# Deploying Linux on IBM @server pSeries Clusters

System installation and administration

Application case studies

Cluster management

Dino Quintero
Prabhakar Attaluri
Tomas Baublys
Xinghong He
Chin Yau Lee
Francois Thomas

Redbooks

ibm.com/redbooks

International Technical Support Organization

# Deploying Linux on IBM eserver pSeries Clusters

December 2003

SG24-7014-00

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

**First Edition (December 2003)**

This edition applies to Cluster Systems Management (CSM) Version 1, Release 3, Modification 2, General Parallel File System (GPFS) Version 2, Release 2, ESSL 4.1, Parallel ESSL 3.1, SLES 8, and RHES 3.

# Contents

**Notices** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix
Trademarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x

**Preface** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xi
The team that wrote this redbook. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xi
Become a published author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xiv
Comments welcome. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xiv

**Chapter 1. Introduction**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2 Why use Linux on pSeries. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2.1 Reasons to deploy Linux on pSeries clusters . . . . . . . . . . . . . . . . . . . . 2
1.3 POWER4 architecture advantages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
1.4 Chapter description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**Chapter 2. System installation**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.1 Before you install. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.1.1 Check the prerequisites . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.1.2 Prepare the system. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.1.3 Monitoring the installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2.1.4 Review your choices . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
2.2 Step-by-step installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
2.2.1 SLES 8 installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
2.2.2 RHAS 3 installation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
2.2.3 Installation tips . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
2.3 From network to fully unattended installation . . . . . . . . . . . . . . . . . . . . . . 49
2.3.1 Network installation fundamentals . . . . . . . . . . . . . . . . . . . . . . . . . . 50
2.3.2 A practical example. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
2.3.3 Unattended installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
2.4 Where is the BIOS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
2.4.1 PowerPC. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
2.4.2 CHRP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
2.4.3 Open Firmware basics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
2.4.4 Yaboot. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
2.5 Post-installation tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
2.5.1 Checking the installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
2.5.2 Applying SLES 8 patches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
2.5.3 Where to get help . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93

**Chapter 3. System administration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
3.1 The bash shell . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
3.2 To YaST or not to YaSt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
3.3 Web-based system administration (Webmin) . . . . . . . . . . . . . . . . . . 100
3.3.1 Webmin installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
3.3.2 Using Webmin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
3.4 Runlevels . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
3.5 Local user management . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
3.5.1 User distribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
3.5.2 Using sudo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
3.6 Logical Volume Manager (LVM) . . . . . . . . . . . . . . . . . . . . . . . . . . 113
3.6.1 What is Logical Volume Manager . . . . . . . . . . . . . . . . . . . . . . . 113
3.6.2 How LVM works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
3.6.3 LVM terminology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
3.6.4 Setting up LVM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
3.6.5 Using the command line to create LVM . . . . . . . . . . . . . . . . . . . 115
3.6.6 Using YaST for LVM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116
3.6.7 LVM features . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
3.7 Software mirroring and LVM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
3.7.1 Using YaST to create RAID . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
3.7.2 Using the command line to create RAID . . . . . . . . . . . . . . . . . . . 127
3.8 File systems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
3.8.1 Second extended file system (ext2) . . . . . . . . . . . . . . . . . . . . . . 132
3.8.2 Third extended file system (ext3) . . . . . . . . . . . . . . . . . . . . . . . . 132
3.8.3 Reiser file system (reiserfs) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
3.8.4 Journaling File system (jfs) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
3.9 RPM management . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
3.9.1 Naming convention . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
3.9.2 Querying RPM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
3.9.3 Installing and uninstalling RPMs . . . . . . . . . . . . . . . . . . . . . . . . 138
3.10 System updates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
3.10.1 Setting up our own SuSE update server . . . . . . . . . . . . . . . . . . 139
3.10.2 Online update on Red Hat AS . . . . . . . . . . . . . . . . . . . . . . . . . 141
3.11 System backup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
3.12 ssh . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
3.12.1 Exchanging ssh keys . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
3.13 DHCP server configuration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
3.14 DNS server configuration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
3.15 Using iptables for security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
3.16 OpenLDAP implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
3.16.1 Setting up an LDAP server . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
3.16.2 LDAP client configuration . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
3.16.3 Defining users in the LDAP directory . . . . . . . . . . . . . . . . . . . . 156

3.16.4 Managing users. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
3.16.5 Using SSL/TLS with OpenLDAP . . . . . . . . . . . . . . . . . . . . . . . . . 160

**Chapter 4. Linux for pSeries RAS and problem determination** . . . . . . . . 163
4.1 Linux on pSeries RAS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
4.1.1 RunTime abstraction service in PPC64 . . . . . . . . . . . . . . . . . . . . 165
4.1.2 IBM diagnostics tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
4.2 System logs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174
4.2.1 Linux syslog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174
4.3 Event logging - enterprise event log . . . . . . . . . . . . . . . . . . . . . . . . . . 179
4.3.1 evlog in a centralized environment . . . . . . . . . . . . . . . . . . . . . . . . 180
4.4 Log rotation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
4.5 Linux rescue methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
4.5.1 Boot loader corruption. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 184
4.5.2 File system corruption . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
4.5.3 RHAS 3 rescue mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 189
4.5.4 Using /proc file systems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 189
4.6 Performance monitoring . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
4.6.1 Performance monitoring with ganglia . . . . . . . . . . . . . . . . . . . . . . 193
4.6.2 Basic tuning tips for Linux . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207

**Chapter 5. Cluster Systems Management (CSM)** . . . . . . . . . . . . . . . . . . 211
5.1 CSM concepts and architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
5.1.1 Clustering basics. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
5.1.2 Types of clusters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
5.1.3 RSCT, RMC, and RM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 218
5.1.4 CSM hardware control . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
5.1.5 Cluster File Manager (CFM) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
5.1.6 CSM software maintenance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
5.1.7 CSM diagnostic probes. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
5.1.8 CSM security. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
5.1.9 Distributed shell (dsh) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 221
5.2 CSM planning, installation and configuration . . . . . . . . . . . . . . . . . . . 222
5.2.1 Planning hardware . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 223
5.2.2 Planning software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 224
5.2.3 Planning network. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 226
5.2.4 Planning security. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 227
5.2.5 Installing CSM on the management server. . . . . . . . . . . . . . . . . . 228
5.2.6 Installing the CSM license. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 233
5.2.7 Install verification. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 234
5.2.8 CSM management server configuration . . . . . . . . . . . . . . . . . . . . 235
5.2.9 Defining/adding nodes to the CSM cluster . . . . . . . . . . . . . . . . . . 237
5.2.10 Preparing for node installation . . . . . . . . . . . . . . . . . . . . . . . . . . 240

5.2.11 Installing managed nodes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 245
5.2.12 Node install verification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 250
5.3 CSM administration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251
5.3.1 Log file management. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251
5.3.2 Node groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253
5.3.3 Hardware control. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 254
5.3.4 Configuration File Manager (CFM) . . . . . . . . . . . . . . . . . . . . . . . . . 256
5.3.5 Software maintenance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 259
5.3.6 CSM Monitoring . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 262
5.3.7 Diagnostic probes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 265
5.3.8 Querying the CSM database. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 267
5.3.9 Un-installing CSM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 268
5.3.10 Distributed Command Execution Manager (DCEM) . . . . . . . . . . . 269
5.3.11 Backing up CSM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 269
5.3.12 CSM problem determination and diagnostics . . . . . . . . . . . . . . . . 269
5.4 CSM hostname changes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 272
5.5 CSM interoperability . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 272
5.5.1 CSM for pSeries AIX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 273
5.5.2 CSM for pSeries Linux . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 273
5.5.3 CSM for xSeries Linux and e325 (Cluster 1350). . . . . . . . . . . . . . . 274
5.5.4 Parallel Systems Support Program (PSSP)-3.5 . . . . . . . . . . . . . . . 274
5.5.5 Matrix of CSM products. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 275
5.6 Future of CSM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 277

**Chapter 6. Getting started with GPFS** . . . . . . . . . . . . . . . . . . . . . . . . . . 279
6.1 GPFS description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 280
6.1.1 Principles of operation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 280
6.1.2 Terminology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 282
6.1.3 Components . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 283
6.1.4 Advantages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 284
6.2 RSCT peer domain setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 285
6.2.1 RSCT requisites . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 286
6.2.2 Overview of peer domain operations . . . . . . . . . . . . . . . . . . . . . . . . 286
6.2.3 Creating a peer domain. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 286
6.2.4 Starting and stopping a peer domain . . . . . . . . . . . . . . . . . . . . . . . . 287
6.2.5 Removing a peer domain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 289
6.3 GPFS installation and basic configuration. . . . . . . . . . . . . . . . . . . . . . . . 290
6.3.1 Planning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 290
6.3.2 Software installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 291
6.3.3 Compiling the portability layer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 291
6.3.4 Configuring and bringing up GPFS . . . . . . . . . . . . . . . . . . . . . . . . . 293
6.3.5 Shutting down and restarting GPFS . . . . . . . . . . . . . . . . . . . . . . . . 301

