NFS works, configure it to start at boot-time with the command:

```
# /sbin/chkconfig nfs on
```

7.2.4 Dynamic Host Configuration Protocol (DHCP)

DHCP can be automatically configured using the `gendhcp` script supplied with xCAT. Note that this script sets up an alias on your cluster vlan adapter. If you specified `eth1` for `clustervlan` in `site.tab`, `gendhcp` will create the alias `eth1:0`. Do not use this alias yourself or it will be overwritten.

The `gendhcp` command creates a skeleton of the `dhcpd.conf` in `/tmp`:

```
# gendhcp
```

The output from the script will suggest you edit `dhcpd.conf` by adding IP addresses for other devices, such as your Myrinet switch.

Note: You need to remove unwanted network entries. In our lab we found some unwanted entries that were removed.

To add an entry for a static host, you need to know the MAC address of the device. Make sure the entry to the stanza matches your management network. An example of a static host is shown in Example 7-10.

Example 7-10 subnet stanza for `dhcpd.conf` file

```
## subnet declaration for management network
subnet 192.168.2.0 netmask 255.255.255.0 {
  ## Static host entry for the myrinet switch
  host mgmyril {
    hardware ethernet 00:60:dd:7f:94:b0;
    fixed-address mgmyril;
  }
}
```

```
}  
}
```

When you are done editing the `dhcpd.conf`, copy it to `/etc` and restart the DHCP server with the command:

```
# cp -f /tmp/dhcpd.conf /etc/  
# /etc/rc.d/init.d/dhcpd restart
```

If there is an error in `dhcpd.conf`, the server will fail to start and will produce an error message. Check the `dhcpd.conf` and your `site.tab` and re-run `gendhcp` if necessary. Also check errors at `/var/log/message`.

Tip: An alternative to avoid having DHCP to offer and assign IP addresses in all interfaces (`eth0`, `eth1`, ...) is to modify the `dchpd` script by adding the interface name in the `dhcpd` startup. The script is located at `/etc/rc.d/init.d/dhcpd`. For example:

```
...  
# Start daemons.  
echo -n "Starting dhcpd: "  
daemon /usr/sbin/dhcpd eth1  
...
```

7.3 Collect MAC addresses

Once the DHCP server has been set up, we are almost ready to collect the MAC addresses of the nodes. Two scripts must be run before we do the collection:

```
# /usr/local/xcat/stage/mkstage  
# /usr/local/xcat/ks62/mkks
```

Part of the work these scripts do is validation. If they report missing files or similar errors, it probably means you have missed a step of the installation process. Their main task is to create various files in `/ftptboot` for the remote boot process.

If `mkstage` and `mkks` ran OK, it is time to collect the MAC addresses. This is done using the `getmacs` script, as shown in Example 7-11.

Example 7-11 Running `getmacs`

```
[root@master /root]# cd /tmp  
[root@master /tmp]# /usr/local/xcat/sbin/getmacs all
```

Please reset nodes: node1 node2 node3 node4

Press [Enter] when ready...

Saving output to mac.lst in current directory /tmp.

```
node4 00:06:29:1F:2B:CD
node3 00:06:29:1F:6C:87
node2 00:02:55:54:2B:50
node1 00:06:29:1F:29:69
```

Edit mac.lst and merge with /usr/local/xcat/etc/mac.tab

```
[root@master /tmp]#
```

The getmacs script will prompt you to reset the nodes; right now, this has to be done on the front panel, because the service processor network (SPN) is not yet set up. Power on or press the reset button on all the nodes you are collecting MAC addresses for and then press Enter.

As it runs, getmacs outputs to the panel and to the mac.lst file. If no nodes report NO MAC, then you have successfully captured the addresses of all your nodes. Unfortunately, it is not uncommon for one or two nodes to fail. If this happens, you must run getmacs again for the nodes that failed. Be sure and save mac.lst somewhere before you run getmacs again, because it will overwrite your original file. Once you have the MAC addresses of all the nodes, merge them all into a single file and copy this to mac.tab. This can be done with a text editor or, if you named all your files with the mac.lst. prefix, you could use the command:

```
# grep -hv 'NO MAC' mac.lst* | sort > /usr/local/xcat/etc/mac.tab
```

If the MAC address capture fails on all your nodes or you have persistent errors with a particular node, there are a number of things to check:

- ▶ MAC address collection (depends on DHCP)
- ▶ TFTP and the remote console facility and rcons

The first place to look for clues is the console of the node; Pre-Execution Environment (PXE) produces very helpful error messages. If the node is actually booting Linux, you may need to use rcons to see all of the messages or look in /var/log/consolas. The other place to check (as for most things) is the syslog output in /var/log/messages.

Important: At this point, if you still have problems collecting MAC addresses, we recommend you go back to the Section 6.4, “Additional software installation” on page 93 for help.

7.4 Setup rangers (stage3)

Once you have the MAC addresses of the nodes, the next stage is to set up the ASM network. This uses a similar process to `getmacs`; if that worked without any problems, this should be fairly straightforward.

First, you must configure the DHCP server with the MAC addresses of the nodes, since the ASM setup sets the ASM name to the host name of the node's IP. Run the `makedhcp` command (that is different from the `gendhcp` command used earlier), then use the `nodeset` command to set the nodes to stage3.

```
# /usr/local/xcat/sbin/makedhcp
# /usr/local/xcat/bin/nodeset all stage3
```

The management node is now ready for ASM setup. Press the reset button on all the nodes and their rangers should be automatically set up. You can monitor the progress of the setup by running:

```
# /usr/bin/watch /usr/local/xcat/bin/nodeset all stat
```

The nodes will start out at stage3 and as they complete, setup will change over to boot. If any of the nodes fail to change over, they might simply need resetting again when the network is quieter; do not forget to check the nodes and syslog for errors. The most likely causes of persistent problems are corrupted `dhcp.conf` or a named error.

When all the nodes have reverted to boot, check to see if the SPN setup completed correctly by running:

```
# /usr/local/xcat/bin/rinv all serial
```

Ensure all the nodes show up and with a unique serial number.

7.5 Install first node

The nodes are now ready to be installed. It is a good idea to install a single node before installing the entire cluster so that any problems can be easily identified.

Nodes are installed using the `rinstall` command or `winstall` if you want to watch the install. For example:

```
# /usr/local/xcat/bin/winstall node1
```

Note that `winstall` uses `rcons`, so it may take a little while before any output appears in the panel.

If something fails, it is most likely NFS or a problem caused by an error in the xCAT tables, since DHCP and TFTP have been tested by now. Again, check the consoles and logs for more information, paying particular attention to the console log. If you ran **winstall**, there may be partially obscured messages on the panel; you can read these from the console log if you need to.

7.6 Install remaining nodes

Once you know the install process works for one node go ahead and install the remaining nodes. If you do not have very many nodes, it is best to use **winstall** for this process, since you can watch all the nodes installing at once. To open console panels and install all the nodes except node1, use the command:

```
# /usr/local/xcat/bin/winstall all,-node1
```

As with stage3, you may want to use watch to verify when all the nodes have been installed.

7.7 Post install

Once all the nodes are installed, the last thing you need to do is generate the ssh global known hosts file. First, verify that all the nodes are up and running with the command:

```
# /usr/local/xcat/bin/pping all
```

If any nodes come back with noping, they may not have finished rebooting after the install or something may have gone wrong at reboot. Have a look in that node's console log for errors.

Collect the host keys for all the nodes using **makesshgkh**:

```
# /usr/local/xcat/sbin/makesshgkh all
```

You should see a message from all the nodes, but it is sometimes difficult to see if it succeeded or not. Check the file `/usr/local/xcat/etc/gkh` and make sure there is an entry for all the nodes. If not, run **makesshgkh** again for the missing nodes. If it continues to fail, there may have been a problem with the nodes installation; check the console log for possible error messages.

You can verify that the keys are correct using the **psh** command, which relies on ssh. Distribute the **hostname** command to all the nodes and ensure they all respond without errors. See Example 7-12 on page 121 for details.

Example 7-12 Using psh to verify the operation of all the nodes

```
[root@master /root]# psh all
Executing on: node1 node2 node3 node4
> hostname
node1: node1
node2: node2
node3: node3
node4: node4
> ^D
[root@master /root]#
```

Congratulations! You have now completed your basic cluster installation.



Installation of additional components

In this chapter, the installation of some additional components are shown, such as compilers and libraries.

This is the list of the installation components presented in this chapter:

- ▶ GCC compiler
- ▶ PGI compiler
- ▶ MPICH, using TCP/IP
- ▶ MPICH, using Myrinet GM

8.1 Install GCC or PGI Compiler

In this section, we will discuss how to install the GNU C compiler (gcc) and PGI compilers.

8.1.1 GNU C compiler (gcc)

gcc is included in every common Linux distribution and is installed as a default. If for some reason you do not have it installed, you can download it from:

<http://www.gnu.org/software/gcc/gcc.html>

Follow the instructions on:

<http://www.gnu.org/software/gcc/install/index.html>

If you prefer RPM, try the biggest RPM repository at:

<http://rpmfind.net/linux/RPM/>

and install it by using the command:

```
rpm -Uvh gcc-x.xx..
```

To check which version of GCC you have, use the following command:

```
[root@master]# gcc -v
Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/specs
gcc version egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)
```

Note: Several bugs have been found and fixed in the GNU Compiler Collection snapshot that shipped with Red Hat Linux 7.0. For RH Security fixes, bug fixes, and package enhancements, go to:

<http://www.redhat.com/support/errata/>

After installing the gcc, you can proceed to the MPICH installation.

8.1.2 PGI Workstation code

The PGI Workstation is a full suite of parallel F77, F90, HPF, C and C++ compilers and development tools. The release we used was v.3.2; you can obtain it from:

<http://www.pgroup.com>

You must read the licensing details and register. The file you need is linux86-HPF-CC.tar.gz, and the size of the file is 16.9 MB.

Note: If you are using the trial version of the PGI Workstation, the compiler and all compiled executables will cease to function after 15 days. Each time you compile using the trial version, it will remind you of the time you have left. To overcome this situation, you need to obtain the permanent license keys from the Portland Group. Any executable, object files, or libraries created using the PGI compilers in trial mode must be recompiled with the permanent license keys in place.

Install PGI Workstation code

For full instructions, refer to the PGI Workstation online documentation.

Quick path

1. Create the following directories with the command:

```
[root@master local]# mkdir /usr/local/pgi
[root@master local]# mkdir /tmp/pgi
[root@master local]# cd /tmp/pgi
```

2. Set environment variables. If you prefer, you can include the exports in your login file at `$HOME/.bashrc` or create `/etc/profile.d/pgi.sh`. Use the commands:

```
[root@master pgi]# export PGI=/usr/local/pgi
[root@master pgi]# export PATH=$PGI/linux86/bin:$PATH
[root@master pgi]# export MANPATH=$MANPATH:$PGI/man
```

3. Unpack the downloaded archive with the commands:

```
[root@master pgi]# cp <PGI_DOWNLOAD_DIR>/linux86-HPF-CC.tar.gz /tmp/pgi
[root@master pgi]# tar -zxfv linux86-HPF-CC.tar.gz
```

4. Install the package (PGI Workstation or PGI Server) with the following script:

```
[root@master pgi]# ./install
This script installs PGI Workstation products for the
Linux and Solaris86 operating systems. Select the product
below that you purchased, and the directory where the software
is to be installed.
```

- 1 PGCC Workstation or PGCC Server
- 2 PGF77 Workstation or PGF77 Server
- 3 PGHPF Workstation or PGHPF Server
- 4 PGF77/PGCC Workstation or PGF77/PGCC Server
- 5 PGI Workstation or PGI Server

(PGI, PGHPF, PGCC, and PGF77 are all Reg. U.S. Patent and Trademark Office to The Portland Group, Inc.)

Enter a number between 1 and 5:

2 <--- in this demo we prefer option 2, in production use you should probably use option: 5

Please specify the directory path under which the software will be installed. The default directory is /usr/pgi, but you may install anywhere you wish, assuming you have permission to do so.

Installation directory [/usr/pgi]?
/usr/local/pgi

Installing software into /usr/local/pgi (this may take some time).

If you don't already have permanent keys for this product/release, a fifteen-day evaluation license can be created now.

Create an evaluation license? [y/n]

y

Do you accept these terms? [accept,decline]

accept

Creating temporary license.

Please enter your name: *Name*

Please enter your user name: *User Name*

Please enter your E-mail address: *eMail*

[...]

Do you wish to change anything? [yes/no]: no

[...]

The above information was saved to /usr/local/pgi/license.info.

Do you want the files in the install directory to be read-only? [y,n]

y

Once you are done. you can check the versions of the compilers with the appropriate command:

- ▶ For Fortran 77, use **pgf77 -V**
- ▶ For Fortran 90, use **pgf90 -V**
- ▶ For HPF, use **pghpf -V**
- ▶ For C++, use **pgCC -V**
- ▶ For ANSI C, use **pgcc -V**

8.2 Install MPICH to use either TCP/IP or Myrinet GM

In this section, we will discuss how to install MPICH so you can use it with TCP/IP or Myrinet GM.

8.2.1 Remove local area multicomputer/message passing interface

To avoid conflicts with MPICH, you will have to remove the local area multicomputer/messaging passing interface (LAM/MPI) components that are installed as the defaults. To remove the component, use the following command:

```
[root@master local]# rpm -e lam
```

8.2.2 Command mpimaker

xCAT includes a command called **mpimaker** that simplifies the installation and configuration of MPICH. You can configure it to utilize elements like TCP/IP or Myrinet, symmetric multiprocessing (SMP) or uni-processing (UP), gcc or PGI compilers, ssh (Preferred), or rsh.

The command **mpimaker** is located in `/usr/local/xcat/build/mpi/mpimaker`.

Attention:

- ▶ If you are using ethernet cards for IPC, use MPICH with TCP/IP.
- ▶ If you have Myrinet cards for IPC, install GM MPICH (you need to specify both the GM MPICH version and the GM version, as will be demonstrated).
- ▶ Use the SMP or UP switch according to your hardware setup.
- ▶ We prefer PGI over gcc (PGI is optimized).
- ▶ We prefer ssh over rsh (ssh scales better and is secure).

8.2.3 Configure MPICH to use TCP/IP

In this section, we discuss how to configure MPICH to be compatible with TCP/IP.

Download and Install MPICH code

The current release we used was version 1.2.1. It can be downloaded from:

<http://www-unix.mcs.anl.gov/mpi/mpich/>

The file you need is `mpich.tar.gz`, and the size of the file is about 9.5 MB.

Note: Write down the release version; you will need it later.

Copy the MPICH source and mpimaker script files to a temp work directory that you will create using the commands:

```
[root@master root]# mkdir /tmp/mpi
[root@master root]# cd /tmp/mpi
```

You need to rename mpich.tar.gz to mpich-VERSION.tar.gz (in our case, it is mpich-1.2.1.tar.gz) You can do this as you copy it to the temp work directory. Use the commands:

```
[root@master mpi]# cp <Path to MPICH Download Dir>/mpich.tar.gz
/tmp/mpi/mpich-1.2.1.tar.gz
[root@master mpi]# cp /usr/local/xcat/build/mpi/* .
```

You can launch the mpimaker script to display the options available with the command:

```
[root@master mpi]# ./mpimaker
usage: [mpich version|gm mpich version:gm version] [up|smp] [gnu|pgi] [rsh|ssh]
```

Compile with the options you choose using the command:

```
[root@master mpi]# ./mpimaker 1.2.1 smp gnu ssh
```

This example is for MPICH Version 1.2.1, multiprocessor, GNU compiler, and ssh. The files will be created in /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/ using the command:

```
[root@master mpi]# ./mpimaker 1.2.1 smp pgi ssh
```

For MPICH Version 1.2.1, multiprocessor, PGI compiler, and ssh, the files are created in /usr/local/mpich/1.2.1/ip/smp/pgi/ssh/.

Before you can run any program on the cluster, it is necessary to know how many nodes and processors you have and what are they called in order to launch jobs on them. You will need to edit a configuration file for this purpose. In this case, the configuration file (called machines.LINUX) is located in the directory /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/share/. The file format is one host name per line, with either hostname or hostname:n, where n is the number of processors in an SMP. The host name should be the same as the result from the command **hostname**.

Use your favorite text editor to edit the machines.LINUX file using the commands:

```
[root@master mpi]# cd /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/share/
```

```
[root@master share]# vi machines.LINUX
```

Examples of machines.LINUX file:

```
#4 Uni processor nodes
node1
node2
node3
node4
master
```

Or

```
#4 SMP nodes
node1:2
node2:2
node3:2
node4:2
master
```

In this case, we have four SMP nodes:

```
[root@master share]# pwd
/usr/local/mpich/1.2.1/ip/smp/gnu/ssh/share
[root@master share]# cat machines.LINUX
node1:2
node2:2
node3:2
node4:2
master
```

Note: You might need to add the path that goes to mpicc into the \$PATH variable (in our case, mpicc resides in /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/bin/), so the format in our example would be:

```
export PATH=/usr/local/mpich/1.2.1/ip/smp/gnu/ssh/bin/:$PATH
```

Small test - CPI

With MPICH, you will have many test applications to ensure everything has installed OK. We used a program called CPI to test basic connectivity. It computes the value of pi and compares it to the value 3.14159265358979.

The program CPI can be found in /usr/local/mpich/xxxx/examples (depending on your compile options); in our case, the directory is the following:

```
/usr/local/mpich/1.2.1/ip/smp/gnu/ssh/examples
[root@master local]# cd /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/examples
```

You can compile CPI with the command:

```
[root@master examples]# make cpi
```

In the examples directory, you will also see a linked mpirun, which you can use to launch the CPI.

Tip: Generally, you can use the command **which** to see where the **mpirun** command is executed from. For example:

```
[root@master examples]# which mpirun
/usr/local/mpich/1.2.1/ip/smp/gnu/ssh/bin/mpirun )
```

Launch the CPI test program using the command:

```
[root@master examples]# ./mpirun -np 8 cpi
Process 0 on master.cluster.austin.ibm.com
Process 5 on node3
Process 2 on node1
Process 3 on node2
pi is approximately 3.1416009869231249, Error is 0.0000083333333318
wall clock time = 0.003784
Process 4 on node2
Process 1 on node1
```

Change the number 8 according to the number of processes you want to launch. It is a good idea to have one process per CPU and no more processes than the total number of CPUs.

Now you are ready to test your cluster. In our lab, we used the program POVRay with TCP/IP to perform the test. For more information about POVRay, see Appendix C, “POVRay test” on page 185.

8.2.4 Configure MPICH to use Myrinet GM

In this section, we discuss how to download and install the Myrinet GM driver and then configure MPICH to be compatible with Myrinet GM.

Download and Install Myrinet GM driver

First, download the Myrinet GM source package from:

```
http://www.myri.com/
```

Then go to GM Software downloads. In our lab, we used the card Type PCI64B (LANai 9).

The name of the file you need is gm-1.4.tar.gz size, and the size of the file is about 6 MB. You must register in order to download the file.

Note: Write down the GM version; you will need it later. In our example, it is 1.4.

Copy the GM source and xCAT gmaker script files to /tmp directory using the commands:

```
[root@master /root]# mkdir /tmp/gm
[root@master /root]# cd /tmp/gm
[root@master gm]# cp <Path to GM Download Dir>/gm-1.4.tar.gz .
[root@master gm]# cp /usr/local/xcat/build/gm/* .
```

Build and compile the gm-x.x package for Myrinet, as shown in the following command:

```
[root@master gm]# ./gmmaker 1.4 smp
```

This command will build the driver for GM Version 1.4 and SMP machine. Currently, the gmaker script from xCAT v.1.1 will use the following compile options:

```
--enable-linux-modversions --enable-new-features --enable-directcopy and
--enable-linux-smp (if you specified SMP machine)
```

You can edit the gmmaker shell script and add new options as new versions of GM are released.

When the script is completed, you will find a new directory in /usr/local/ gm version - Kernel version, for example, in our lab it was /usr/local/gm-1.4-2.2.18smpx. Write down the newly created directory; you will need it later for the mpimaker script. The following example shows the parallel installation using the **psh** command:

```
[root@master gm]# psh all "export PATH=$PATH:/sbin; cd
/usr/local/gm-1.4-2.2.18smpx; ./scripts/gm_install"
```

Edit the /usr/local/xcat/build/gm/gm script line GM_VER=1.3.1 to reflect your GM version; in this case, it should read GM_VER=1.4. Use your favorite text editor. For example:

```
[root@master gm]# vi /usr/local/xcat/sys/gm
#!/bin/sh
# gm          This shell script takes care of starting and stopping
#            gm (myrinet).
# chkconfig: 2345 60 60
# description: gm
# processname: gm module
```

```
# config: /usr/local/gm*
GM_VER=1.3.1 <----- edit this line to reflect your gm version (GM_VER=1.4)
```

Copy the GM script and set the script to load the driver each time the node boots using the commands:

```
[root@master gm]# psh all cp /usr/local/xcat/sys/gm /etc/rc.d/init.d/gm
[root@master gm]# psh all /sbin/chkconfig --level 345 gm on
```

Next, edit the file `/usr/local/xcat/build/gm/mapper` with your favorite text editor. Edit the `/usr/local/xcat/build/gm/mapper` script line `GM_VER=1.3.1` to reflect your GM version; in this case, it should read `GM_VER=1.4`. For example:

```
[root@master gm]# vi /usr/local/xcat/sys/mapper
#!/bin/sh
# mapper          This shell script takes care of starting and stopping
# mapper (myrinet).
# chkconfig: 2345 60 60
# description: mapper
# processname: mapper
# config: /usr/local/gm*
GM_VER=1.3.1 <----- edit this line to reflect your gm version, GM_VER=1.4
```

Copy the edited mapper file to node1 `/etc/rc.d/init.d/` directory using the command:

```
[root@master gm]# psh node1 cp /usr/local/xcat/sys/mapper
/etc/rc.d/init.d/mapper
```

Make it run each time node1 is booted by using the command:

```
[root@master gm]# psh node1 /sbin/chkconfig --level 345 mapper on
```

To check if it was added correctly, use the command:

```
[root@master gm]# psh node1 /sbin/chkconfig --list mapper
node1: mapper          0:off  1:off  2:off  3:on   4:on   5:on   6:off
```

Important: Run mapper once for every time a node is added/deleted or any other changes have been made in the Myrinet network, because you will always need a current map of the Myrinet hosts. In this example, we chose Node1 to be the mapper; this function can be performed by any of the Myrinet hosts.

Start the mapping process by issuing the following commands:

```
[root@master gm]# psh node1 /etc/rc.d/init.d/mapper start
```

After mapping, begin the GM driver installation with the command:

```
[root@master gm]# psh all /usr/local/gm-1.4-2.2.18smpx/bin/gm_board_info
```

You will see information about the Myrinet cards in each node and the IDs that are mapped.

Configure MPICH to use Myrinet GM

Once you have the Myrinet hardware set up, you will need to configure MPICH. In order to benefit from the Myrinet hardware in a cluster, you will use a version of MPICH message passing specially modified and optimized to utilize Myrinet.

You can download the source from:

<http://www.myri.com/>

We used GM MPICH Version 1.2..5.

The name of the file is mpich-1.2..5.tar.gz, and the size of the file is about 8 MB. You must register to download the file.

Do not forget to write down the exact version of the package; in this case, it is 1.2..5.

Copy the source and xCAT mpimaker script files to /tmp with the commands:

```
[root@master /root]# mkdir /tmp/gmpi
[root@master /root]# cd /tmp/gmpi
[root@master gmpi]# cp <Path to Myri mpich Download Dir>/mpich-1.2..5.tar.gz .
[root@master gmpi]# cp /usr/local/xcat/build/mpi/* .
```

You can launch the mpimaker script to display the options available using the command:

```
[root@master gmpi]# ./mpimaker
usage: [mpich version|gm mpich version:gm version] [up|smp] [gnu|pgi] [rsh|ssh]
```

Now you can compile. To do this, you have to input the exact GM MPICH version name and GM install directory (mentioned earlier), separated by a semicolon.

Use the command:

```
[root@master gmpi]# ./mpimaker 1.2..5:1.4-2.2.18smpx smp pgi ssh
```

This will build MPICH Version 1.2..5 and GM Version 1.4 and use SMP machine with the PGI compiler and ssh.

You will see the following output:

```
./mpimaker: 1.2..5:1.4-2.2.18smpx smp pgi ssh build start
./mpimaker: 1.2..5:1.4-2.2.18smpx smp pgi ssh make
./mpimaker: 1.2..5:1.4-2.2.18smpx smp pgi ssh build successful
```

The process will take some time to complete. If you wish, you can easily monitor the status by opening another shell panel. In our case, we use the following command:

```
[root@master /root]# tail -f /tmp/gmpi/mpich-1.2..5/make.log
```

Before you can run any programs on the cluster, you need to know how many nodes and processors you have and what are they called in order to launch jobs on them.

The next step is to create and populate a configuration file in order to use MPICH with GM of Myrinet card. In this case, the configuration file must reside in `$HOME/.gmpi/conf`. The file must include all the nodes that you are planning to run jobs on. The file has a special format, where the first line must contain the total number of nodes followed by lines describing the node name and the GM port to use. If you have an SMP machine, you need to add a node description twice, and use different GM port numbers for each entry. For example:

```
[root@master gmpi]# cd
[root@master root]# mkdir .gmpi
[root@master .gmpi]# cd .gmpi
```

Use your favorite text editor here to create the conf file. Use the command:

```
[root@master .gmpi]# vi conf
```

Here are some examples of the `.gmpi/conf` file:

```
#4 Uni processor nodes $HOME/.gmpi/conf file :
4 <----- total number of cpus
node1 2
node2 2
node3 2
node4 2
```

or

```
#4 SMP nodes $HOME/.gmpi/conf :
8 <----- total number of cpus
node1 2
node1 4
node2 2
node2 4
node3 2
node3 4
node4 2
node4 4
```

In our lab, we had four SMP nodes:

```
[root@master .gmpi]# cat conf
```

```
# .gmpi/conf file begin
# first the number of nodes in the file
8
# the list of (node, port, board) that make the MPI World
node1 2
node1 4
node2 2
node2 4
node3 2
node3 4
node4 2
node4 4
```

Small test - CPI

With MPICH, you will also have many test applications to ensure everything has installed OK. We used a program called CPI to test basic connectivity. It computes the value of pi and compares it to the value 3.14159265358979.

The program CPI can be found in `/usr/local/mpich/xxxx/examples` (depending on your compile options); in our case, the directory is the following:

```
[root@master]# cd
/usr/local/mpich/1.2..5/gm-1.4-2.2.18smpx/smp/pgi/ssh/examples
```

To compile the CPI program, use the **make** command:

```
[root@master examples]# make cpi
```

Tip: Now would be a good time to check the link between `mpirun` and `mpirun.ch_gm`. If you prefer, you can change the path to point to the directory that `mpirun` resides in. Use the command:

```
[root@master examples]# ln -sf ../bin/mpirun.ch_gm mpirun
```

In the examples directory, you will also see a linked **mpirun**, which you can use to launch the CPI. Generally, you can use the command **which** to see where the command is executed from. This command is useful when you have a long `$PATH` environment. The command is as follows:

```
[root@master examples]# which mpirun
```

Launch the CPI test program with the command:

```
[root@master examples]# ./mpirun -np 8 cpi
Process 1 on node1
Process 2 on node2
Process 3 on node2
Process 4 on node3
Process 6 on node4
```

```
Process 7 on node4
Process 0 on node1
Process 5 on node3
pi is approximately 3.1416009869231245, Error is 0.0000083333333314
wall clock time = 0.000733
[root@master examples]#
```

Change the number 8 according to the number of processes you want to launch. It is a good idea to have one process per CPU and no more processes than the total number of CPUs.