**Chapter 7. High Performance Computing case studies** . . . . . . . . . . . . . 303
7.1 Hardware and software overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 305
7.2 Myrinet. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 305
7.2.1 Myrinet hardware . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 306
7.2.2 GM installation and enablement . . . . . . . . . . . . . . . . . . . . . . . . . . 307
7.2.3 IP over Myrinet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 310
7.2.4 Checking GM status and performance . . . . . . . . . . . . . . . . . . . . . 311
7.2.5 MPICH_GM installation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 311
7.3 Gigabit Ethernet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 312
7.4 Compilers and ESSL libraries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 313
7.4.1 Installing GNU and cross-compilers . . . . . . . . . . . . . . . . . . . . . . . 314
7.4.2 Installing XL Fortran and VisualAge C++ compilers . . . . . . . . . . . 315
7.4.3 Installing Absoft Pro Fortran 8.2 compiler. . . . . . . . . . . . . . . . . . . 316
7.4.4 Installing ESSL 4.1 and PESSL 3.1 . . . . . . . . . . . . . . . . . . . . . . . 316
7.5 More packages ported and installed . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
7.6 Benchmarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
7.6.1 System configurations. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
7.6.2 Linpack DP n=100 Benchmark . . . . . . . . . . . . . . . . . . . . . . . . . . . 318
7.6.3 Linpack TPP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 320
7.6.4 Linpack HPC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 322
7.6.5 NetPIPE - point-to-point communication. . . . . . . . . . . . . . . . . . . . 331
7.6.6 Pallas PMB - more on MPI . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 338
7.6.7 NAS Parallel benchmarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 347

**Chapter 8. Commercial application case studies**. . . . . . . . . . . . . . . . . . 361
8.1 High availability with heartbeat and DB2. . . . . . . . . . . . . . . . . . . . . . . . 362
8.1.1 Hardware and software components . . . . . . . . . . . . . . . . . . . . . . 362
8.1.2 Preparing the nodes for cluster ready. . . . . . . . . . . . . . . . . . . . . . 363
8.1.3 Application installation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 367
8.1.4 Heartbeat clustering software installation . . . . . . . . . . . . . . . . . . . 373
8.1.5 Testing the cluster. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 376
8.1.6 Extending the cluster with Apache and PHP for front-end . . . . . . . 380
8.1.7 Using LVS to load balance Web servers. . . . . . . . . . . . . . . . . . . . 384
8.1.8 Final landscape of the setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 391
8.1.9 Alternate solutions to improve the cluster . . . . . . . . . . . . . . . . . . . 392
8.2 High Availability with Red Hat ClusterSuite. . . . . . . . . . . . . . . . . . . . . . . 393
8.2.1 Red Hat ClusterSuite . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 394
8.3 Tivoli Storage Manager (TSM) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 406
8.3.1 TSM server installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 406

**Appendix A. Linux for AIX system administrators**. . . . . . . . . . . . . . . . . 417
Major features . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 418
Common system files. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 419

Task-specific command comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 420

**Appendix B. Feature comparison** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 423

**Appendix C. Additional material** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 427
Locating the Web material . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 427

**Abbreviations and acronyms** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 429

**Related publications** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 431
IBM Redbooks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 431
Other publications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 431
Online resources . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 432
How to get IBM Redbooks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 435
Help from IBM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 435

**Index** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 437

# Notices

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

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

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

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

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

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

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

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

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

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

# Trademarks

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

AIX 5L™
AIX/L®
AIX®
DB2®
Enterprise Storage Server®
@server™
@server ™
Enterprise Storage Server®
IBM®
IntelliStation®
iSeries
Micro Channel®
Notes®
OS/2®
PowerPC Reference Platform®
PowerPC®
POWER3™
POWER4™
pSeries®
Redbooks™
Redbooks (logo) ™
RS/6000®
SP2®
ThinkPad®
Tivoli®
VisualAge®
WebSphere®
xSeries®
zSeries®

The following terms are trademarks of other companies:

Intel and Intel Inside (logos) are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

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

# Preface

For customers interested in Linux who are seeking powerful and reliable servers to support their applications, IBM® eServer pSeries® offers 64-bit POWER systems to match their requirements at lower costs. Unlike other servers supporting solely high performance Linux needs, IBM eServer pSeries provides leadership price/performance with high reliability and a roadmap to provide the customer with investment protection.

This IBM Redbook provides information and guidelines on how to deploy Linux on IBM eServer pSeries clusters. Topics covered include:

- System installation
- System administration
- Linux for pSeries RAS and problem determination
- Cluster Systems Management (CSM)
- Getting started with GPFS
- High Performance Computing case studies
- Commercial application case studies

This redbook also includes hints and tips that illustrate particular observations discovered during the residency. Additional reference materials and Web sites have been included as well, to provide the reader with other sources of information necessary to implement Linux on pSeries clusters.

Note that this document is not a replacement for the IBM product manuals on this topic; instead, it is intended as an additional informational deskside tool to help technical professionals understand, architect, deploy, and manage Linux on pSeries clusters.

## The team that wrote this redbook

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

**Dino Quintero** is a Senior IT Specialist at ITSO in Poughkeepsie, New York. Before joining ITSO, he worked as a Performance Analyst for the Enterprise Systems Group, and as a Disaster Recovery Architect for IBM Global Services. His areas of expertise include disaster recovery and pSeries clustering solutions. He is certified on pSeries system administration and pSeries clustering technologies. He is also an IBM Certified Professional on pSeries technologies.

Currently, he leads technical teams in delivering redbook solutions on pSeries clustering technologies, and delivers technical workshops worldwide.

**Prabhakar Attaluri** is an AIX Team lead and Senior Systems Integration Professional at IBM Global Services, eBusiness Hosting in Schaumburg, IL. He holds a Masters degree in Engineering from Utah State University at Logan, UT. He is an IBM Certified Advanced Technical Expert and Sun Certified Systems Administrator. He has over eight years of experience with the UNIX® operating system and its management, and his expertise includes UNIX administration, high availability clustering, virtual load balancing, and the design and setup of complex enterprise e-business solutions.

**Tomas Baublys** works for IBM pSeries technical sales support in Germany. He has over five years of experience in Linux, Netware and AIX support and consulting. He holds a Masters degree in Philosophy from Friedrich-Wilhelms-University in Bonn. He is a Certified Novell Engineer (CNE) and IBM Certified Specialist in 690 technical support and pseries AIX system administration.

**Xinghong He** is an Advisory Engineer/Scientist at the pSeries and HPC Benchmark Center in Poughkeepsie, New York. He has 10 years of experience in parallel and high performance computing. He holds a PhD degree in Theoretical and Computational Physics from Queen’s University of Belfast. His areas of expertise include application porting and performance tuning for AIX and Linux. He has written extensively on performance comparison and analysis.

**Chin Yau Lee** works as an Advisory IT Specialist in the pSeries Field Technical Sales Team at the IBM Systems Group of IBM Singapore. He holds a Honours Degree in Computing and Information System from the University of Staffordshire. He has been using Linux since 1997, and is an IBM Certified Advance Technical Expert, Sun Certified Network Administrator, and Red Hat Certified Engineer. His areas of expertise include Linux/UNIX system administration, high availability, and Internet-based solutions, in which he has done extensive work for the last four years.

**Francois Thomas** works as an HPC Pre-sales Specialist in the EMEA Deep Computing organization in France. He has 14 years of experience in the field of scientific and technical computing. He holds a PhD in Physics from ENSAM/Paris VI University. His areas of expertise include application code tuning and parallelization, as well as AIX/Linux clusters management. He has written extensively on the use of distributed memory methods for solving non-linear structural analysis problems.

*Team members (left to right): Francois Thomas, Xinghong He, Prabhakar Attaluri, Dino Quintero (project leader), Tomas Baublys, Chin Yau Lee*

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

Octavian Lascu
International Technical Support Organization, Austin Center

Keshav Ranganathan, Satish Dodda, Scott Sakolish, Chris Derobertis, Joan McComb, Janet Elsworth, Greg Rodgers, Joyce Mak, Ananthanaraya Sugavanam (Suga)
IBM Poughkeepsie

James Holdren
IBM Raleigh

Michael Perzl, Thilo Mende
IBM Germany

David Engebretsen, David C. Boutcher
IBM Rochester

Dave Randall, Michael T. Strosaker, Jeffery J. Scheel, Yuri L. Volobuev, Bala Ekambaram, Ken King, Chuck Bryan
IBM Austin

Goetz Rieger, Stephan Duehr, Olaf Hering
SuSE Germany

Daniel Riek
Red Hat Germany

Konstantin Starodubtsev
Saint-Petersburg State University

Thanks also to the Open Source software developers and community who made GNU/Linux possible.

## Become a published author

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

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

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

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us!

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

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

  **ibm.com**/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 003 Internal Zip 2834
  11400 Burnet Road
  Austin, Texas 78758-3493

# 1

# Introduction

In this chapter, we introduce and discuss the major reasons to deploy Linux on pSeries clusters.

We cover the following topics:

- 1.1, “Introduction” on page 2
- 1.2, “Why use Linux on pSeries” on page 2
- 1.3, “POWER4 architecture advantages” on page 3
- 1.4, “Chapter description” on page 3

## 1.1 Introduction

The pSeries platform addresses the need for reliability by providing high-availability solutions to meet today's requirements for e-business. To meet these requirements, the pSeries products offer a full range of high-performance servers, with a full set of highly functional, state-of-the-art software to match the highest customer requirements of reliability, scalability, manageability, Internet affinity, and security.

## 1.2 Why use Linux on pSeries

For customers interested in Linux who need powerful and reliable servers to support their applications, IBM eServer pSeries offers 64-bit POWER servers to match their requirements at lower costs. Unlike other servers supporting solely high performance Linux needs, IBM eServer pSeries provides leadership price/performance with high reliability and a future roadmap to provide the customer with investment protection.

Excellent quality and reliability are inherent in all aspects of the IBM eServer pSeries design and manufacture, and the fundamental principle of the design approach is to minimize outages. The reliability, availability, and serviceability (RAS) features help to ensure that the system operates when required, performs reliably, and efficiently handles any failures that may occur.