8.3 Installing libraries and debugger

Installing libraries and debugger depends on which applications you are planning to run and if you really need to debug or troubleshoot them. Also, you need to find out if the applications you are planning to run will benefit from these libraries; you might need libraries that we have not mentioned for special scientific applications. Each of the libraries and the debugger have install info included in the package or in their home page.



A

xCAT help

For an easier understanding of the way xCAT should be used, we provide a short introduction to xCAT, explaining what xCAT is and what it is not. Furthermore, we will cover its structure and basic contents.

What xCAT is

xCAT is a collection of shell scripts written by Egan Ford to, more or less, automate the installation and maintenance of xSeries Linux clusters. Additionally, it is packed into a distribution providing you with a consistent directory structure and a few small patches for third party tools.

What xCAT is not

xCAT is by no means complete commercial all-in-one xSeries installation and management tool. It is using the UNIX approach by combining several different GNU/Linux tools and third party software with a more powerful package of scripts that are used to install and manage a Linux xSeries cluster.

Directory structure and contents

xCAT should be untarred into `/usr/local/`; the following subdirectories can be found below `/usr/local/xcat/`:

| | |
|---------------|---|
| bin/ | Commands (shell scripts) for cluster management. |
| build/ | Directories for building third party software, such as PBS, Maui, and so on; some directories contain patches to be |

applied before building the software to fix incompatibilities or small bugs.

| | |
|-----------------|--|
| etc/ | Configuration files and tables used by xCAT commands. |
| html/ | Uncompleted documentation in HTML. |
| ks62/ | Default Red Hat Kickstart and PXELinux configuration files for compute nodes installation. |
| lbin/ | Additional, mostly binary, executables used by some commands. |
| lib/ | Scripts used by commands in bin/. |
| post/ | Files for post installation modifications of the installed nodes. |
| samples/ | Sample files for different third party applications. |
| sbin/ | Commands for setting up the cluster (node) installation. |
| src/ | Sources for some supplied tools. |
| stage/ | Staging image for node installation and scripts to generate the complete staging images. |
| sys/ | init process scripts for several applications and services. |
| tftp/ | PXELinux default configuration file and link to tftpd in post/. |

Please pay attention to the INSTALL file and follow its instructions for populating the xCAT directory tree with the missing third party software before actually using the xCAT commands.

Table A-1 provides a quick reference of the xCAT commands. In this same appendix, we provide a full description of each xCAT command, including the syntax, and examples on how to use them.

Table A-1 xCAT commands

| Command | Description |
|---------------|---|
| rpower | rpower can control the power (on/off/stat) for a range of nodes. |
| rreset | Sends a hard reset (like reset button) to a range of nodes. |
| rcad | Remote soft reset (Ctrl+Alt+Del). |
| rcons | Remote (text) console. |
| wcons | Remote (video) console. |

| Command | Description |
|-----------------|---|
| rvid | Redirects console text video. Useful for debugging boot issues. |
| wvid | Redirects console text video. It is functionally equivalent to rvid , except that wvid will launch an xterm |
| rvitals | Retrieves vital hardware information from the on-board Service Processor for a range of nodes. |
| revent | Displays any number of remote hardware event log entries or clears them for a range of nodes. |
| rinv | Retrieves hardware configuration information from the on-board Service Processor for a range of nodes. |
| psh | Runs a command across a list of nodes in parallel. |
| pping | Pings a list of nodes in parallel. |
| rinstall | Forces an unattended network install for a range of nodes. |
| winstall | Forces an unattended network install for a range of nodes. It is functionally equivalent to rinstall , except that will launch an xterm panel for the remote console display |

The remainder of this appendix is devoted to describing the xCAT commands.

rpower - Remote power control

Description

rpower can control the power (on/off/stat) for a range of nodes.

Synopsis

```
rpower [noderange] [on|off|stat]
For example:
rpower node1,node2,node8,node20 stat
rpower node14-node56,node70-node203 stat
rpower node1,node2,node8,node20,node14-node56,node70-node203 stat
rpower all -node129-node256 off
```

Example

```
# rpower node4,node5 stat
node4: on
node5: off
# rpower node5 on
node5: on
```

rreset - Remote hardware reset

Description

`rreset` sends a hard reset to a range of nodes.

Synopsis

```
rreset [noderange]
For example:
rreset node1,node2,node8,node20
rreset node14-node56,node70-node203
rreset node1,node2,node8,node20,node14-node56,node70-node203
rreset all -node129-node256
```

Example

```
# rreset node3-node5
node3 reset
node4 reset
node5 reset
```

rcad - Remote software reset

Description

rcad sends a Ctrl+Alt+Del (soft reset) to a range of nodes. This command is currently only supported by x330 nodes. Support for other node models is forthcoming.

Synopsis

```
rcad [noderange]
For example:
rcad node1,node2,node8,node20
rcad node14-node56,node70-node203
rcad node1,node2,node8,node20,node14-node56,node70-node203
rcad all -node129-node256
```

Example

```
# rcad node3-node5
node3 Ctrl+Alt+Del function not supported
node4 Ctrl+Alt+Del function not supported
node5 Ctrl+Alt+Del
Nodes 3 and 4 are 4500Rs, node5 is an x330.
```

Files

| | |
|----------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tabService | Processor Network definition table. |

Diagnostics

| | |
|--|--|
| Ctrl+Alt+Del function not supported | Returned if node model does not support Ctrl+Alt+Del. |
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | Telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |

node xxx not on spn yyy

xxx is not listed in SPN yyy. Use **spncheck yyy** to verify. Check cabling and try **spnreset**.

Remote session timeout

Telnet was successful but communication with node failed. Wait one minute and try again.

See also

noderange

rreset

rcons - Remote console

Description

`rcons` provides access to the remote serial console.

Synopsis

```
rcons [single node]
For example:
rcons node23
```

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| ts.tab | Terminal Server definition table. |

Diagnostics

| | |
|-------------------------------|--|
| xyz is not a node | Returned if node name not in <code>nodelist.tab</code> . |
| ts:port is unavailable | Returned if port is in use by another user. <code>ts</code> = Terminal Server host name, <code>port</code> = 1-16. |
| telnet to xyz failed | telnet command timed out. Check Ethernet cable and IP. Check DNS and <code>ts.tab</code> . |

See also

`wcons`

wcons - Remote console

Description

wcons provides access to the remote serial console. **wcons** is functionally equivalent to **rcons**, except that **wcons** will launch an xterm panel for the remote console display, as shown in Figure A-1.

Synopsis

```
wcons [noderange]
```

For example:

```
wcons node1,node2,node8,node20
```

```
wcons node14-node56,node70-node203
```

```
wcons node1,node2,node8,node20,node14-node56,node70-node203
```

```
wcons all -node129-node256
```

Use **kill** to quickly get rid all your **wcons** panels.

To change the font size: shift right click in the center of the panel.

Example

```
# wcons node001-node010
```

(Nodes 4 and 5 are installing. The rest of the nodes are at a login.)



Figure A-1 wcons output

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| ts.tab | Terminal Server definition table. |

Environment

| | |
|----------------|--------------|
| DISPLAY | Your DISPLAY |
|----------------|--------------|

Diagnostics

| | |
|-------------------------------|--|
| DISPLAY not set | DISPLAY environmental variable is not set. |
| xyz is not a node | Returned if node name not in nodelist.tab. |
| ts:port is unavailable | Returned if port is in use by another user. ts = Terminal Server host name, port = 1-16. |
| telnet to xyz failed | te1net command timed out. Check Ethernet cable and IP. Check DNS and ts.tab. |

See also

noderange
rcons
wkill

rvid - Remote video

Description

rvid redirects the console text video. Useful for debugging boot issues.

rvid differs from **rcons** in that **rvid** is slow and is not always interactive.

wvid is preferred.

Note: While viewing remote video, no other Service Processor functions may be performed to any other node sharing the same Service Processor Network.

Synopsis

```
rvid [singlenode] [boot|noboort]
```

For example:

```
rvid node23
```

Options

| | |
|----------------|--|
| boot | Forces reset or power on before redirecting video. |
| noboort | Does not force reset or power on before redirecting video. |

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |

Remote session timeout `telnet` was successful but communication with node failed. Wait one minute and try again.

See also

`wvid`

wvid - Remote video

Description

wvid redirects the console text video. **wvid** is functionally equivalent to **rvid**, except that **wvid** will launch an xterm with geometry 80x25 and a VGA font for the remote video display, as shown in Figure A-2. Useful for debugging boot issues.

wvid differs from **wcons** in that **wvid** is slow, is not always interactive, and operates on a single node.

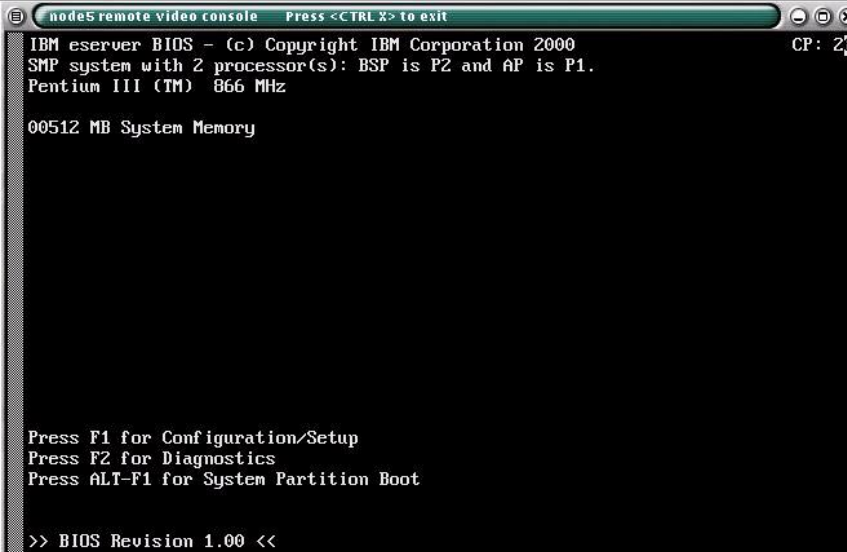
Note: While viewing remote video, no other Service Processor functions may be performed to any other node sharing the same Service Processor Network.

Synopsis

```
wvid [singlenode] [boot|noboot]
For example:
wvid node23
```

Example

```
# wvid node5 boot
```

A screenshot of a terminal window titled "node5 remote video console" with a subtitle "Press <CTRL X> to exit". The terminal displays the following text:

```
IBM eserver BIOS - (c) Copyright IBM Corporation 2000
SMP system with 2 processor(s): BSP is P2 and AP is P1.
Pentium III (TM) 866 MHz

00512 MB System Memory

Press F1 for Configuration/Setup
Press F2 for Diagnostics
Press ALT-F1 for System Partition Boot

>> BIOS Revision 1.00 <<
```

Figure A-2 wvid output

Options

| | |
|---------------|--|
| boot | Forced reset or power on before redirecting video. |
| noboot | Does not force reset or power on before redirecting video. |

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |
| Remote session timeout | telnet was successful but communication with node failed. Wait one minute and try again. |

See also

rvid

rvitals - Remote vitals

Description

rvitals retrieves hardware vital information from the on-board Service Processor for a range of nodes, such as fan speed and temperature.

Synopsis

```
rvitals [noderange]
[cputemp|disktemp|ambtemp|temp|voltage|fanspeed|power|powertime|reboots|state|all]
For example:
rvitals node1,node2,node8,node20 temp
rvitals node14-node56,node70-node203 temp
rvitals node1,node2,node8,node20,node14-node56,node70-node203 temp
rvitals all -node129-node256 all
```

Example

```
# rvitals node5 all
node5: CPU 1 Temperature: + 26.00 C (+ 78.8 F)
node5: CPU 2 Temperature: + 16.00 C (+ 60.8 F)
node5: DASD Sensor 1 Temperature: + 30.00 C (+ 86.0 F)
node5: System Ambient Temperature Temperature: + 23.00 C (+ 73.4 F)
node5: +5V Voltage: + 5.01V
node5: +3V Voltage: + 3.31V
node5: +12V Voltage: + 11.98V
node5: +2.5V Voltage: + 2.52V
node5: VRM1 Voltage: + 1.67V
node5: VRM2 Voltage: + 1.68V
node5: Fan 1 Percent of max:      86%
node5: Fan 2 Percent of max:      86%
node5: Fan 3 Percent of max:      93%
node5: Fan 4 Percent of max:      91%
node5: Current Power Status On
node5: Power On Seconds          2239
node5: Number of Reboots         4
node5: System State Booted Flash or System partition
```

Options

| | |
|-----------------|---------------------------------------|
| cputemp | Retrieves CPU temperatures. |
| disktemp | Retrieves HD back plane temperatures. |
| ambtemp | Retrieves ambient temperature. |

| | |
|------------------|---|
| temp | Retrieves all temperatures. |
| voltage | Retrieves power supply and VRM voltage reading. |
| fanspeed | Retrieves fan speeds. |
| power | Retrieves power status. |
| powertime | Retrieves number of total power uptime. This value only goes up and does not get reset. |
| reboots | Retrieves total number of reboots. This value only goes up and does not get reset. |
| state | Retrieves system state. |
| all | Retrieves all information. |

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |
| Remote session timeout | telnet was successful but communication with node failed. Wait one minute and try again. |

See also

noderange

reventlog - Remote hardware event logs

Description

reventlog can display any number of remote hardware event log entries or clear them for a range of nodes.

Synopsis

```
reventlog [noderange] [number of entries|all|clear]
For example:
reventlog node1,node2,node8,node20 10
reventlog node14-node56,node70-node203 10
reventlog node1,node2,node8,node20,node14-node56,node70-node203 all
reventlog all -node129-node256 clear
```

Examples

```
# reventlog node4,node5 5
node4: SERVPROC I 09/06/00 15:23:33 Remote Login Successful User ID =
USERID[00]
node4: SERVPROC I 09/06/00 15:23:32 System spn1 started a RS485 connection with
us[00]
node4: SERVPROC I 09/06/00 15:22:35 RS485 connection to system spn1 has
ended[00]
node4: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User ID =
USERID[00]
node4: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection with
us[00]
node5: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User ID =
USERID[00]
node5: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection with
us[00]
node5: SERVPROC I 09/06/00 15:21:34 RS485 connection to system spn1 has
ended[00]
node5: SERVPROC I 09/06/00 15:21:30 Remote Login Successful User ID =
USERID[00]
node5: SERVPROC I 09/06/00 15:21:29 System spn1 started a RS485 connection with
us[00]
# reventlog all clear
node4: clear
node5: clear
```

Options

| | |
|--------------|--------------------------------------|
| n | Retrieve <i>n</i> number of entries. |
| all | Retrieve all entries. |
| clear | Clear event logs. |

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |
| Remote session timeout | telnet was successful but communication with node failed. Wait one minute and try again. |

See also

noderange

rinv - Remote hardware inventory

Description

rinv retrieves hardware configuration information from the on-board Service Processor for a range of nodes.

Note: Currently only x330 nodes support this feature. Support for other node models is forthcoming.

Synopsis

```
rinv [noderange] [pci|config|model|serial|all]
For example:
rinv node1,node2,node8,node20 config
rinv node14-node56,node70-node203 config
rinv node1,node2,node8,node20,node14-node56,node70-node203 config
rinv all -node129-node256 all
```

Example

```
# rinv node5 all
node5: Machine Type/Model 865431Z
node5: Serial Number 23C5119
node5: PCI Information
node5: Bus  VendID  DevID  RevID  Description  Slot
Pass/Fail
node5: 0  1166  0009  06  Host Bridge  0
PASS
node5: 0  1166  0009  06  Host Bridge  0
PASS
node5: 0  5333  8A22  04  VGA Compatible Controller  0
PASS
node5: 0  8086  1229  08  Ethernet Controller  0
PASS
node5: 0  8086  1229  08  Ethernet Controller  0
PASS
node5: 0  1166  0200  50  ISA Bridge  0
PASS
node5: 0  1166  0211  00  IDE Controller  0
PASS
node5: 0  1166  0220  04  Universal Serial Bus  0
PASS
```

```

node5: 1 9005 008F 02 SCSI Bus Controller 0
PASS
node5: 1 1014 00DC 02 Bridge Device 1
PASS
node5: 1 14C1 8043 03 Unknown Device Type 2
PASS
node5: Machine Configuration Info
node5: Number of Processors: 2
node5: Processor Speed: 866 MHz
node5: Total Memory: 512 MB
node5: Memory DIMM locations: Slot(s) 1 2

```

Options

| | |
|---------------|---|
| pci | Retrieves PCI bus information. |
| config | Retrieves number of processors, speed, total memory, and DIMM location. |
| model | Retrieves model number. |
| serial | Retrieves serial number. |
| all | Retrieves all information. |

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--|---|
| inventory functions not supported | Returned if not a x330 node. |
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |

Remote session timeout

telnet was successful but communication with node failed. Wait one minute and try again.

See also

noderange

psh - Parallel remote shell

Description

psh is a utility used to run a command across a list of nodes in parallel. **psh** relies on the **rsh** field in **site.tab** to determine whether to use **rsh**, **ssh**, or any other method to launch a remote shell. **rsh**, **ssh**, or any other method must be set up to allow no prompting (that is **.rhosts** for **rsh** and **sshd_config** and **.rhosts** for **ssh**) for **psh** to work.

Synopsis

```
psh [noderange] -c [command]
For example:
psh node1,node2,node8,node20 -c uptime
psh node14-node56,node70-node203 -c uptime
psh node1,node2,node8,node20,node14-node56,node70-node203 -c uptime
psh all -node129-node256 -c uptime
```

Examples

```
# psh node1-node10 -c uptime
Get the uptime for nodes 1 through 10 returned in order of completion.
# psh all -c /sbin/halt
Shutdown down all nodes listed in nodelist.tab.
Enclosing the command in ' (single quotes) will forward any redirection to the
nodes. For example:
# psh all -c 'grep processor /proc/cpuinfo | wc -l' | sort
Will return a list of nodes with the number of processors per node sorted by
node.
# psh all -c grep processor /proc/cpuinfo | wc -l
Will return the total number of processors in the cluster.
```

Options

-c Required option. **-c** is followed by the command you wish to execute on the range of nodes specified.

Files

nodelist.tab Only nodes in this table will be affected by this command.
site.tab Used to determine **rsh** command.

Diagnostics

If **psh** is unable to ping the remote node(s), it will return a noping error. The **rsh** method may also return errors.

See also

noderange

pping - Parallel ping

Description

pping is a utility used to ping a list of nodes in parallel. **pping** will return an unsorted list of nodes with a ping or noping status.

Synopsis

```
pping [noderange]
For example:
pping node1,node2,node8,node20
pping node14-node56,node70-node203
pping node1,node2,node8,node20,node14-node56,node70-node203
pping all -node129-node256
```

Examples

```
# pping node1-node10
Ping nodes 1 through 10.
# pping all -node5-node7
Ping all nodes listed in nodelist.tab less nodes 5 through 7.
```

Files

nodelist.tab Only nodes in this table will be affected by this command.

See also

noderange

rinstall - Remote network Linux install

Description

rinstall will force an unattended network install for a range of nodes. **nodeset** is called to set the install tftp flag. If a node is off, it will be powered on. If a node is on, it will be reset.

rinstall returns nodename: powerstate action, where powerstate is on or off and action is on or reset for each node.

Synopsis

```
rinstall [noderange]
For example:
rinstall node1,node2,node8,node20
rinstall node14-node56,node70-node203
rinstall node1,node2,node8,node20,node14-node56,node70-node203
rinstall all -node129-node256
```

Example

```
# rinstall node4,node5
node4: off on
node5: on reset
node4 was off then powered on, node5 was on then reset.
```

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |

Diagnostics

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |

Remote session timeout `telnet` was successful but communication with node failed. Wait one minute and try again.

See also

`noderange`

`nodeset`

winstall - Windowed remote network Linux install

Description

winstall will force an unattended network install for a range of nodes. **winstall** passes its arguments to **rinstall** and **wcons**.

winstall is functionally equivalent to **rinstall**, except that **wcons** will launch an xterm panel for the remote console display, as shown in Figure A-3 on page 164. As the number of nodes requested increase the panels will get smaller.

Use **wkill** to quickly get rid all your **wcons** panels.

To change the font size, Shift-right-click in the center of the panel.

winstall returns nodename: powerstate action, where powerstate is on or off and action is on or reset for each node.

Synopsis

```
winstall [noderange]
For example:
winstall node1,node2,node8,node20
winstall node14-node56,node70-node203
winstall node1,node2,node8,node20,node14-node56,node70-node203
winstall all -node129-node256
```

Example

```
# winstall node4,node5
node4: off on
node5: on reset
node4 was off then powered on, node5 was on then reset.
```

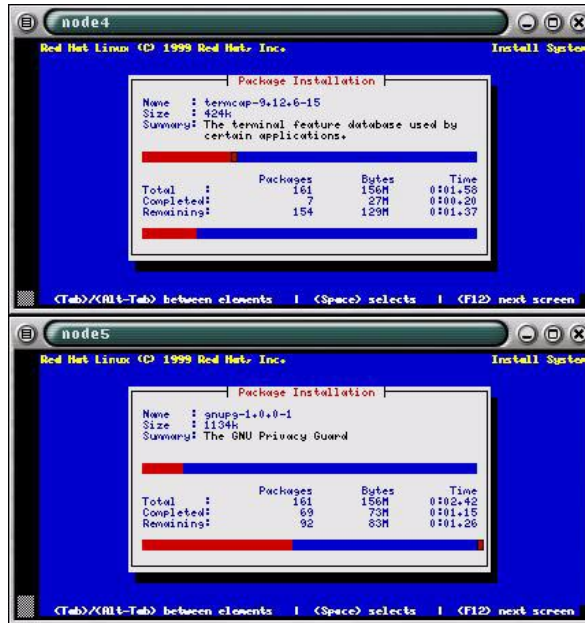


Figure A-3 winstall output

Files

| | |
|---------------------|--|
| nodelist.tab | Only nodes in this table will be affected by this command. |
| sp.tab | Service Processor Network definition table. |
| ts.tab | Terminal Server definition table. |

Environment

| | |
|----------------|--------------|
| DISPLAY | Your DISPLAY |
|----------------|--------------|

Diagnostics

| | |
|-------------------------------|--|
| DISPLAY not set | DISPLAY environmental variable is not set. |
| xyz is not a node | Returned if node name not in nodelist.tab. |
| ts:port is unavailable | Returned is port is in use by another user. ts = Terminal Server host name, port = 1-16. |
| telnet to xyz failed | telnet command timed out. Check Ethernet cable and IP. Check DNS and ts.tab. |

| | |
|--------------------------------|---|
| unknown spn xyz | Returned if spn name (xyz) in sp.tab cannot be resolved by DNS. |
| Connection refused | Returned if telnet fails. PCI ASMA may be in use or telnet interface is locked up. spnreset can be used to reboot the PCI ASMA. |
| telnet to xyz failed | telnet command timed out. PCI ASMA adapter may not have power. Check Ethernet cable and IP. |
| node xxx not on spn yyy | xxx is not listed in SPN yyy. Use spncheck yyy to verify. Check cabling and try spnreset . |
| Remote session timeout | telnet was successful but communication with node failed. Wait one minute and try again. |

See also

noderange

nodeset

rinstall

wcons

wkill



xCAT configuration files

This appendix describes the tables used by xCAT during the configuration and installation process. These tables must be updated before starting the nodes installation.

Table B-1 provides a quick reference of the xCAT tables. We also provide a full description of each xCAT table in this appendix.

Table B-1 xCAT tables

| Table | Description |
|--------------|--|
| site.tab | This table is used to configure the cluster network. It includes a list of parameters which have to be updated for your cluster configuration. |
| nodelist.tab | This table describe all the nodes and gives each of them a group name, defines what kind of node it is, and on which ASMA card the node is connected. |
| noderes.tab | This table describes where the compute nodes can find the services they need. The line describes which server the group will find the resources it needs for each group. |

| Table | Description |
|--------------|---|
| nodetype.tab | This table contains, for each node, one line which describes the name of the Kickstart file to use for the installation of each node. |
| nodehm.tab | This table contains information on how to use each node. As the material used to install a cluster could be different from one installation to another one, this file gives xCAT commands the flexibility to adapt themselves to your environment. |
| asma.tab | This table describes which paths the xCAT commands have to follow to communicate with the service processor of each node. |
| mac.tab | This table will be generated by the getmacs script. It contains one line per node, for each node, which is the MAC address used for the management for this node. |
| passwd.tab | This table will be used during the installation process and to perform some operations on the nodes, like power off or power on through the ASMA card. It contains two fields: the text we are waiting for, and what we have to type in. |
| apc.tab | This table gives xCAT commands the ability to power on or power off pieces of equipment that are not controlled by the ASMA/ASM card (network switch, Equinox, Myrinet switch, and so on). It works in conjunction with the APC MasterSwitch to control the electrical power on specified hardware. |

site.tab

This table is used to configure the cluster network. It includes a list of parameters (shown in Table B-2), which have to be updated for your cluster configuration.

Table B-2 Definitions of *site.tab* parameters

| Parameter | Description | Sample |
|--------------|---|-------------------------|
| rsh | Command used to open a connection on a compute node. | /usr/bin/ssh |
| rcp | Command used to do a remote copy. | /usr/bin/scp |
| tftpdirdir | Directory used by the tftp daemon. | /tftpboot |
| tftpxcatroot | Directory under tftpdirdir, used for xCAT files. | xcat |
| domain | DNS domain used for the cluster. | cluster.austin.ibm.com |
| nameservers | IP address of the name server, often the IP address of a master node | 192.168.0.254 |
| net | This is where the network:netmask is used by the nodes for the cluster management virtual local area network (VLAN). Only this VLAN will be resolved by the DNS sever installed on the management node. | 192.168.0:255.255.255.0 |
| gkhfile | The file that will contain the ssh host keys for the nodes. | /usr/local/xcat/etc/gkh |
| master | Name of the master node. | master |
| pbshome | Value of PBS_HOME. See PBS documentation. | /var/spool/pbs |
| pbsprefix | Value of PBS_PREFIX. See PBS documentation. | /usr/local/pbs |
| pbsserver | Name of the node which is running the PBS server. | master |

| Parameter | Description | Sample |
|-------------|--|---|
| scheduler | Name of the node which is running the Maui scheduler. | master |
| nisdomain | NIS domain if using NIS; NA if not available. | NA |
| nismaster | Name of the master node if you use NIS; NA if not available. | NA |
| xcatprefix | Directory where xCAT files reside. | /usr/local/xcat |
| timezone | Name of the timezone used for the cluster. | US/Central |
| offutc | Time difference between UTC and local time (Should be consistent with timezone parameter). | -6 |
| mapperhost | Host that runs the GM mapper daemon. | node1 |
| clustervlan | Name of the network card on the management node connected to the compute nodes. | eth1 |
| serialmac | Console serial port number for MAC address collection; represents N in /dev/ttySN. | This value can be one of the following: 0 for COM A/1 1 for COM B/2 NA for no MAC collection |
| dynamicb | Private class B network used when getting MAC addresses. | 172.16 |

Note: (For net stanza above) The network must be truncated based on the netmask. For instance:

```
192.168.0:255.255.255.0 CORRECT!
192.168:255.255.0.0 CORRECT!
192.168.0.0:255.255.255.0 INCORRECT!
192.168:255.255.255.0 INCORRECT!
```


nodelist.tab

This table describes all the nodes and gives each of them a group name, describes what kind of node it is, and on which ASMA card the node is connected.

The table has one line per node, as shown in Example B-1.

Example: B-1 Description of nodelist.tab table

| | |
|----------|-------------------------------------|
| NodeName | GroupName,GroupName2,...,GroupNameN |
|----------|-------------------------------------|

Table B-3 gives the definition of the nodelist.tab parameters.