### 1.2.1 Reasons to deploy Linux on pSeries clusters

The following are important reasons to consider for deploying Linux on pSeries clusters:

- 64-bit POWER architecture
- High performance processors
- Highly scalable systems
- Highly reliable architecture
- High memory capabilities with high memory bandwidth
- Logical partitioning support
- 64-bit price/performance leadership
- Full range of servers supporting Linux
- POWER architecture roadmap
- e-business On Demand features

## 1.3 POWER4 architecture advantages

The POWER4™ system is a new generation of high-performance 64-bit microprocessors and associated subsystems especially designed for server and supercomputing applications

Following are highlights of the many architectural advantages offered by POWER4:

- Numerous technologies are incorporated into the POWER4, to create a high-performance, highly scalable chip design to power pSeries systems.
- The POWER4 processor utilizes the IBM advanced semiconductor, Silicon-on-Insulator (SOI), and packaging technology.
- Copper interconnects provide less resistance than aluminum, which permits the use of smaller circuits with reduced latency for faster propagation of electrical signals.
- Reduced resistance and heat output make it possible to shrink electronic devices even further, while increasing clock speed and performance.
- Included in the POWER4 architecture are functions which are logically referred to as *pervasive functions*. These include trace and debug facilities used for First Failure Data Capture (FFDC), Built-In Self Test (BIST) facilities, performance monitoring unit, and interface to the Service Processor (SP) used to control the overall system, power-on Reset (POR) sequencing logic, and error detection and logging circuitry.

## 1.4 Chapter description

In this redbook, we provide information and guidelines on how to deploy Linux on IBM eServer pSeries clusters. Subsequent chapters cover these topics:

- Chapter 2, “System installation” on page 5

  This chapter contains important information you need to install Linux on pSeries systems. Pre-installation steps, installation procedures, hardware information including firmware basics, network installation, and post-installation checks are discussed.

- Chapter 3, “System administration” on page 95

  After the installation of Linux on a pSeries cluster, administration of such an environment requires special attention. This chapter includes information on the Logical Volume Manager (LVM), file systems, local user management, RPM management and updates, system updates, system backups, and user administration using LDAP.

- Chapter 4, “Linux for pSeries RAS and problem determination” on page 163

  This chapter provides information on the IBM diagnostics tools, systems logs, event logging, Linux rescue methods, and on the performance monitoring necessary to keep your system healthy. Additional tips are included to help you troubleshoot your system in the event that problems occur in your cluster.

- Chapter 6, “Getting started with GPFS” on page 279

  This chapter introduces GPFS and describes the RSCT peer domain, GPFS installation and basic configuration, GPFS on a CSM cluster, and GPFS problem determination.

- Chapter 5, “Cluster Systems Management (CSM)” on page 211

  For a Linux cluster, CSM provides the capability to manage many nodes from one single point of control. The topics discussed in this chapter are CSM concepts and architectures, planning, installation, configuration, administration, and CSM interoperability.

- Chapter 7, “High Performance Computing case studies” on page 303 and Chapter 8, “Commercial application case studies” on page 361

  In these chapters, we provide information on HPC and commercial applications.

  **Note:** These chapters do not include or discuss all the applications that can be deployed on Linux on pSeries clusters, so to obtain detailed information on all the applications that are supported, contact your IBM representative or visit the IBM Linux Web site:

  `ibm.com/linux`

- Appendix A, “Linux for AIX system administrators” on page 417 and Appendix B, “Feature comparison” on page 423

  Additional information to help you to successfully deploy Linux on pSeries servers is included on the appendix sections of this publication.

# 2

# System installation

In this chapter, we show you how to install the 64-bit Linux operating system on various types of pSeries servers, either standalone, or in LPAR mode. The steps presented here apply to the two distributions currently supported on pSeries: SLES 8 (SuSE Linux Enterprise Server 8), and RHAS 3 (Red Hat Enterprise Linux Advanced Server 3).

This chapter contains:

- 2.1, “Before you install” on page 7. This section describes the steps you need to get started. We review the contents of each distribution and the hardware currently supported. We then describe how to set up the target hardware. This covers LPARs, if needed, and serial console and graphical display. Then we discuss the choices that the installation programs prompts for during installation.
- 2.2, “Step-by-step installation” on page 25. This section presents the installation of SLES 8 and RHAS 3 in great detail. It contains many screenshots from our test bed installations, which should be useful for your own experiments.
- 2.3, “From network to fully unattended installation” on page 49. When the number of systems to be installed grows, manually installing them becomes tedious and some form of network installation is required, as discussed in this section.

  A system designated as the installation server provides the initial boot mechanism, as well a source for all the software packages to be installed on the target client systems. This way, the installation of many client nodes is performed from a single point of control.

Network installation offers other useful features, which are also described in this section. And with further customization, network installation can be automated, leading to fully unattended installations. SLES 8 and RHAS 3 both provide a mechanism for this (autoyast2 for SLES 8, and kickstart for RHAS 3), as described here.

We also examine variants of network installation, available through the Open Source community, which resemble the AIX method by using a concept similar to system backups (mksysb) and cloning.

Customers who decide to implement IBM Clusters Systems Management (CSM) do not need to worry about the details of network installation, since this is included in CSM; the management server is used as the installation server for the managed nodes. However, understanding the underlying mechanisms will help you to troubleshoot potential installation problems.

- 2.4, “Where is the BIOS” on page 79. The pSeries systems supporting the 64-bit Linux operating system are Common Hardware Reference Platform (CHRP) PowerPC® systems. Linux users may be more familiar with x86-based systems that use the Basic Input Output System (BIOS), and in this section, we address the differences.
- 2.5, “Post-installation tasks” on page 88. After the Linux operating system is installed, some tasks still need to be performed. In this section, we discuss checking the installation, backup/restore/cloning, how to install additional packages, and how to perform automatic updates of the system.

## 2.1 Before you install

According to http://gate.crashing.org/doc/ppc/doc003.htm[1], a port of the Linux kernel to the PowerPC architecture started in 1994 with Gary Thomas. Around 1997, various source code branches (Native Linux, MkLinux, Linux-pmac) merged to form the LinuxPPC developers release. From this release evolved the first commercial distributions from SuSE, Debian and Yellowdog. The kernel in these releases was a 32-bit kernel.

As described in http://penguinppc64.org/history.shtml, in 2002, IBM decided to create a port of the Linux kernel for ppc64 to run on the latest 64-bit PowerPC hardware. It is the evolution of this work that we are now using.

Following are two projects related to Linux for PowerPC:

- http://penguinppc.org is the home of the development of the 32-bit kernel to run on a variety of hardware: embedded systems based on the 4xx, and the 7xx chips from IBM, Apple desktop machines (PowerMac, PowerBook, iMac, iBook), or IBM pSeries and iSeries™ servers.
- http://penguinppc64.org is the home of the development of the 64-bit kernel to run on IBM pSeries servers (POWER3™, POWER4) and iSeries servers with POWER4 processors. The 64-bit kernel can run 32-bit and 64-bit applications.

For further information, consult the following Web sites:

- IBM PowerPC processors

  `http://www.ibm.com/chips/products/powerpc/`
- Main mailing lists for Linux on PowerPC

  `http://lists.linuxppc.org`
- Linux on pSeries portal

  `http://www.ibm.com/servers/eserver/pseries/linux/l`

Next, we describe how to get the ppc64 Linux kernel running on your 64-bit pSeries server.

### 2.1.1 Check the prerequisites

In this section, we present a checklist for you to follow before you begin installing your first pSeries system with the Linux operating system. First check that your hardware is fully supported, and that it has the correct level of firmware and

[1] Mark Hatle, based on posts to the LinuxPPC-users mailing list.

hypervisor code in case you are installing an LPAR. Also ensure that you have CDs containing the latest Linux distribution and patches at hand.

### Hardware

We cover the installation of the 64-bit Linux operating system on POWER4-based pSeries systems. At the time of writing, the supported models are:

- p615 (standalone)
- p630 (LPAR and standalone)
- p650 (LPAR and standalone)
- p655 (LPAR and standalone)
- p670, p690 (LPAR only)

As a quick guide, the following adapters are currently supported under Linux on pSeries hardware:

- Storage interfaces
  - 6203 Ultra3 SCSI
  - 6204 Differential Ultra SCSI
  - 6228 2 Gigabit Fibre Channel
  - 6239 2 Gigabit Fibre Channel PCI-X
- Communications and connectivity
  - 4962 10/100 Mbps Ethernet PCI II
  - 2969 Gigabit Ethernet - SX PCI (Fibre)
  - 2975 Gigabit Ethernet – Base-T PCI (UTP)
  - 4961 4-port 10/100 Mbps Ethernet4962 10/100 Mbps Ethernet PCI II
  - 5700 Gigabit Ethernet - SX PCI-X (Fibre)
  - 5701 Gigabit Ethernet – Base-TX PCI-X (UTP)
  - 5706 Gigabit Ethernet (UTP) 2-port
  - 5707 Gigabit Ethernet (Fibre) 2-port
- Display adapters
  - 2848 GXT135P
- Disk drives and subsystems
  - 2104-DU3/TU3 Expandable Storage Plus
  - 2105 Enterprise Storage Server® Model 800
  - 3159 73.4GB SCSI Disk, 10K rpm
  - 3277 36.4GB SCSI Disk, 15K rpm
  - 3275 146.8GB SCSI Disk, 10K rpm
  - FAStT200/500/600/700/900 Storage Servers
- Fibre channel directors, switches and hubs
  - 2031-032/224/232 McDATA Fibre Chan Switch
  - 2032-064/140 McDATA Fibre Channel Director

  - 2042-128/256 INRANGE FC/9000 Fibre Dir
  - 2062-D01/D02/T07 Cisco Fabric Switch / Dir
  - 3534-F08 SAN Switch
  - 2109-F16/F32 SAN Switch
- Optical drives and libraries
  - 2624 32X/40X CD-ROM
  - 2628 40X CD-ROM auto-dock
  - 2629 4.7GB SCSI DVD-RAM

> **Note:** The most recent list of supported hardware and features is available at:
>
> `http://www.ibm.com/servers/eserver/pseries/hardware/linux_facts.pdf`
>
> For more general news, announcements, resources, consult the IBM portal for Linux on pSeries:
>
> `http://www.ibm.com/servers/eserver/pseries/linux/`

## Firmware levels

We were able to install SLES 8 and RHAS 3 on p650 LPARs and a p630 with these levels of firmware:

- p650: 3K030916
- p630: 3R030501

To check your current level of firmware on an already running system, use the **`lscfg`** command. Under Linux, this command is not part of the SLES 8 or RHAS 3 distributions. Refer to 4.1.2, “IBM diagnostics tools” on page 168 for information on how to set it up under Linux. If you need to upgrade the firmware, refer to:

`http://www.ibm.com/servers/eserver/support/pseries/fixes/hm.html`

If you are already running Linux in a partition or in standalone mode and you wish to upgrade the firmware, you can do it from Linux. The procedure is detailed in 4.1.2, “IBM diagnostics tools” on page 168.