Table B-3 Definitions of nodelist.tab parameters

| Parameter | Description | Possible values |
|------------|---|---|
| NodeName | This is the node name, based on the naming convention used in the cluster. | nodeX, where X is the node number. node is not mandatory; it can be changed to whatever you want, as long as it begins with non-numeric characters. |
| GroupNameN | You can include the node in N group. The groups are used to send commands on multiple node at the same time. For example, rinv rack1 instead of rinv node1,node2,node3 | Whatever you want, as long as it begins with non-numeric characters. (that is, frame01, frame02, if the nodes are in different frames.) It is recommended to create a group which includes all the cluster's nodes. |

noderes.tab

This table describes where the compute nodes can find the services they need.

The line describes which server the group will find the resources it needs, as shown in Example B-2.

Example: B-2 Description of noderes.tab table

```
GroupName  Tftp,NFS_Inst,NFS_Dir,Post_Dir,NIS_srv,Time,Serial,InstallRoll
```

Table B-4 lists the definitions of the noderes.tab parameters.

Table B-4 Definitions of noderes.tab parameters

| Parameter | Description | Possible values |
|-----------|--|---|
| GroupName | Node or group name the parameters apply to. | The group or node should exist in the nodelist.tab table. |
| Tftp | Name of the tftp server to be used for the installation and boot processes. | Usually the host name of the master node, unless multiple install servers are used. |
| NFS_Inst | Name of the NFS server where the Red Hat CD-ROM and other install files reside. | Usually the host name of the master node, unless multiple install servers are used. |
| NFS_Dir | Path name of the NFS exported Red Hat CD-ROM on NFS_Inst server. | By default, the installation directory is the following: /install/rh62. |
| Post_Dir | Path name of the NFS exported xCAT post install directory on NFS_Inst server | By default, the post installation directory is the following: /install/post. |
| NIS_srv | Host name of NIS master. | NIS server name; NA if you do not use NIS. |
| Time | Host name of the timeserver. Unless your timeservers are synchronized, this should be the same for all nodes in the cluster. | Usually the host name of the master node. |

| Parameter | Description | Possible values |
|-------------|--|---|
| Serial | Console serial port number. Represents N in /dev/ttySN | This value can be one of the following: 0 for COM A/1 1 for COM B/2 NA for no serial console |
| InstallRoll | Indicates if this node/group will be used as the installation node. This is only needed on large clusters where a small number of installation nodes are installed first and then used as install servers for remaining nodes. | The value can be only Y or N. |

nodetype.tab

This table contains one line for each node and describes the name of the Kickstart file to use for its installation.

With this table, it is possible to have different nodes in the same cluster. For example, some installation nodes could have some installed services that are not running on the compute node. Example B-3 gives an example description of the table.

Example: B-3 Description of nodetype.tab table

| | |
|----------|-------------------|
| NodeName | KickstartFileName |
|----------|-------------------|

Table B-5 lists the definitions of the nodetype.tab parameters.

Table B-5 Definitions of nodetype.tab parameters

| Parameter | Description | Possible values |
|-------------------|--|--|
| NodeName | Name of the node this applies to. | nodeX, where X is the node number. node is not mandatory; it can be changed to whatever you want, as long as it begins with non-numeric characters. |
| KickstartFileName | Name of the Kickstart template file used by xCAT to install this node. | The short name of a kstmpl file in /usr/local/xcat/ks62. By default, either compute62 or user62. |

nodehm.tab

This table contains information on how to use each node. As the material used to install a cluster could be different from one installation to another, this file gives xCAT commands the flexibility to adapt themselves to your environment.

The line describe how to do some management functions on the node for each node, as shown in Example B-4.

Example: B-4 Description of nodehm.tab table

```
NodeName    Power,Reset,Cad,Vitals,Inv,Cons,BiosCons,EventLogs,GetMacs
```

Table B-6 lists the definitions of the nodehm.tab parameters.

Table B-6 definition of nodehm.tab parameters

| Parameter | Description | Possible values |
|-----------|--|---|
| NodeName | This is the node name, based on the name convention used in the cluster. | nodeX, where X is the node number. node is not mandatory; it can be changed to whatever you want, as long as it begins with non-numeric characters. |
| Power | The method used by the rpower command to control the power to the node. | NA: The node is not manageable. asma: The rpower command will use the ASMA card to manage the node. apc: The rpower command will use the APC MasterSwitch to manage the node. |

| Parameter | Description | Possible values |
|-----------|--|--|
| Reset | The method used by the rreset command to reset the node. | <p>NA: The node is not manageable.</p> <p>asma: The rreset command will use the ASMA card to manage the node.</p> <p>apc: The rreset command will use the APC MasterSwitch to manage the node.</p> |
| Cad | The method used by the rcad command to send Control+Alt+Delete to the node. | <p>NA: The node is not manageable.</p> <p>asma: The rcad command will use the ASMA card to send the cad keys to a node.</p> |
| Vitals | The method used by the rvitals command to query the node for its vitals sensors, such as CPU, disk, and ambient temperature, voltages, fan speeds, and so on. | <p>NA: The node is not manageable.</p> <p>asma: The rvitals command will use the ASMA card to obtain the information from the node.</p> |
| Inv | The method used by the rinv command to query the node about its inventory data, such as machine model, serial number, and so on. | <p>NA: The node is not manageable.</p> <p>asma: The rinv command will use the ASMA card to obtain the information from the node.</p> |

| Parameter | Description | Possible values |
|-----------|---|--|
| Cons | The method used by the rcons and wcons commands to obtain a remote serial console for the node. | <p>NA: The node is not manageable.</p> <p>conserver: The commands will use <code>conserver</code> to obtain a console for the node.</p> <p>rtel: The commands will use <code>telnet</code> to obtain a console for the node.</p> <p>tty: The commands will use a local <code>tty</code> to obtain a console from the node.</p> |
| BiosCons | The method used by the rvid and wvid commands to open a boot console for the node. | <p>NA: The node is not manageable.</p> <p>asma: The rvid and wvid commands will use the ASMA card to obtain a boot console from the node.</p> |
| EventLogs | The method used by the reventlog command to query and manage the ASM event log for the node. | <p>NA: The node is not manageable.</p> <p>asma: The reventlog command will use the ASMA card to obtain and manage the event log.</p> |

| Parameter | Description | Possible values |
|-----------|--|---|
| GetMacs | The method used by the getmacs command to obtain the MAC address of the node. | <p>NA: The node is not manageable; manual collection only.</p> <p>rcons: The getmacs command will use a remote console to obtain the MAC address of this node.</p> <p>Unsupported options:</p> <p>cisco3500: The getmacs command will query a Cisco 3500 switch to obtain the MAC address of this node.</p> <p>extreme: The getmacs command will query an Extreme network switch to obtain the MAC address of this node.</p> |

asma.tab

This table describes which paths the xCAT commands have to follow to communicate with the service processor of each node.

The table gives, for each node, the ASMA card and the valid service processor name. The description of the table is given in Example B-5.

Example: B-5 Description of asma.tab table

NodeName AsmaCardName, SvcProcName

Table B-7 gives the definitions of the asma.tab parameters.

Table B-7 Definitions of asma.tab parameters

| Parameter | Description | Possible values |
|--------------|--|--|
| NodeName | This is the node name, based on the name convention used in the cluster. | nodeX, where X is the node number. node is not mandatory; it can be changed to whatever you want, as long as it begins with non-numeric characters. |
| AsmaCardName | The IP host name of the ASMA card to which this node is attached. | The value should be the alias name of the asma card, as described in the /etc/hosts file. |
| SvcProcName | The service processor name of the node. | The nodes are automatically configured so that it is the same as the NodeName. |

mac.tab

This table will be generated by the getmacs script. It contains one line per node and describes the MAC address used for each node.

Do not edit this file yourself.

Example B-6 has a description of the mac.tab table.

Example: B-6 Description of mac.tab table

| | |
|----------|-------------------|
| nodeName | MacAddressMgtCard |
|----------|-------------------|

Table B-8 gives the definitions of the mac.tab.parameters.

Table B-8 Definitions of mac.tab parameters

| Parameter | Description | Possible values |
|-------------------|--|--|
| nodeName | This is the node name, based on the name convention used in the cluster. | nodeX, where X is the node number. node is not mandatory; it can be changed to whatever you want, as long as it begins with non-numeric characters. |
| MacAddressMgtCard | The MAC address of the management ethernet adapter (eth0) in the node. | The card's MAC address. |

passwd.tab

This table is used during the installation process and to perform operations on the nodes, like power off or power on, through the ASMA card. It contains two fields: the text we are waiting for, and what we have to type in.

Example B-7 has a description of the passwd.tab table.

Example: B-7 Description of passwd.tab table

| | |
|----------------|---------------|
| PasswordNeeded | PasswordValue |
|----------------|---------------|

Table B-9 gives the definition of the passwd.tab parameters.

Table B-9 Definitions of passwd.tab parameters

| Parameter | Description | Possible values |
|----------------|---|--|
| PasswordNeeded | This field describes the value we are looking for when we have to type in a password. | Values for this field are not defined because they depend on your hardware, but you need at least one line for the connection to the node. For the connection to the node, you need this line: rootpw If you use ASMA and/or ASM hardware, you need to put two lines in this file: asmauser asmapass If you use Cisco products, you should have a line: cisco |

| Parameter | Description | Possible values |
|---------------|---|--|
| PasswordValue | This field contains the value you need to type in when a password is asked. | <p>Again, the content of this field will depend on the value used to install the hardware.</p> <p>By default, for the ASMA/ASM hardware, the values for asmauser and asmapass are: USERID PASSWORD</p> <p>The value for the rootpw is the password you select during the Red Hat installation process.</p> |

apc.tab

This table gives the xCAT commands the ability to power on or power off some equipment not controlled by the ASMA/ASM card (network switch, Equinox, Myrinet switch, and so on). It works in conjunction with APC MasterSwitch to control the electrical power on specified hardware.

If your nodes are not using the ASMA hardware but they are connected to an APC MasterSwitch, then you still are able to control them via this table.

Example B-8 contains a description of the apc.tab table.

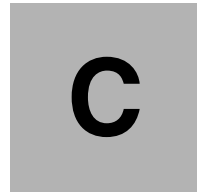
Example: B-8 Description of apc.tab table

| | |
|-----------|--------------------------|
| Equipment | ApcToUse,1,ApcPortNumber |
|-----------|--------------------------|

Table B-10 gives the definitions of the apc.tab parameters.

Table B-10 Definitions of apc.tab parameters

| Parameter | Description | Possible values |
|---------------|--|--|
| Equipment | This is the name used with the rpower or rreset commands. | The name should be compliant with the name convention chosen before. For the node, or a network router, it can be the following: nodeX routerY Where X,Y represent the node or the router number. |
| ApcToUse | This field is the APC alias which has been chosen in the /etc/hosts table for this hardware. | Insert the alias name for the APC MasterSwitch. For example: apc1 |
| ApcPortNumber | This field contains the port number where the equipment is plug in. | Insert the port number that controls the equipment. |



POVRay test

To test and demonstrate the power of a cluster, we decided to use a program called POVRay (Persistence of Vision). It is a powerful rendering application, it is free, and the source code is distributed.

This appendix describes how to install and execute the POVRay program and it is structured as follows:

- ▶ POVRay Install and Patch
- ▶ POVRay test with TCP/IP
- ▶ POVRay test with MYRINET GM

POVRay Installation

To make the POV-Ray cluster “aware,” we patch the source code using a mpipov patch and then compile it. The patch enables POV-Ray to operate parallel to MPI.

Downloads

- ▶ Download the POV-Ray source code from:
<http://www.povray.org/binaries/linux.html>
Download the files povuni_s.tgz and povuni_d.tgz. The size of the files are under 1 MB each.
- ▶ Download the POVMPI Patch from:
<http://www.verrall.demon.co.uk/mpipov/>
Download the file mpi-povray-1.0.patch.gz. The size of the file is about 127 KB

Install

To install POV-Ray, use these commands:

```
[root@master ]# cp "Path to POV-Ray Download Dir"/povuni_* /usr/local
[root@master ]# cd /usr/local
[root@master local]# tar zxfv povuni_s.tgz
[root@master local]# tar zxfv povuni_d.tgz
[root@master local]# cd povray31
[root@master povray31]# cp "Path to POV-Ray Download
Dir"/mpi-povray-1.0.patch.gz .
[root@master povray31]# gunzip mpi-povray-1.0.patch.gz
[root@master povray31]# cat mpi-povray-1.0.patch | patch -p1
[root@master povray31]# cd source/mpi-unix
```

Note: You might need to add the path to mpicc to the \$PATH variable. In our case, mpicc resides in /usr/local/mpich/1.2.1/ip/smp/gnu/ssh/bin/, so you would use the commands:

```
[root@master mpi-unix]# export
PATH=/usr/local/mpich/1.2.1/ip/smp/gnu/ssh/bin/:$PATH
[root@master mpi-unix]# make newxwin
```

Edit povray.ini file line : Library path to match the correct path.

POVRay using TCP/IP and GCC

TEST

1. Find a couple of .pov files to be rendered from /usr/local/povray31/scenes/advanced/, for example, skyvase.pov and chess2.pov.
2. Copy them to /home/ibm/, or /root, or any directory that is accessible from all nodes. Another option is to use the actual NFS shared/mounted /usr/local/ directory, and then launch the mpi-x-povray. This is what we have done in our lab:

```
[root@master mpi-unix]# psh all cp
/usr/local/povray31/scenes/advanced/skyvase.pov /root
[root@master mpi-unix]# cp /usr/local/povray31/scenes/advanced/skyvase.pov
/root
```

3. Check which mpirun you are running. This is found on the first path; fix the path accordingly. You do not want to run the wrong mpirun, such as the one that is compiled to support TCP/IP or GM. This is what we have done in our lab:

```
[root@master mpi-unix]# cd
[root@master root]# mpirun -np 6
/usr/local/povray31/source/mpi-unix/mpi-x-povray -I/root/chess2.pov +v1 +ft
-x +mb25 +a0.300 +j1.000 +r3 -q9 -w640 -H480 -S1 -E480 -k0.000 -mv2.0
+b1000 +L/usr/local/povray31/include/ +NH128 +NW128
```

RESULT

The results of the test is summarized here:

PE Distribution Statistics:

```
Done Tracing Slave PE [ done ]
    1 [10.67%]
    2 [10.67%]
    3 [18.67%]
    4 [14.67%]
    5 [20.00%]
    6 [14.67%]
    7 [10.67%]
```

POV-Ray statistics for finished frames:

skyvase.pov Statistics (Partial Image Rendered), Resolution 640 x 480

```
-----
Pixels:          45440  Samples:          56240  Smp1s/Px1: 1.24
Rays:           276464  Saved:           3272  Max Level: 0/5
-----
```

```
-----
Ray->Shape Intersection      Tests      Succeeded  Percentage
-----
```

| | | | |
|-------------------|-------------------|------------------|-------------|
| CSG Intersection | 857640 | 116916 | 13.63 |
| Plane | 5145840 | 2742574 | 53.30 |
| Quadric | 857640 | 594915 | 69.37 |
| Sphere | 857640 | 208399 | 24.30 |
| ----- | | | |
| Calls to Noise: | 496771 | Calls to DNoise: | 815936 |
| ----- | | | |
| Shadow Ray Tests: | 933768 | Succeeded: | 40903 |
| Reflected Rays: | 220224 | | |
| ----- | | | |
| Smallest Alloc: | 26 bytes | Largest: | 1024008 |
| Peak memory used: | 4979073 bytes | | |
| ----- | | | |
| Time For Trace: | 0 hours 0 minutes | 8.0 seconds | (8 seconds) |
| Total Time: | 0 hours 0 minutes | 8.0 seconds | (8 seconds) |

For a more reliable test, run the chess2.pov scene. It produces a practical, nonscientific, visual performance test that ensures MPICH/GM is working properly throughout the cluster. In addition, POVray can be configured to show the graphical image process as it renders block by block. For more information about POVray, please refer to the following Web site:

<http://www.povray.org/>

POV-Ray using Myrinet and PGI

TEST

1. Find a couple .pov files to be rendered from /usr/local/povray31/scenes/advanced/, for example, skyvase.pov and chess2.pov.
2. Copy them to /home/ibm/ or /root or any directory that is accessible from all nodes; a third option is to use the actual NFS shared/mounted /usr/local/... directory, and then launch the mpi-x-povray. This is what we have done in our lab:

```
[root@master mpi-unix]# psh all cp
/usr/local/povray31/scenes/advanced/skyvase.pov /root
[root@master mpi-unix]# cp /usr/local/povray31/scenes/advanced/skyvase.pov
/root
```

3. Check which mpirun you are running. This is found on the first path; fix the path accordingly. You do not want to run the wrong mpirun, such as the one that is compiled to support TCP/IP or GM. This is what we have done in our lab:

```
[root@master mpi-unix]# cd
[root@master root]# mpirun -np 6
/usr/local/povray31/source/mpi-unix/mpi-x-povray -I/root/chess2.pov +v1 +ft
-x +mb25 +a0.300 +j1.000 +r3 -q9 -w640 -H480 -S1 -E480 -k0.000 -mv2.0
+b1000 +L/usr/local/povray31/include/ +NH128 +NW128
```

The following lines describe the execution of skyvase.pov:

```
[root@master mpi-unix]# cd /usr/local/povray31/scenes/advanced
mpirun -np 8 /usr/local/povray31/source/mpi-unix/mpi-x-povray -Iskyvase.pov
+v1 +ft -x +mb25 +a0.300 +j1.000 +r3 -q9 -w640 -H480 -S1 -E480 -k0.000
-mv2.0 +b1000 +L/usr/local/povray31/include/ +NH128 +NW128
```

RESULT

The results of the test is summarized here:

PE Distribution Statistics:

| Slave PE | [done] | Slave PE | [done] |
|----------|------------|----------|------------|
| 1 | [8.81%] | 3 | [11.25%] |
| 2 | [12.75%] | 5 | [11.25%] |
| 4 | [10.12%] | 7 | [12.56%] |
| 6 | [11.25%] | 9 | [10.88%] |
| 8 | [11.12%] | | |

POV-Ray statistics for finished frames:

skyvase.pov Statistics (Partial Image Rendered), Resolution 640 x 480

```

-----
-
Pixels:          27496  Samples:          32376  SmpIs/Pxl: 1.18
Rays:           134410  Saved:            561  Max Level: 0/5
-----
-
Ray->Shape Intersection      Tests      Succeeded  Percentage
-----
-
CSG Intersection            420954      40935      9.72
Plane                       2525724     1339496    53.03
Quadric                     420954      226872     53.89
Sphere                      420954      65914      15.66
-----
-
Calls to Noise:            187278  Calls to DNoise:      307718
-----
-
Shadow Ray Tests:          459768  Succeeded:            7895
Reflected Rays:           102034
-----
-
Smallest Alloc:            37 bytes  Largest:            1024008
Peak memory used:          2734675 bytes
-----
-
Time For Trace:   0 hours  0 minutes  6.0 seconds (6 seconds)
Total Time:      0 hours  0 minutes  6.0 seconds (6 seconds)

```

For a more reliable test, run the chess2.pov scene. It produces a practical, nonscientific, visual performance test that ensures MPICH/GM is working properly throughout the cluster. In addition, POVRay can be configured to show the graphical image process as it renders block by block. For more information about POVRay, please refer to the following Web site:

<http://www.povray.org/>

Summary

POVRay results

If you want to compare your POVRay cluster results against other clusters, you can see some of the old skyvase.pov test results at:

<http://www.haveland.com/index.htm?povbench/index.htm>

Since most modern machines have very fast processors, the old test is usually obsolete. For a more complex test, check out some new results from chess2.pov at:

<http://www.tabsnet.com>

For detailed POVRay Rendering info and tips, go to:

<http://www.povray.org>

Other tests

For more complex testing, there is also Linpack, which is a good package to test and benchmark most aspects of a cluster. You can find it at:

<http://www.netlib.org/benchmark/>

For the High-Performance Computing TOP500 list go to:

<http://www.top500.org>



D

Hardware configuration used in our lab

This appendix presents the xSeries Cluster hardware configuration that was used in this redbook and is suitable for the Linux Cluster ITSO Workshop. It lists our lab environment and the IBM and non-IBM parts. While going through the list, please keep in mind that there might be newer or different versions of the parts available.

If you want to use other equipment, make sure it is Linux and xCAT compatible before ordering them. This will ensure easy setup and installation, as well as proper functionality.

Hardware Environment for the Lab

Example D-1 shows the hardware environment used in our lab.

Example: D-1 Hardware environment

| Qty./P/N | Description |
|-----------------------------|--|
| IBM Products | |
| Compute Nodes | |
| 4 865431Y | xSeries 330 1000 256 256/OPEN 24X |
| 4 37L7202 | 18.2GB Ultra160 HDD |
| 4 10K3806 | 866Mhz 133MHz 256K |
| 12 33L3144 | 256MB 133MHz ECC SDRAM RDIMM MEMORY |
| 1 06P4792 | C2T Cable Kit |
| Management Node | |
| 1 86564RY | xSeries 340, 866Mhz, 128Mb |
| 1 19k4630 | 866Mhz 133MHz 256K |
| 4 33L3144 | 256MB 133MHz ECC SDRAM RDIMM MEMORY |
| 1 37L6091 | ServeRAID 4L LVD SCSI Adapter |
| 3 37L7202 | 18.2GB 7200rpm Ultra160 SCSI Hot-Swap SL H |
| 1 34L1501 | Netfinity 10/100 Ethernet PCI Adapter 2 |
| 1 34L0301 | Netfinity Gigabit Ethernet SX Adapter |
| 1 37L6880 | 270W Redundant HS Power Supply |
| Shared Resources | |
| 1 9306200 | Netbay 22 Half Rack |
| 1 3619702 | Netbay 22 Rack Extension Kit |
| 1 9411AG1 | Monitor (flatpanel) |
| 1 37L6888 | Flatpanel rack mount kit |
| 1 09N4291 | 8x2 Console Switch (KVM Switch) |
| 2 94G7447 | Console Cable Set 12ft (to KVM switch) |
| 1 28L3644 | Spacesaver Keyboard/trackpoint |
| 1 28L4707 | Keyboard tray |
| 2 37L6866 | Netbay Universal Voltage PDU |
| 1 01K7209 | ASMA Adapter (Wiseman card) |
| 1 36L9973 | 1M Fibre Channel Cable |
| 1 03K9308 | Short Wave GBIC (Gigabit module) |
| Equinox Products | |
| 1 990209 | Equinox ELS-16 |
| 1 210059 | Micro-Transceiver, AUI (DB-15) to 10BaseT |
| 1 790091 | ELS Rackmount kit |
| 4 210062 | Equinox Serial Adapters |
| 4 690226 | 10' Serial Cables |
| Myrinet Networking Products | |
| 1 M3-E16 | Myrinet2000 3-slot Chassis |
| 1 M3-M | Management Module |
| 4 M3S-CB-5M | Myricom Myrinet LAN cables |

4 M3S-PCI64B-2 Myrinet LAN Card
1 M3SW16-8S Myrinet 8-port Serial modules

Miscellaneous Products

8 3' CAT5 Cables
5 1' CAT5 Cables

Extreme Networks Products

1 13020 Summit24 - Full Layer 3-X



E

Installation experiences

This appendix provides some real life examples of our experiences installing Linux Clusters. This is far less about the technical details of an installation and far more about the experiences we have shared and the fun we had.

One of the first clusters was a small, eight node cluster. This was somewhat of a pilot project and everyone was paraded in to see it. The first impressions were the most notable. Instead of a cluster, we had several tables set up with an unbelievable number of cardboard boxes. Everything was shipped separately, whether it be additional memory, CPUs or the racks themselves. One larger box contained all of the memory modules, but in order to build the cluster, each box had to be opened, the contents inventoried, and the compute and head nodes assembled.

It was very impressive to see all the components waiting to be assembled and obtaining all of those components was a large task. We cannot caution you enough about carefully planning your cluster installation. It's an extremely good idea to assemble your equipment order. Then get a peer to review it. After it is reviewed, you should go through and check it again.

There are a lot of pieces that go into making a High-Performance Computing cluster and a little extra work up front can dramatically ease the burden of assembly later.

It is also very important to track all of the parts and their status. Inevitably, one or more key components will be constrained and you will have problems procuring them. This is not nearly as difficult as it is tedious. We recommend allowing a little extra time for this step and advise you to keep careful track of the inventory. When nearly all of the parts have arrived, categorize the parts that are missing by their importance. This will allow you to focus on obtaining those items most critical to your assembly of the cluster.

We can not say it often enough: You cannot plan often enough when building a cluster. In one 256 node cluster we built, we neglected to move the communication port jumper to the COM2 position. Although that task itself is fairly simple, when multiplied by 256 and with the added burden of removing each cover and possibly a PCI Adapter card, one small mistake was amplified into a major task. Proper planning and attention to detail can avoid a lot of unnecessary effort.

To avoid this particular problem, we devised a staging table, where we have power, a monitor, a keyboard, and a mouse. From here, we can easily open the machine and add the required components. Typically, we would label the node at this point. Then we hook up the machine and flash the various BIOSs. This can certainly be done later, but it is far easier to repair any problems at this stage, with the machine opened up and accessible. Once the machine's BIOSs have been flashed and checked out, we replace the cover and put it in a stack where it will wait for installation into the rack.

Now comes the fun part, and it really is. Everything is still fairly messy, but there's a neatness to assembling the rack that is very satisfying. Instead of haphazard piles of equipment everywhere, a cluster begins to evolve in front of you.

One note on tools: If you do not have one already, we strongly advise you get an electric screwdriver. It is beneficial when removing and replacing the covers, but it is almost required for assembling the equipment in the racks, particularly for larger clusters.

Once completed, your cluster should look something like Figure E-1 on page 199. This picture is actually half of a 256 node cluster. Each of the four racks, two on either side, contain 32 nodes each. The center rack contains the head node(s), the KVM switch and, in this case, some of the Myrinet switches. Notice the perforated tiles in front of the rack. These provide cool air which is pulled through the x330's and vents out the back. For this reason, you *cannot* use a glass cover. We usually just leave the front covers off; however, vented front covers are available.



Figure E-1 256 node cluster

There is another thing to notice. There is exactly one Keyboard, Video Display and Mouse (KVM) for the entire 256 node cluster. This is actually a good thing. The entire cluster has been designed to be easily managed without any physical presence. However, during the build process, this single KVM can be a bottleneck, as everyone crowds around and one person types. You should consider this when planning the installation of your cluster. It is difficult or impossible to double up on tasks and it should be assumed that only one person will be doing the actual installation.

As we have discussed, this is a 256 node cluster and so far we have only seen 128 of the nodes. Figure E-2 on page 200 shows the other 128 nodes. Instead of the console, the center rack contains the Ethernet Switching equipment; the large mass of cables are the Ethernet cables. This particular configuration used a set of patch panels on the bottom of each rack. A standard bundle was used to interconnect each 32 node rack with a patch panel near the switch. The mass of cables shown are the patch cables interconnecting each rack's patch panel with the switch, which is on top. Although this might sound difficult, it allows far better rack staging at another location. Recently, we have begun to include a small

switch with each rack. Then the rack switch is tied in using a Gigabit Ethernet and a Fibre Cable. This further simplifies the rack to rack Ethernet cabling. Clearly, cable management is a large concern, and some thought should go into how you will manage the mass of cables in your particular environment.

Also evident in Figure E-2 is a crash cart. In this environment, a crash cart is not really necessary. However, since there is only one KVM on the entire cluster, having the crash cart allows a second person to be troubleshooting any hardware problems while someone is still building the cluster.



Figure E-2 Fibre cables

In Figure E-3 on page 201 we have swung around to the back of the front rack (the back of the rack is shown in Figure E-1 on page 199). This is an early picture. If you look closely on the right side, you can see some black cables. It looks like two cables, a gap of six machines or so and then two more cables. These are the partially populated Myrinet cables, which we'll show more closely in a moment.