## HMC code levels for LPAR mode

The HMC that we used to install the p650 LPARs was running version 3.2.4. If you need to upgrade the HMC code, check:

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

More information about the HMC can be found in the IBM Redbook: *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038.

## Distributions

Today, the officially supported distributions for installing Linux on pSeries are available at the following locations:

- SuSE SLES 8, available at:

  http://www.suse.de/us/business/products/server/sles/i_pseries.html

- Turbolinux TLES 8, available at:

  http://www.turbolinux.com/products/eserver/#pSeries

- Red Hat RHAS 3, available at:

  http://www.redhat.com/software/rhel/as/

Turbolinux Enterprise Server (TLES 8) from Turbolinux is identical to SuSE SLES 8. Both implement the UnitedLinux 1.0 specifications[2]. From now on, we will refer only to SLES 8—but what we write about SLES 8 applies equally to TLES 8.

A more adventurous reader may wish to try alternative distributions. These distributions feature a 32-bit kernel that can be used for cross-compiling the 64-bit kernel. You install a 32-bit PowerPC distribution on some machine (Apple G4, IBM B50) to build the toolchain (PowerPC64 cross-compiler and utilities) in order to compile a 64-bit kernel to be booted on the target machine. It is very likely that some of the following distributions will feature a 64-bit kernel natively in the future:

- Gentoo

  http://www.gentoo.org

- Debian

  http://www.debian.org/ports/powerpc

- Mandrake

  http://www.linux-mandrake.com/en/ppc.php3

- Yellow Dog
- http://www.yellowdoglinux.com/
- Knoppix
- http://debian.tu-bs.de/knoppix/powerPC/

A document written by Manuel Lässer, Ralf Strauss and Florian M. Weps is available that gives detailed instructions on how to install a Debian “Woody” distribution on an IBM pSeries p630. The installation server used is an Apple iBook.This document can be found at:

[2] UnitedLinux is a consortium of Conectiva, SCO Group, SuSE and Turbolinux that aims at producing a unified professional Linux distribution. It is based on SuSE SLES 8.

http://people.debian.org/~fmw/p630-LPAR-Debian-en.txt

## SLES 8 contents

SLES 8 is a full-featured Linux distribution. The complete list of applications can be found at:

http://www.suse.com/us/business/products/server/sles/sles_apps/i_pseries.html

It contains nearly 1000 packages. The main features are as follows:

- UnitedLinux brand 1.0
- LSB certification 1.3[3]
- Kernel 2.4.21
- 32-bit and 64-bit application support
- Glibc 2.2.5
- Gcc 3.2.2
- Cross-compiler suite for generating 64-bit applications
- LVM 1.0.5
- EVMS 1.4.0
- File systems supported: ext2, ext3, JFS, ReiserFS
- Heartbeat 0.4.9e
- XFree86 4.2
- KDE 3.0.3
- GNOME 2.0.6
- Apache 1.3.26
- PHP 4.2.2
- MySQL 3.23.52
- PostgreSQL 7.2.2

The distribution contains three CDs. Only the first CD is bootable and required for a minimal install. SuSE provides also some supplementary CDs containing source RPMS and documentation for all the packages.

## RHAS 3 contents

Red Hat recently announced its Red Hat Enterprise Linux Advanced Server Version 3 for iSeries, pSeries and zSeries®. For a complete description of RHAS 3, refer to:

http://www.redhat.com/software/rhel/whitepapers/

http://www.redhat.com/software/rhel/features/

[3] LSB, or Linux Standard Base, aims at improving the compatibility between Linux distributions. It defines common rules for compiled applications and installation scripts that defines a uniform industry standard. LSB addresses such issues as object formats, dynamic linking, libraries, package formats, and so on.

The main features of RHAS 3 are:

- LSB certification 1.3
- Kernel 2.4.21
- Glibc 2.3.2
- 32-bit and 64-bit application support
- Gcc 3.2.3
- Cross-compiler suite for generating 64-bit applications
- LVM 1.0.3
- File systems supported: ext2, ext3
- XFree86 4.3
- KDE 3.0.3
- GNOME 2.0.6
- Apache 2.0.46
- PHP 4.3.2
- MySQL 3.23.58
- PostgreSQL 7.3.4

The distribution contains four CDs. Only the first CD is bootable. CD1, CD2 and CD3 are required, even for a minimal install. Red Hat also provides five supplementary CDs containing source RPMS and documentation for all the packages. Red Hat also distributes an additional Linux Applications CD (LACD) that contains the IBM Java™ runtime and SDK versions 1.4.1.

### Where to get the distributions

Neither SLES 8 nor RHAS 3 can be freely downloaded from the Web. Unless you purchase a Linux-ready configuration, you have to order SLES 8 or RHAS 3 separately. For more information, check the following:

http://www.suse.com/us/business/products/server/sles/prices.html

http://www.redhat.com/apps/commerce/rhel/as/

The SLES 8 CD1 contains extensive installation documentation in the /docu/ directory. Be sure to have a copy at hand.

The RHAS 3 CD1 contains release notes in the root directory. Complete documentation can be found on the documentation CD. You can find more information about installing RHAS 3 on pSeries from:

http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/ppc-multi-install-guide/

## 2.1.2 Prepare the system

Before installing, verify that all your hardware is supported. Exotic adapters should be removed temporarily during the installation. This applies, for example, for network adapters that have not yet been certified. Also be sure to make a note of the operating systems that might be previously installed, because you may, for example, have an AIX installation on some of the disks that you wish to preserve.

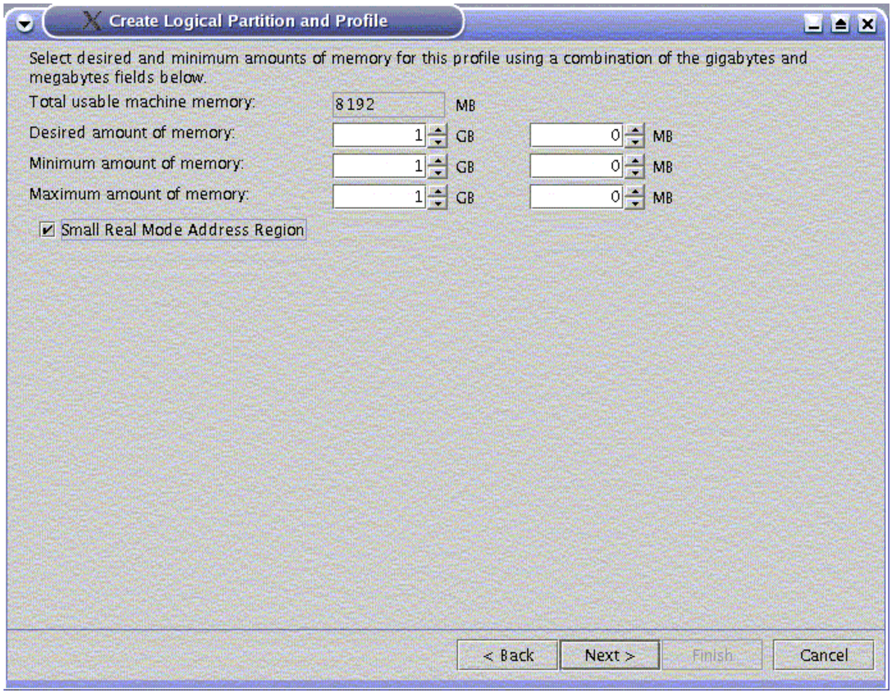
## LPAR

If you are installing an LPAR, you must define it before starting the installation process. Refer to the following IBM documentation on HMC and LPAR:

- *The Complete Partitioning Guide for IBM eServer pSeries Servers*, SG24-7039
- *Configuring p690 in an IBM eServer Cluster 1600*, REDP-0187
- *IBM eServer Certification Study Guide - p690 Technical Support*, SG24-7195
- *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038

Dynamic LPAR is not supported for a Linux partition. If you only have Linux or AIX 5.2 partitions, it is recommended that you activate the Small Real Mode Address Region, which allows for more flexibility in the way the memory can be partitioned.

When defining the LPAR on the HMC, make sure you check this option, as shown in Figure 2-1 on page 14.



Select desired and minimum amounts of memory for this profile using a combination of the gigabytes and megabytes fields below.

Total usable machine memory: 8192 MB

Desired amount of memory: 1 GB 0 MB

Minimum amount of memory: 1 GB 0 MB

Maximum amount of memory: 1 GB 0 MB

☑ Small Real Mode Address Region

*Figure 2-1 Small Real Mode Address Region selection*

Our experimental setup consists of one standalone p630 server and four p650 systems with two LPARs.

## 2.1.3 Monitoring the installation

The installation programs from SuSE and Red Hat can run in text mode or in graphic mode, if the system is equipped with a graphical adapter. If the system being installed does not have a graphical adapter (for example, rack-mounted systems), a graphical installation is still possible using Virtual Network Computing (VNC).