The Ethernet cables are on the left side of the cabinet (they are red in this case). These lead down to a patch panel at the bottom of the rack. As mentioned before, this allowed all of the internal cables to be installed at a staging area and greatly simplified the on-site Ethernet cabling.



Figure E-3 Ethernet cables

Figure E-4 on page 202 is a closer view near the top. It is also at a later time, since the Myrinet cables are all connected. You can see them trailing off and exiting to the left. Just to the left of the Myrinet cables is a black cable daisy chain from one machine to the next. This is the KVM network.

Near the top is a black object, sort of just hanging there. It is the dongle used to connect the ASMA card to the Service Processor Network. There will generally be two RJ45 connectors and two cables on these units. You will be plugging an Ethernet connector into one of the RJ45 jacks. If this does not work, you should switch to the other connector, since they are not the same.

Just to the left of that, and only on the top machine, is the power connector for the ASMA card.

Immediately to the left of the ASMA card and using black cables is the Service Processor Network (SPN). Each x330 has a cluster of four RJ45 jacks. The right two are the SPN in and out jacks. The left two are for the Ethernet. If you are viewing this book online, you can see the red Ethernet cable leaving just to the left of the black SPN cables. The SPNs form a daisy chain. The Ethernet cables run down to the patch panel.

Finally, on the far left, is the power cord.

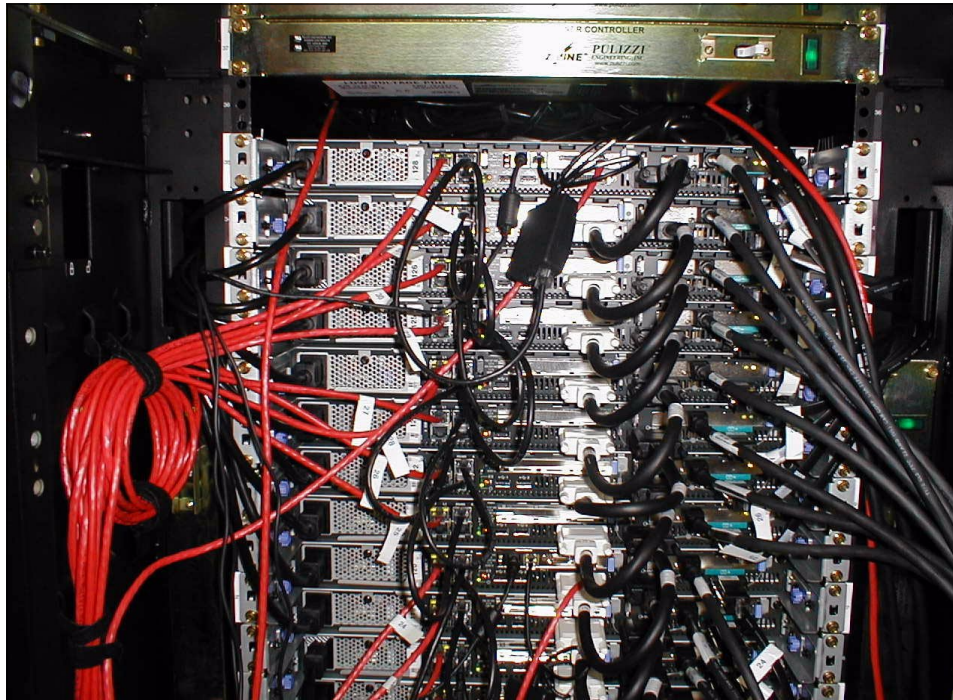


Figure E-4 ASMA and SPN cables

Figure E-5 shows the cabling to only two of the Myrinet switches. In a 256 node cluster, you need six 128 port Myrinet switches to maintain full bisectional bandwidth. In this case, we had four switches in one rack. This led to an incredible amount of cabling in this rack. In future racks, as the clusters grow bigger, we will move to an environment similar to the Ethernet switches. Instead of using a centralized set of switches, we will place a 64 node switch in each rack. This will dramatically reduce the number of rack-to-rack cables needed.

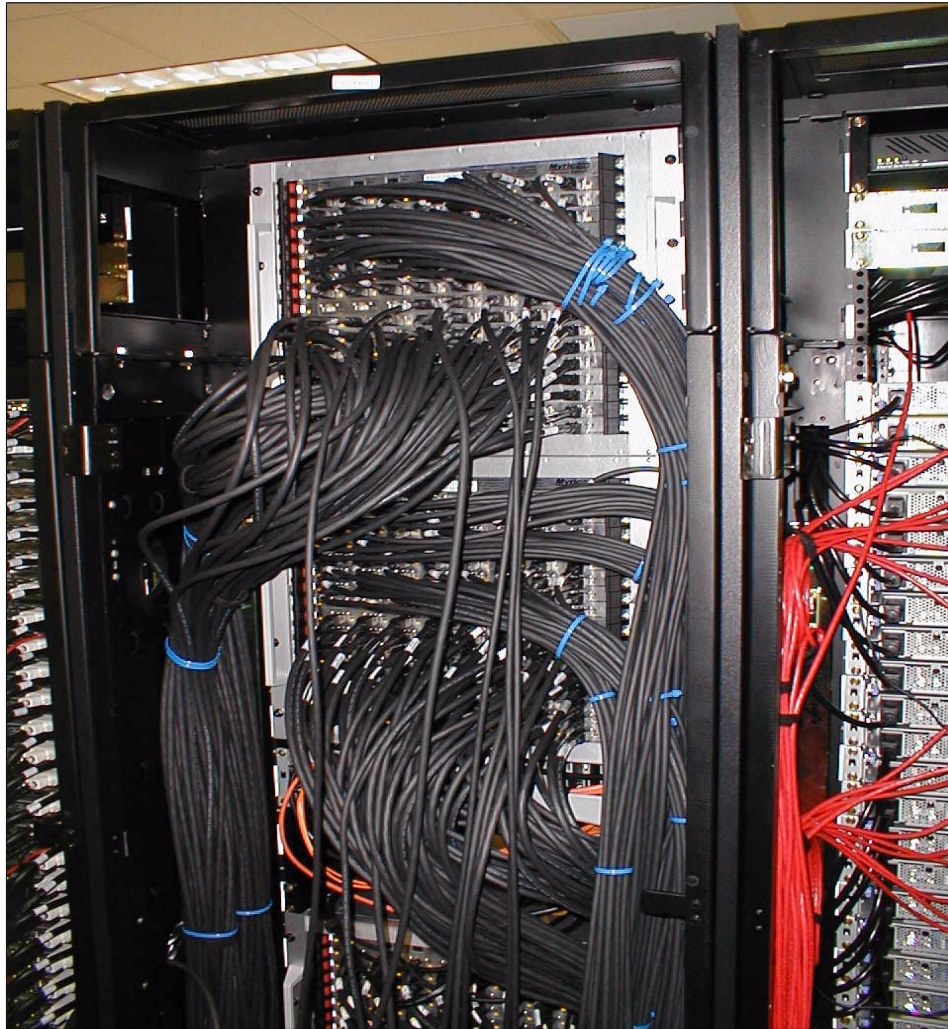


Figure E-5 Myrinet cables

Once the cables are run and neatly dressed, the mechanical installation is complete.

This section was more about our practical observations encountered while building a cluster than about the technical issues. We hope you found this useful.



Cluster questionnaire

This appendix provides the blank cluster questionnaires worksheets to be used during the process of planning, architecture, and deployment:

- ▶ Expectation questionnaire
- ▶ Environmental questionnaire
- ▶ Hardware/configuration questionnaire
- ▶ Software questionnaire

Establishing expectations questionnaire

Table F-1 contains questions regarding your expectations of the cluster.

Table F-1 *Expectations questionnaire*

| Questions | Your comments |
|--|---------------|
| Has the cluster's performance requirements been documented? If so, please specify any required measurements. | |
| Has the cluster configuration been sized and verified to match the requirements? <ul style="list-style-type: none">▶ Application type▶ Processor(s)▶ Memory▶ Cache▶ Disk Storage / RAID (local)▶ Tape (backup)▶ Adapters | |
| Will this cluster require high speed, low latency IPC (Myrinet)? | |
| What will the cluster be used for? | |

| Questions | Your comments |
|---|---------------|
| <p>What are your expectations for this cluster, in 10 words or less?</p> | |
| <p>What cool name did you assign your cluster?</p> | |
| <p>Are you aware of any conditions or issues which could impair your ability to implement this cluster?</p> | |
| <p>Have disk storage requirements and disk layout been established? In particular, this should apply to the cluster's need for connection(s), and the method of connection(s) to external data/storage.</p> | |

Environmental questionnaire

Table F-2 contains questions regarding the environment of your cluster.

Table F-2 *Environment questionnaire*

| Questions | Your comments |
|---|---------------|
| Will the cluster be housed in a computer room and is there space? | |
| Will there be a raised floor? If so, will it provide sufficient room for cabling? | |
| Will the floor support the weight of the cluster? | |
| Is there, or will there be, sufficient power of the proper voltage and with the specified outlets? | |
| Is their adequate cooling for the additional equipment? | |
| Is there an established facilities schedule for making changes or adding/powering on equipment? If so, how will it effect implementation? | |

Hardware/configuration questionnaire

Table F-3 contains questions regarding the hardware/configuration of your cluster.

Table F-3 Hardware/ configuration questionnaire

| Questions | Your comments |
|--|---------------|
| What hardware components, besides Netfinity/eServer xSeries, are included in the solution (RS/6000/pSeries, AS/400/iSeries, S390/zSeries, Storage Subsystems, and so on)? | |
| What are the IP address ranges (for compute nodes, head nodes, Ethernet switches, ASMA adapters, terminal servers, and so on)? Determine if a private address space will be used or if it will be a specified range provided by the Network group. | |
| What are your naming conventions (node, rack, management, and so on)? Do you start with zero or one? Are there any prefix or padding requirements? Note: Naming MUST be prefix/suffix, where prefix is [A-Z a-z \-] and suffix is [0-9]. | |
| Do you have any specific labeling requirements? This is similar to the question on naming conventions but refers to specific labels used to identify equipment. | |

| Questions | Your comments |
|---|---------------|
| <p>Will you be using VLANs? If so, how many? How will they be partitioned?</p> | |
| <p>What security/authentication method do you require (global NIS, local NIS, local users only, rcp /etc/password, and so on)? Note: We do not recommend using NIS on the compute nodes.</p> | |
| <p>How is external storage to be connected?</p> | |
| <p>What throughput is required from the cluster to your enterprise?</p> | |
| <p>Does your network group have any fixed schedule for making changes or adding equipment to the network and if so, how will it effect implementation?</p> | |

| Questions | Your comments |
|--|----------------------|
| Does your network group have any specific equipment requirements for connectivity (brand, specific equipment, copper, fiber, and so on)? | |
| If you have special equipment/peripheral requirements, are production level Linux drivers and software available? | |

Software questionnaire

Table F-4 contains questions regarding the software for your cluster.

Table F-4 *Software questionnaire*

| Questions | Your comments |
|--|---------------|
| What other open source software will be installed? | |
| What commercial software will be installed? | |
| Do you have any existing Linux cluster(s) running the aforementioned applications? | |
| What do you use as a resource manager? It is assumed you will be using PBS, but are there others (Maui, LSF, and so on)? | |
| Do you require special/commercial compilers (the IBM Linux Cluster provides the PGI compilers)? | |
| Does the operating system support all planned features and are any additional tools or libraries supported by the recommended (Red Hat 6.2) level of the operating system? | |
| Are there any version dependencies for any additional software components? | |

| Questions | Your comments |
|---|----------------------|
| Is there a backup/recovery plan in place? Describe and define the backup requirements, frequency and quantity. | |



Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246041/>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246041.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

| <i>File name</i> | <i>Description</i> |
|------------------------------|------------------------------------|
| xCAT-distribution.tgz | Zipped file with xCAT Distribution |

System requirements for downloading the Web material

The following system configuration is recommended:

| | |
|--------------------------|--|
| Hard disk space: | 3 MB minimum |
| Operating System: | An UNIX-based system, such as Linux or AIX |
| Processor: | Any |
| Memory: | 16 MB |

How to use the Web material

Create a directory on your system (/tmp for example), download the file to that directory and then unzip the xCAT distribution file. You should also download the Kickstart configuration file to use as an example.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 220.