### Graphics adapter issues

The p615 and p630 can be fitted with the GXT135P graphics adapter (FC 2848 or 2849). Some adapters have a single DB15 connector for analog displays and are trouble-free. The more recent adapters come with a DB15 connector for analog displays and also a DVI connector for digital displays. This could be troublesome for versions of SLES 8 prior to SP3 if using an analog display, because the default output port for the GXT135P-2848 is the DVI port. The installer program that comes with SLES 8 SP3 can handle this configuration now. This means that you have to boot the SLES 8 SP3 CD1 and then, when prompted by the installation program, insert the base SLES 8 CD1 for the remaining part of the installation.

You may find very useful information in the SLES 8 release notes at:

```
http://www.ibm.com/servers/eserver/pseries/linux/sles8_release_notes.pdf
```

This document contains a section on graphics adapter issues. In some cases, it might be necessary to connect an ASCII console to perform the installation.

## VNC installation

VNC (Virtual Network Computing from http://www.realvnc.com/) is a cross-platform client-server application for remote graphical display. There are clients and servers available for many different operating systems, including AIX, Windows®, MacOS and Linux.

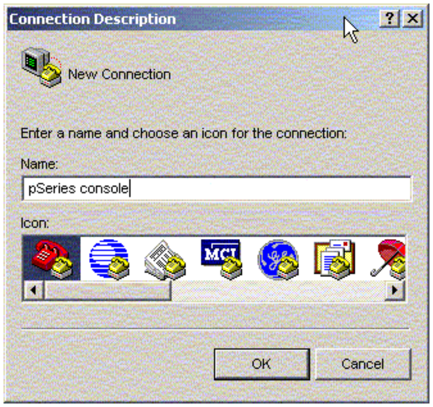
Both SuSE and Red Hat installers can start a VNC server during installation. A VNC client needs to be started on a system that has graphical capabilities itself and that is directly connected on the same network as the system being installed. SuSE and RedHat have slightly different ways of telling the installer to start the VNC server and the VNC client, and these are detailed in the following sections.

## Setting up a serial connection

The easiest way to set up a serial connection is to connect a 3151 type terminal to the first serial connector of the system being installed. Or you can use a null modem cable and connect to a laptop and use HyperTerminal on Windows or Minicom on Linux.

### *HyperTerminal*

HyperTerminal can be started from the **Start** -> **Programs** -> **Accessories** -> **Communication** menu on most Windows systems. Upon startup you are asked to name the session, as shown in Figure 2-2 on page 16.


Connection Description

New Connection

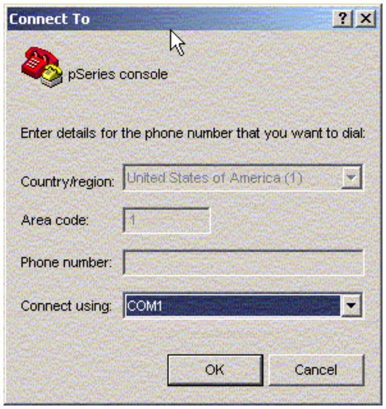
Enter a name and choose an icon for the connection:

Name: pSeries console

Icon:

*Figure 2-2 HyperTerminal: naming the session*

Then you must select the serial port on which you connected the null modem cable. This is usually COM1, as shown in Figure 2-3.


Connect To

pSeries console

Enter details for the phone number that you want to dial:

Country/region: United States of America (1)

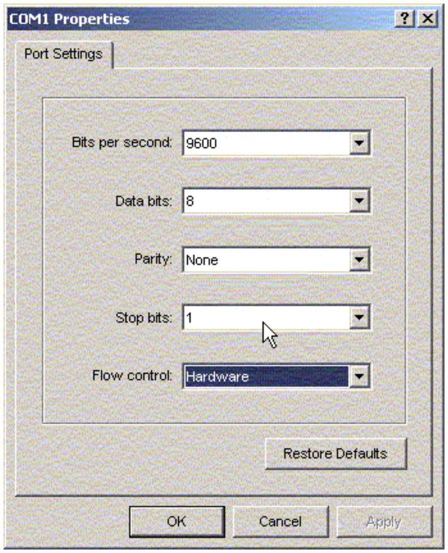
Area code: 1

Phone number:

Connect using: COM1

*Figure 2-3 HyperTerminal: choosing the serial port*

The port settings need to be changed to 9600 bauds, 8N1 as shown in Figure 2-4 on page 17.



COM1 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: Hardware

*Figure 2-4 HyperTerminal: serial port settings*

You should now be connected; Figure 2-5 on page 18 shows the HyperTerminal screen.

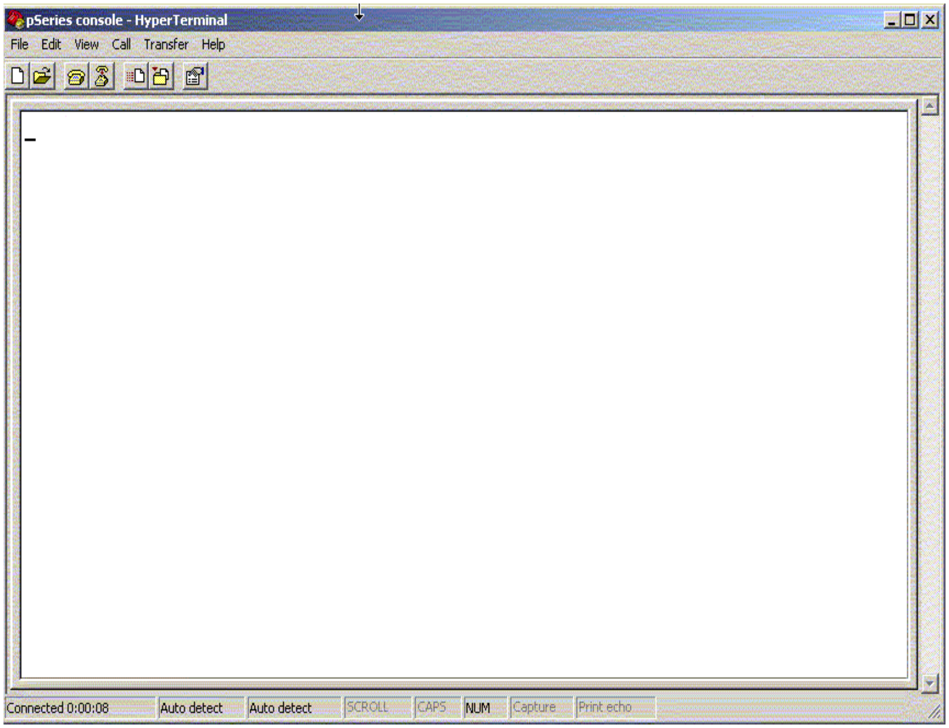


*Figure 2-5 HyperTerminal*

Pressing Enter while in this HyperTerminal screen should connect you to the attached system, in the service processor menu, if the system has just been powered up.

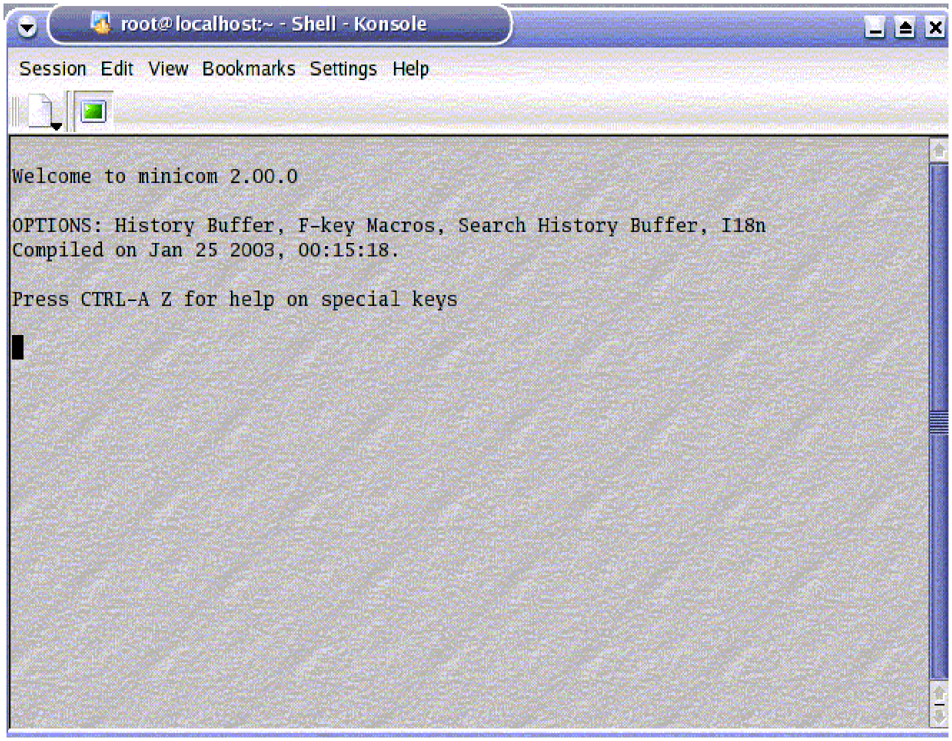
### Minicom

Minicom is a popular serial communication program that can be found on any Linux system. Once your null modem cable is connected to a serial port, you can start it by typing: `minicom`.

**Tip:** On some IBM Thinkpad T30s, the first serial device is disabled in the BIOS. Windows enables it itself, but Linux does not.