- ▶ *GPFS: A Parallel File System*, SG24-5165
- ▶ *Netfinity Server Disk Subsystem*, SG24-2098
- ▶ *PSSP 2.4 Technical Presentation*, SG24-5173
- ▶ *PSSP 3.1 Announcement*, SG24-5332
- ▶ *PSSP 3.2: RS/6000 SP Software Enhancements*, SG24-5673
- ▶ *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5853
- ▶ *Sizing and Tuning GPFS*, SG24-5610

Other resources

These publications are also relevant as further information sources:

- ▶ Sterling, Thomas L., et al., *How to Build a Beowulf - A Guide to the Implementation and Application of PC Clusters*, MIT Press, 1999, ISBN 026269218X
- ▶ Raymond, Eric, *The Cathedral and the Bazaar*, O'Reilly and Associates, Inc., 1999, ISBN 1565927249
- ▶ Albitz, Paul, et al., *DNS and Bind*, O'Reilly and Associates, Inc., 2001, ISBN 0596001584

These publications can be found on the Web:

- ▶ *The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters*, found at:
www.scl.ameslab.gov/Publications/HalsteadPubs/usenix_2k.pdf
- ▶ *IBM Netfinity Advanced Systems Management*, found at:

<http://www.pc.ibm.com/us/eserver/xseries/>

- ▶ *The Failure of TCP in High-Performance Computing Grid*, found at:
www.sc2000.org/techpapr/papers/pap174.pdf

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ <http://www.platform.com/> - LSF (Load Sharing Facility) workload management software information Web site
- ▶ <http://www.pbspro.com/> - PBS (Portable Batch System) Web site
- ▶ <http://www.pgroup.com/> - The Portland Group Web site
- ▶ <http://www.mpi-forum.org/> - MPI Forum Web site
- ▶ <http://www-unix.mcs.anl.gov/mpi/mpich> - MPICH-A Portable Implementation of MPI Web site
- ▶ http://www.netlib.org/scalapack/scalapack_home.html - ScaLAPACK Web site
- ▶ <http://www.netlib.org/atlas> - Automatically Tuned Linear Algebra Software (ATLAS) Web site
- ▶ <http://www.havelland.com/index.htm?povbench/index.htm> - PoVBench Web site
- ▶ <http://www.tabsnet.com> - POV-Ray Benchmark Web site
- ▶ <http://www.netlib.org/benchmark/> - Benchmarks in general Web site
- ▶ <http://www.top500.org/> - High-Performance Computing Information Web site
- ▶ <http://www.etnus.com/Products/TotalView/index.html> - TotalView debugger Web site
- ▶ <http://www.myri.com/> - Myrinet and Myricom, Inc. Web site
- ▶ <http://www.povray.org/> - POV-Ray Web site
- ▶ <http://www.openpbs.org/> - OpenPBS Web site
- ▶ <http://www.can.ibm.com/wwconfig/> - IBM PC Support Configurator Web site
- ▶ <http://www.pc.ibm.com/support/> - IBM PC Support Web site
- ▶ <http://csep1.phy.ornl.gov/ca/node11.html> - Flynn's Taxonomy
- ▶ <ftp://ftp.kernel.org> - Linux Kernel Archive Web site
- ▶ <http://www.transarc.ibm.com/> - Andrew File System Web site

- ▶ <http://www.stacken.kth.se/projekt/ar1a> - ARLA Web site
- ▶ <http://www.globalfilesystem.org/> - GFS Web site
- ▶ <http://www.gnu.org/gnu/thegnuproject.html> - The Gnu Project Web site
- ▶ <http://www.linuxdoc.org/HOWTO/Beowulf-HOWTO-4.html> - Beowulf cluster system design how-to document Web site
- ▶ <http://www.linuxdoc.org/HOWTO/KickStart-HOWTO.html> - Red Hat Linux KickStart HOWTO Web site
- ▶ <http://www.pc.ibm.com/us/compat> - IBM Server Proven Web site
- ▶ <ftp://www.redbooks.ibm.com/redbooks/SG246041/> - Additional materials for this Linux HPC Cluster Installation Redbook Web site
- ▶ <http://www.redhat.com/support/errata/> - Red Hat Linux Errata Web site
- ▶ <http://www.icase.edu/coral/LinuxTCP2.html> - Linux 2.2.12 TCP Performance Fix for Short Messages Web site
- ▶ <ftp://ftp.cistron.nl/pub/people/miquels/kernel/v2.2> - Cistron FTP site for kernel 2.2
- ▶ <http://www.csd.uu.se/~mikpe/linux/perfctr> - perfctr patch Web site
- ▶ <http://support.intel.com/support/network> - Intel Networking Support Web site
- ▶ <ftp://ftp.ora.com/pub/examples/nutshell/dnsbind/> - *DNS and Bind* examples FTP site
- ▶ <http://developer.intel.com> - Intel's Developer Web site
- ▶ <ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/> - Linux Kernel Archives Web site
- ▶ <ftp://ftp.mamalinux.com/pub/atftp> - atftp FTP site
- ▶ <http://www.conserver.com> - Conserver Web site
- ▶ <http://www.fping.com> - fping Web site
- ▶ <http://www.openssh.com> - OpenSSH Web site
- ▶ <http://www.gnu.org/software/gcc/gcc.html> - gcc home page
- ▶ <http://www.gnu.org/software/gcc/install/index.html> - Installing GCC Web site
- ▶ <http://rpmfind.net/linux/RPM> - RPM repository on rpmfind.net Web site
- ▶ <http://www.supercluster.org> - Supercluster.org Web site

How to get IBM Redbooks

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web Site

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web Site for information about all the CD-ROMs offered, updates and formats.

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of a specific Statement of General Direction.

The following terms are trademarks of other companies:

C-bus is a registered trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Linux is a registered trademark in the United States and other countries of Linus Torvalds.

Red Hat, RPM, and all Red Hat-base trademarks and logos are trademarks or registered trademarks of Red Hat Software in the United States and other countries.

SuSe, SuSe Linux and all SuSe-base trademarks and logos are trademarks or registered trademarks of S.u.S.E. Linux in the United States and other countries.

OpenLinux and all OpenLinux-base trademarks and logos are trademarks or registered trademarks of Caldera in the United States and other countries.

Corel Linux and all Corel Linux-base trademarks and logos are trademarks or registered trademarks of Corel in the United States and other countries.

Linux Mandrake and all Linux Mandrake-base trademarks and logos are trademarks or registered trademarks of Mandrake in the United States and other countries.

TurboLinux and all TurboLinux-base trademarks and logos are trademarks or registered trademarks of TurboLinux in the United States and other countries.

Debian GNU/Linux and all Debian GNU/Linux-base trademarks and logos are

trademarks or registered trademarks of Debian in the United States and other countries.

GNU Project, GNU, GPL and all GNU-base trademarks and logos are trademarks or registered trademarks of Free Software Foundation in the United States and other countries.

Intel, IA-32, IA-64, Itanium, Pentium, and all Intel-base trademarks and logos are trademarks or registered trademarks of Intel Corporation in the United States and other countries.

ActionMedia, LANDesk, MMX, and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

NFS and Network File System are trademarks of Sun Microsystems, Inc.

Open Software Foundation, OSF, OSF/1, OSF/Motif, and Motif are trademarks of Open Software Foundation, Inc.

POSIX is a trademark of the Institute of Electrical and Electronic Engineers (IEEE).

Myrinet is a trademark of Myricom, Inc.

Equinox is a trademark of Equinox Systems, Inc.

Netware is a trademark of Novell, Inc.

ASM is trademark of Microtec Research, Inc.

AFS is marketed, maintained, and extended by Transarc Corporation.

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

Index

A

- AFS (Andrew File System) 46
- ALICE (Advanced Linux Installation and Configuration Environment) 45
- API (Application Program Interface)
 - architecture 16
 - messages and threads 16
- applications 13
 - batch and parallel jobs 16
 - concurrency and parallelism 17
- architecture
 - distributed clusters 15
 - MPP (Massively Parallel Processor) 15
 - parallel computing 14
 - SMP (Shared Memory Processors) 15
- ASM (Advanced System Management)
 - ASMA card 40
 - ASMA setup 76
 - processor 40
 - Wiseman card 40
- asma.tab 179

B

- Beowulf 2
 - applications 13
 - architecture 9, 11
 - components 12
 - hardware architecture 19
 - logical view 10
 - parallelism 13
- Beowulf clusters
 - (see Beowulf)
- bottlenecks 17

C

- cluster node functions
 - compute 24
 - control 22
 - installation 23
 - management 22
 - storage 23
 - user 21

clusters

- application server 7
- Beowulf clusters
 - (see Beowulf)
- database server 7
- HA (High Availability) 5
- load balancing 6
- server consolidation 7
- types of 4
- web clusters 5
- web farms 5
- web server 7

compilers 50

compute nodes 39, 74

configuration

- lab example 69, 193
- naming conventions 63
- questionnaire 66
- real example 197
- schemes 63

configurator 60

Conserver 104

CSM (Cluster Systems Management) 49

D

- development environment
 - compilers 50
 - debugger 52
 - libraries 51
 - parallel applications 53

E

- Equinox 29, 39, 79, 101

F

- Fast Ethernet 41
- file systems
 - AFS (Andrew File System) 46
 - GFS (Global File System) 46
 - GPFS (General Parallel File System) 47
 - local file system 45
 - NFS (Network File System) 45, 115

- parallel file systems 31
- PVFS (Parallel Virtual File System) 47

G

- GFS (Global File System) 46
- Gigabit Ethernet 41
- GM 43
- GNU
 - GNU project 3
 - GPL (General Public License) 3
- GPFS (General Parallel File System) 47

H

- hardware architecture
 - compute nodes 39
 - head node 38
 - management
 - KVM (keyboard, video, mouse) 41
 - SPN (Service Processor Network) 40
 - terminal server
 - (see Equinox)
- hardware setup 73
 - ASMA 76
 - BIOS 76
 - C2T 78
 - cabling 77, 82
 - compute nodes 74
 - Equinox 79
 - FastEthernet 74
 - flash updates 76
 - Gigabit Ethernet 74, 80
 - head node 74, 82
 - KVM 77
 - Myrinet 75, 79
 - PDU 77
 - rack 77
 - serial port 74
 - ServeRAID 74
 - SPN 74
 - terminal servers 79
 - UPS 77
- head node 38, 74
- HPC (High-Performance Computing)
 - Beowulf clusters
 - (see Beowulf)
 - definition 7

I

- IBM Director 50
- installation
 - compute node
 - ASM setup 119
 - collect MAC addresses 117
 - description 109
 - DHCP (Dynamic Host Configuration Protocol) 116
 - DNS (Domain Name System) 114
 - install first node 119
 - install remaining nodes 120
 - NFS (Network File System) 115
 - NIS (Network Information System) 115
 - populate xCAT tables 110
 - post install 120
 - services configuration 114
 - xntpd 114
 - GCC compiler 124
 - libraries and debugger 136
 - management node
 - add login user 89
 - additional software 93
 - atftp 100
 - before you start 86
 - Conserver 104
 - description 85
 - Equinox 101
 - ethernet drivers 97
 - fping 105
 - h2n 99
 - install Red Hat updates 90
 - kernel 93, 96
 - kernel patches 94
 - Kickstart configuration file 87
 - OpenSSH 106
 - PXELinux 99
 - Red Hat 86
 - set root password 88
 - syslogd 91
 - updating host table 90
 - verify network configuration 89
 - xCAT 92
 - MPICH 127
 - PGI compiler 124
 - POVRay 186
- IPC (Inter Process Communication)
 - definition 32
 - technologies 26

J

job management 54
 LSF (Load Sharing Facility) 55
 Maui 54
 PBS (Portable Batch System) 54
job scheduler 35

K

Kickstart 44
KVM (keyboard, video, mouse) 41, 77

L

libraries 51
LSF (Load Sharing Facility) 55

M

mac.tab 180
Maui 54
MPI (Message Passing Interface) 33
Myrinet 43

N

network architecture
 components 27
 Fast Ethernet 41
 Gigabit Ethernet 41
 GM 43
 how to design 27
 Myrinet 43
 switch 26
 technologies 26, 41
 VIA (Virtual Interface Architecture) 42
 VLAN (Virtual Local Area Networks) 28
NFS (Network File System) 45, 115
nodehm.tab 175
nodelist.tab 171
noderes.tab 172
nodetype.tab 174

O

open source 3
OpenMP 34
OpenSSH 106

P

parallel applications 53

parallel computing 13
parallel file system 31
passwd.tab 181
PBS (Portable Batch System) 54
planning cluster installation 58
 cabling 61
 compatibility 61
 environment 58
 hardware 58
 software 59
POVRay 185
pping 160
psh 158
PVFS (Parallel Virtual File System) 47
PVM (Parallel Virtual Machine) 33
PXELinux 44, 100

Q

questionnaire 66
 environmental questions 208
 establishing expectations 206
 hardware/configuration questions 209
 software questions 212

R

rcad 142
rcons 144
Red Hat 86
Redbooks Web Site 220
 Contact us xx
remote access
 (see Equinox)
resource management
 job management 34
 job scheduler/policy manager 35
 LSF (Load Sharing Facility) 55
 PBS (Portable Batch System) 54
 resource manager 34
reventlog 153
rinstall 161
rinv 155
rreset 141
rvid 147
rvitals 151

S

ServeRAID 74, 86

- site.tab 169
- software architecture
 - compilers 50
 - components 30
 - debugger 52
 - development 50
 - file systems 31
 - IPC (Inter Process Communication) 32
 - libraries 51
 - MPI (Message Passing Interface) 33
 - OpenMP 34
 - operating system 44
 - parallel applications 53
 - PVM (Parallel Virtual Machine) 33
- software installation
 - Kickstart 44
 - operating system 44
 - PXELinux 44
- SPN (Service Processor Network) 28, 40, 74
- system management 48
 - CSM 49
 - IBM Director 50
 - xCAT 48

- rcons 144
- reventlog 153
- install 161
- rinv 155
- rreset 141
- rvid 147
- rvitals 151
- wcons 145
- winstall 163
- wvid 149
- directory structure 137
- explanation 48
- installation 92
- tables
 - apc.tab 183
 - asma.tab 179
 - mac.tab 180
 - nodehm.tab 175
 - odelist.tab 171
 - noderes.tab 172
 - nodetype.tab 174
 - passwd.tab 181
 - populate tables 110
 - site.tab 169

T

- terminal server
 - (see Equinox)

V

- VIA (Virtual Interface Architecture) 42
- VLAN (Virtual Local Area Networks)
 - management 28
 - public 28
 - SPN (Service Processor Network) 28

W

- wcons 145
- winstall 163
- Wiseman card 40
- wvid 149

X

- xCAT (xSeries Cluster Administration Tool)
 - commands
 - pping 160
 - psh 158
 - rcad 142



Linux HPC Cluster Installation



Cluster installation using xCAT - xCluster Administration Tools

This redbook will guide system architects and systems engineers toward a basic understanding of cluster technology, terminology, and the installation of a Linux High-Performance Computing (HPC) cluster (a Beowulf type of cluster) into an IBM @server xSeries cluster.

Linux clustering based on IBM @server xSeries

This document focus on xCAT (xCluster Administration Tools) for installation and administration. All nodes and components of the cluster, such as compute nodes and management nodes, are installed with xCAT. This tool is a collection of scripts, tables, and commands used to build and administer a Beowulf type of cluster or a farm of replicated nodes. All these xCAT components are explained in the appendixes of the Redbook. Detailed procedures on how to install and properly configure a Linux Red Hat operating system in the nodes of an IBM @server xSeries HPC cluster are presented.

Installing Red Hat with Kickstart and xCAT

For architectural design, we present a generic cluster architecture, specifics for an HPC type of cluster, and the physical and logical components of an HPC cluster. A solution design guideline is presented as well, giving the key concepts and design principals to be used during the architectural and planning phases.

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