To activate it permanently for use by Linux, you must either enable it in the BIOS, or through a Windows utility program that can be found under C:\Program Files\ThinkPad®\Utilities\PS2.exe; for example:

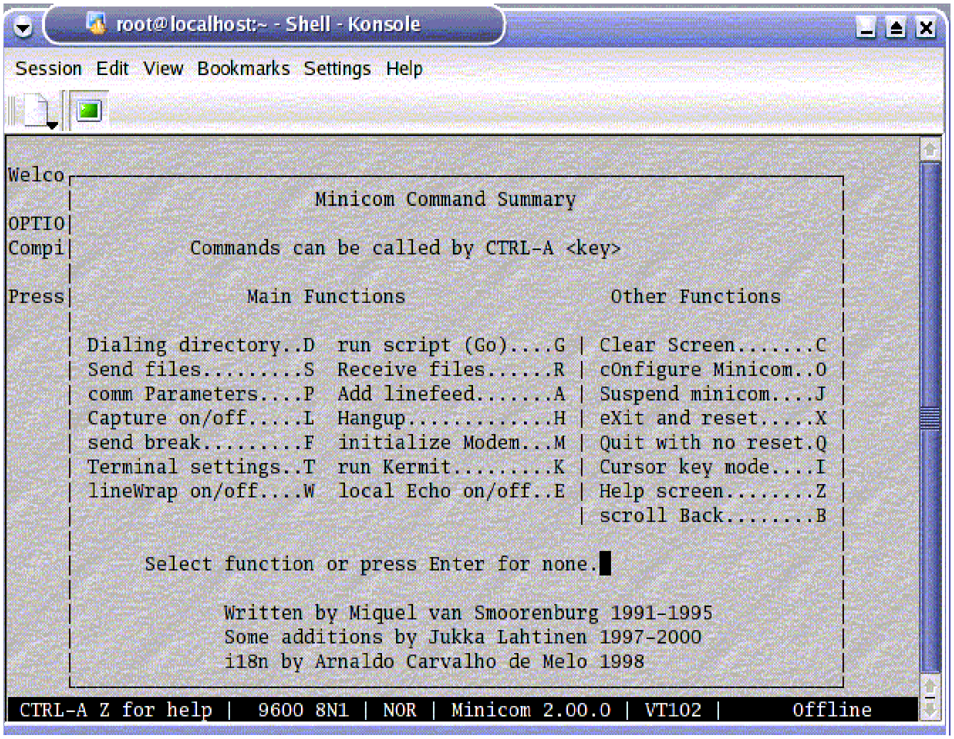
```
C:>ps2 ? sera # to check the current status
C:>ps2 sera enable # to enable it
```


root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
Welcome to minicom 2.00.0

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Jan 25 2003, 00:15:18.

Press CTRL-A Z for help on special keys
```

*Figure 2-6 The minicom startup screen*

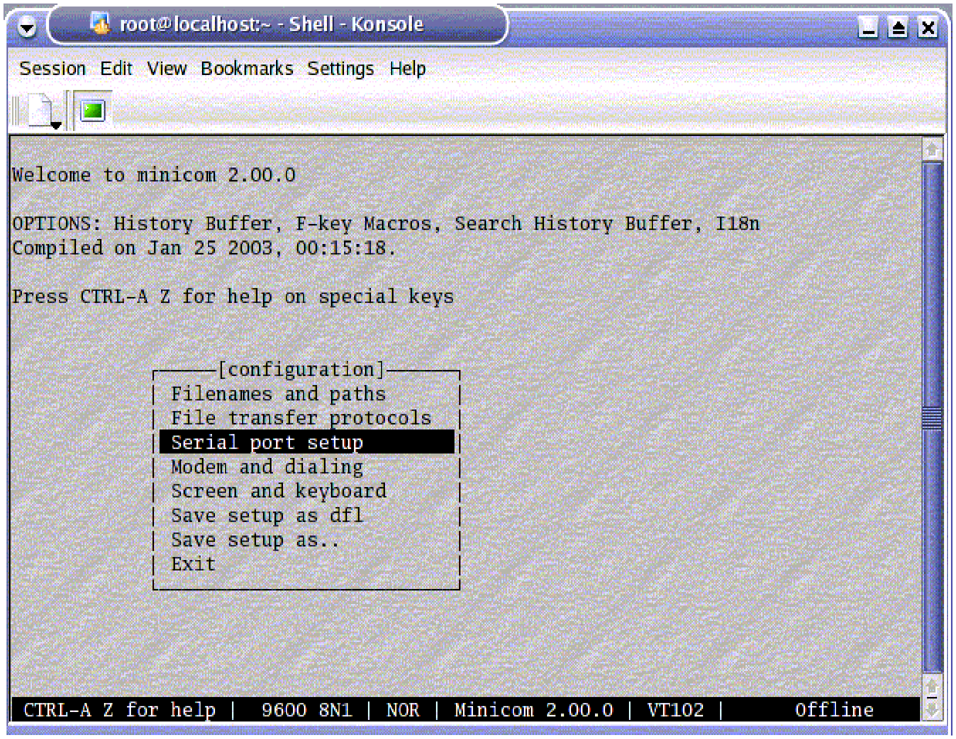
Pressing Ctrl-A then Z brings up the help screen, as shown in Figure 2-7 on page 20.


root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
Welco┌──────────────────────────────────────────────────────────────────┐
     |                    Minicom Command Summary                       |
OPTIO|                                                                  |
Compi|          Commands can be called by CTRL-A <key>                  |
     |                                                                  |
Press|               Main Functions                  Other Functions    |
     |                                                                  |
     | Dialing directory..D  run script (Go)....G | Clear Screen.......C |
     | Send files.........S  Receive files......R | cOnfigure Minicom..O |
     | comm Parameters....P  Add linefeed.......A | Suspend minicom....J |
     | Capture on/off.....L  Hangup.............H | eXit and reset.....X |
     | send break.........F  initialize Modem...M | Quit with no reset.Q |
     | Terminal settings..T  run Kermit.........K | Cursor key mode....I |
     | lineWrap on/off....W  local Echo on/off..E | Help screen........Z |
     |                                            | scroll Back........B |
     |                                                                  |
     |      Select function or press Enter for none.                    |
     |                                                                  |
     |             Written by Miquel van Smoorenburg 1991-1995          |
     |             Some additions by Jukka Lahtinen 1997-2000           |
     |             i18n by Arnaldo Carvalho de Melo 1998                |
     └──────────────────────────────────────────────────────────────────┘
 CTRL-A Z for help |  9600 8N1 | NOR | Minicom 2.00.0 | VT102 |      Offline
```

*Figure 2-7 The minicom help screen*

In the minicom help screen, select **O**. Next, select **Serial port setup** to choose the serial port and its settings, as shown in Figure 2-8 on page 21.


root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
Welcome to minicom 2.00.0

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Jan 25 2003, 00:15:18.

Press CTRL-A Z for help on special keys


            ┌─────[configuration]─────┐
            | Filenames and paths     |
            | File transfer protocols |
            | Serial port setup       |
            | Modem and dialing       |
            | Screen and keyboard     |
            | Save setup as dfl       |
            | Save setup as..         |
            | Exit                    |
            └─────────────────────────┘

 CTRL-A Z for help |  9600 8N1 | NOR | Minicom 2.00.0 | VT102 |      Offline
```

*Figure 2-8 The minicom configuration screen*

The first serial port on a PC running Linux is usually named /dev/ttyS0. It corresponds to COM1 for Windows.

If you are using another serial port, change it accordingly as shown in Figure 2-8. To check if a particular port is connected, use the **stty** command as shown in Example 2-1.

*Example 2-1 stty command*

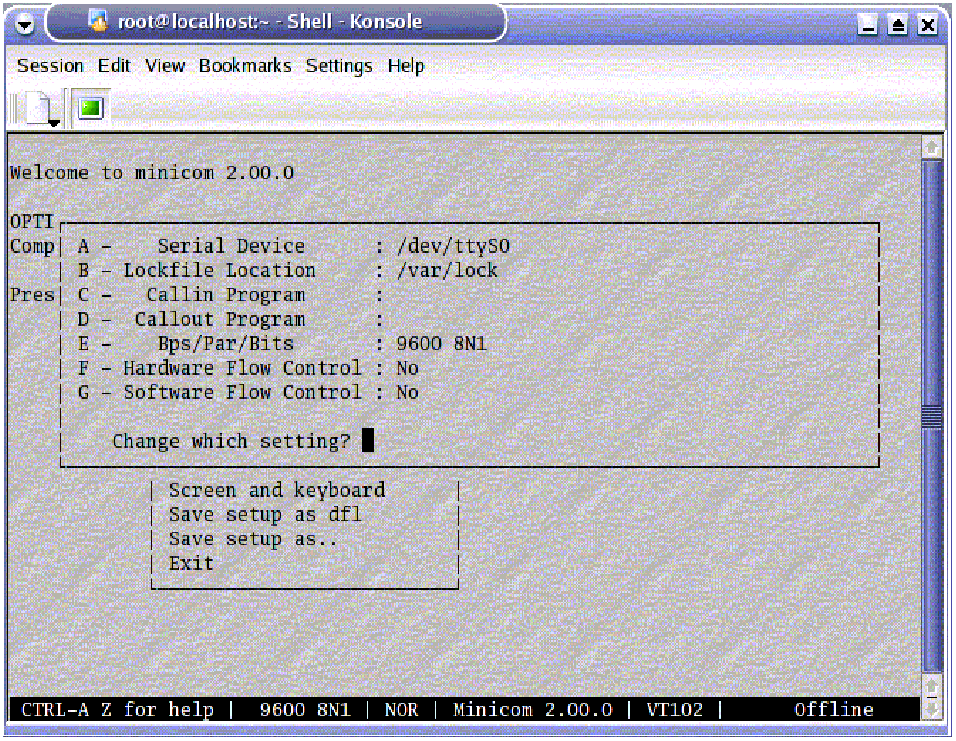
```
# stty -a -F /dev/ttyS0
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^U; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

Using **stty** on a device that is not connected results in an error, as shown in Example 2-2.

*Example 2-2 stty error*

```
# stty -a -F /dev/ttyS5
stty: /dev/ttyS5: Input/output error
```

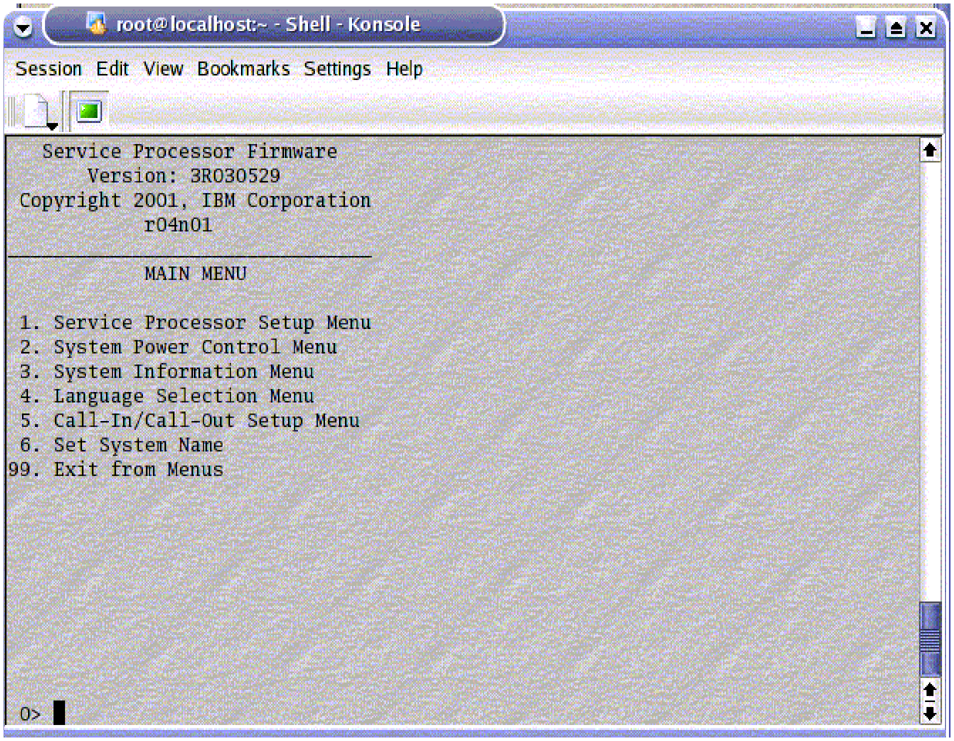
If the serial port settings must be changed, refer to Figure 2-9 for information on how to change the serial port settings.


root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
Welcome to minicom 2.00.0

OPTI┌────────────────────────────────────────────────────────────────────────┐
Comp│ A -    Serial Device      : /dev/ttyS0                                 │
    │ B - Lockfile Location     : /var/lock                                  │
Pres│ C -   Callin Program      :                                            │
    │ D -  Callout Program      :                                            │
    │ E -    Bps/Par/Bits       : 9600 8N1                                   │
    │ F - Hardware Flow Control : No                                         │
    │ G - Software Flow Control : No                                         │
    │                                                                        │
    │    Change which setting?                                               │
    └────────────────────────────────────────────────────────────────────────┘
            │ Screen and keyboard        │
            │ Save setup as dfl          │
            │ Save setup as..            │
            │ Exit                       │
            └────────────────────────────┘

 CTRL-A Z for help |  9600 8N1 | NOR | Minicom 2.00.0 | VT102 |      Offline
```

*Figure 2-9 minicom: changing the serial port settings*

After you save the settings, reinitialize the connection by pressing Ctrl A + M keys. Then press Enter and the pSeries console appears; see Figure 2-10 on page 23.


root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
   Service Processor Firmware
       Version: 3R030529
 Copyright 2001, IBM Corporation
            r04n01
--------------------------------
            MAIN MENU

 1. Service Processor Setup Menu
 2. System Power Control Menu
 3. System Information Menu
 4. Language Selection Menu
 5. Call-In/Call-Out Setup Menu
 6. Set System Name
99. Exit from Menus







 0>
```

*Figure 2-10 minicom: the pSeries console*

## 2.1.4 Review your choices

Whichever distribution you are now ready to install, you will be asked a few questions during the installation. In this section, we briefly review the choices.

### Boot options

Once you boot the installation media, you may have to enter some boot options for the installation program. This is where you specify your interface to the installation program as text mode, graphical mode, or VNC.

**Tip:** If you have adapter cards that do not support the Enhanced Error Handling (EEH) feature, then add eeh-force-off to your boot options. For example, the Myricom PCI-X adapters do not support EEH. Add the eeh-force-off to the append option in the /etc/lilo.conf boot loader configuration file, as shown:

```
image = /boot/vmlinuz

   label = linux

   root = /dev/sda3

   append = " eeh-force-off"
```

A typical invocation is shown in Example 2-3 for SLES 8.

*Example 2-3 boot options for SLES 8*

```
Welcome to yaboot version 1.3.6.SuSE
Enter "help" to get some basic usage information

boot: install vnc=1 vnc_password=itsoadmin dhcp=1
```

Example 2-4 shows the typical invocation used for RHAS 3.

*Example 2-4 boot options for RHAS 3*

```
Welcome to yaboot version 1.3.10
Enter "help" to get some basic usage information

boot: linux vnc=1 vncconnect=192.168.100.2
```

### Type of installation

The media can be used to install a new system or to boot an already installed system in case of trouble with the boot loader. In our case, we choose a new installation.

### Disk partitioning

You will make important decisions in this step. You have to choose if you are going to use Logical Volume Manager (LVM), which type of file system you wish to have and the size of each of the file systems you intend to define in this step. The installers offer default configurations which are sensible, but you may wish to make changes.

LVM is described in great detail in 3.6, “Logical Volume Manager (LVM)” on page 113. LVM is supported by both distributions. As for the file systems, the

SLES 8 installer proposes ext2/ext3, ReiserFS and JFS. The RHAS 3 installer supports only ext2/ext3.

### Software selection

Both distributions offer minimal, standard, full and customizable software configurations. The minimal installation is useful for testing purposes as it is quick to install. However, a product like GPFS will not install with a minimal installation because it requires compilers to be available.

### Kernel modules for your hardware

During installation, you may have to load or select kernel modules for your hardware. Table 2-1 lists the most common drivers.

*Table 2-1 Kernel module names*

| Feature codes | Description | Linux driver |
|---|---|---|
| 4962 | 10/100 Mbps Ethernet PCI II | e100 |
| 5700 | Gigabit Ethernet | e1000 |
| 6203 | Ultra SCSI 3 | sym83c8xx (built in on SLES 8) |
| 6239 | Gigabit Fibre Channel PCI-X | lpfcdd |

## 2.2 Step-by-step installation

In this section, we describe how to install Linux on a pSeries system from CDs, either a standalone system or a LPAR. This is viable for a limited number of systems. If more systems need to be installed concurrently, network installation is the way to go.

Still, if you decide to implement the installation server on a Linux pSeries system, this server needs to be installed first, and this will likely be from CD. Also, apart from the boot process (which is different if you boot from CD or from the installation server), most of the screens that the installer program will display are identical.

For both distributions, the installation proceeds as follows:

- Start the system from the Systems Management Services (SMS) menu.
- Select the boot device to be the CD-ROM.
- Boot the installation media.
- Enter install options at the boot: prompt.
- Apply the choices you made in 2.1.4, “Review your choices” on page 23 when you are prompted for input.

- Perform the final configuration tasks (root password, additional userid, network cards configurations, boot loader configuration).

**Important:** Currently, neither SLES 8 nor RHAS 3 provide commands to alter the boot list of a pSeries system. Therefore, after the initial installation, you will have to set the boot list order from the SMS menu and put in first position the hard disk you have just installed Linux on.

### 2.2.1 SLES 8 installation

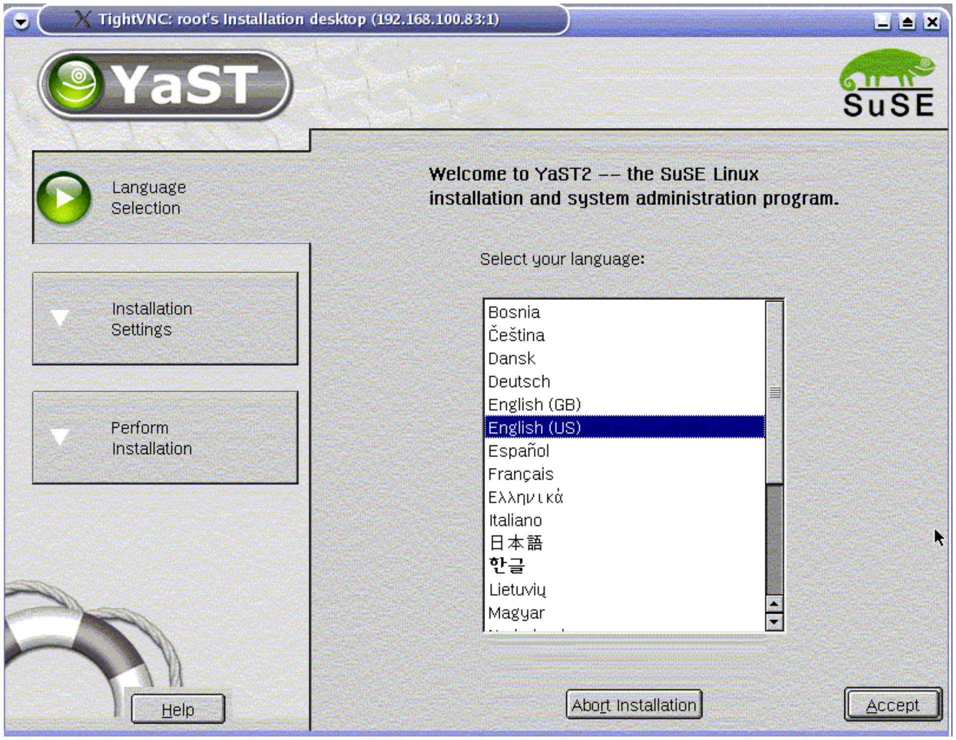
If you are installing a system with a DVI graphics adapter, you should boot the SLES 8 SP3 CD1 to avoid the issues described in 2.1.3, “Monitoring the installation” on page 14. Soon after, you will be asked to insert the base SLES 8 CD1.

We assume a VNC installation. We monitor the installation from a system that is connected to the same network as the system you are installing.

The VNC installation is enabled at the boot prompt, as shown in Example 2-3 on page 24. One of the first steps of the installation is to load the Ethernet module so that the installation program can communicate with the VNC client. You can choose a static IP address or use DHCP. You then need to start your VNC client and point it to the IP address of the system being installed.

YaST2 then takes control of the installation; the first screen that you will see should look similar to Figure 2-11 on page 27.

YaST2 is the SuSE system administration tool. It is described in some detail in 3.2, “To YaST or not to YaSt” on page 97.


TightVNC: root's Installation desktop (192.168.100.83:1)

Welcome to YaST2 -- the SuSE Linux installation and system administration program.

Select your language:

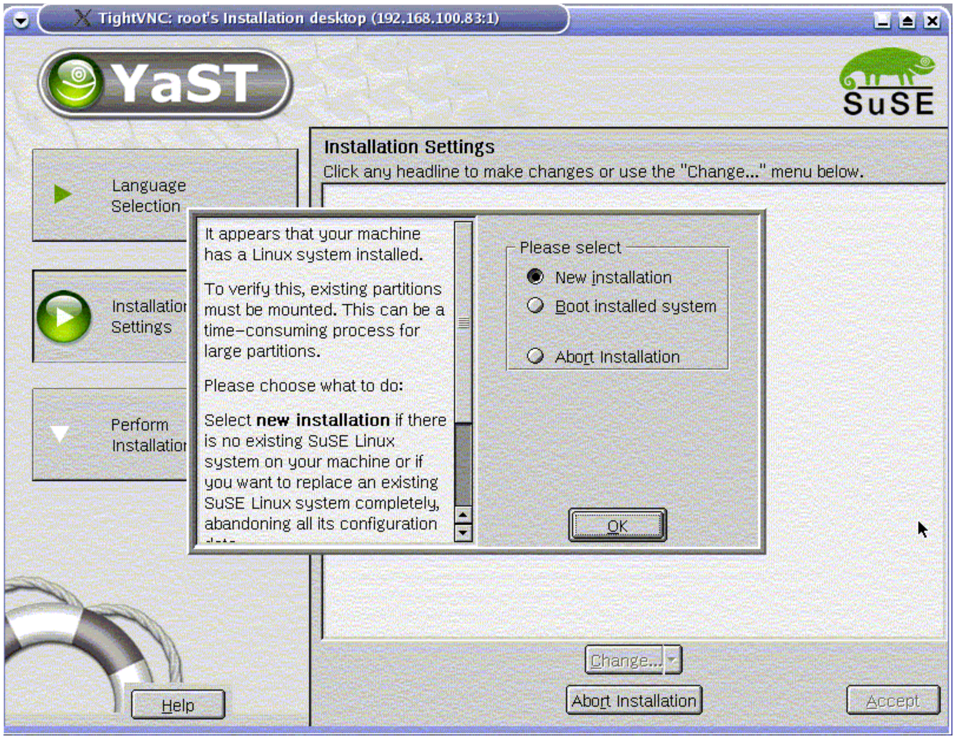
- Bosnia
- Čeština
- Dansk
- Deutsch
- English (GB)
- English (US)
- Español
- Français
- Ελληνικά
- Italiano
- 日本語
- 한글
- Lietuvių
- Magyar

*Figure 2-11 Language settings*

The installation media can also be used to boot a system that is already installed. This can be useful in case the boot partition is damaged. With this mode, you will be able to choose the root partition to be mounted.

If the disk partitions are damaged and cannot be mounted, the same installation media can be used in rescue mode, where a minimal root partition will be mounted from CD-ROM with the necessary utilities to fix disk partition problems. This is described in great detail in 4.5, “Linux rescue methods” on page 184.

At this point, however, we are interested in installing a new system, as shown in Figure 2-12 on page 28.



Installation Settings

Click any headline to make changes or use the "Change..." menu below.

It appears that your machine has a Linux system installed.

To verify this, existing partitions must be mounted. This can be a time–consuming process for large partitions.

Please choose what to do:

Select **new installation** if there is no existing SuSE Linux system on your machine or if you want to replace an existing SuSE Linux system completely, abandoning all its configuration

Please select

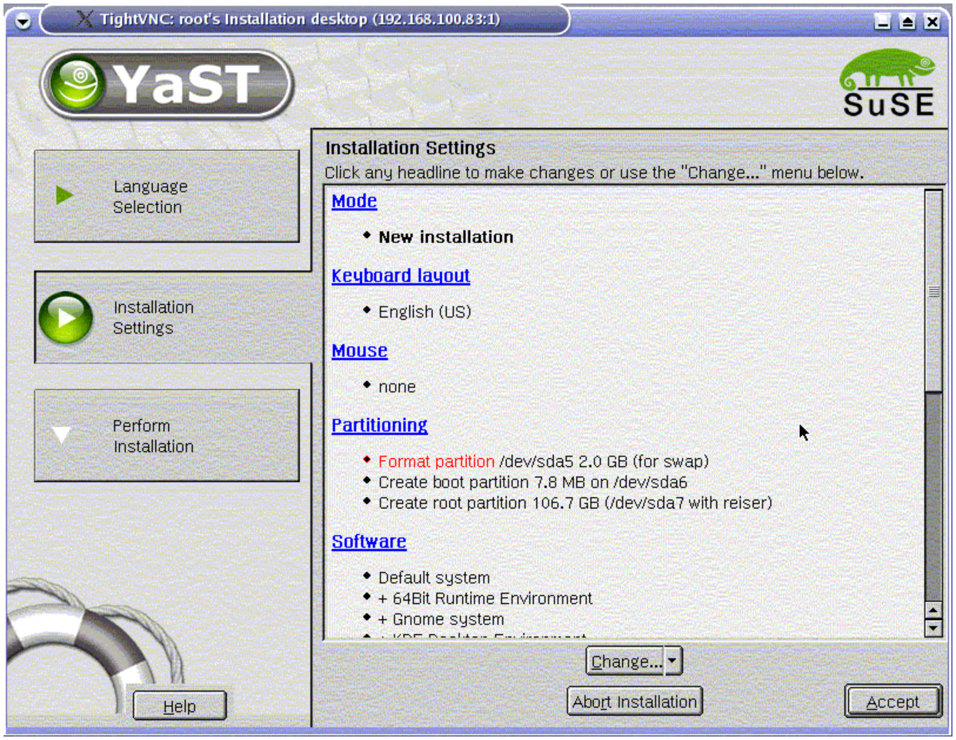
● New installation

○ Boot installed system

○ Abort Installation

*Figure 2-12 Type of installation*

The next screen, shown in Figure 2-13 on page 29, summarizes the current choices.



Installation Settings

Click any headline to make changes or use the "Change..." menu below.

Mode

- New installation

Keyboard layout

- English (US)

Mouse

- none

Partitioning

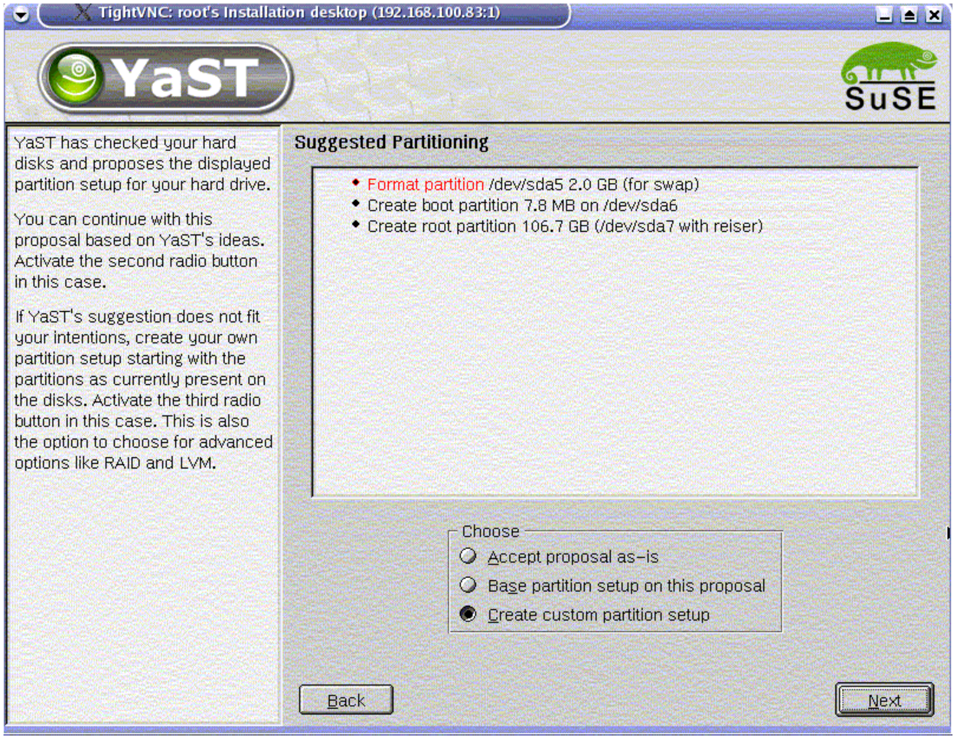
- Format partition /dev/sda5 2.0 GB (for swap)
- Create boot partition 7.8 MB on /dev/sda6
- Create root partition 106.7 GB (/dev/sda7 with reiser)

Software

- Default system
- + 64Bit Runtime Environment
- + Gnome system

*Figure 2-13 Current installation settings*

You may decide to change the partitioning settings. This is shown in Figure 2-14 on page 30.


TightVNC: root's Installation desktop (192.168.100.83:1)

YaST has checked your hard disks and proposes the displayed partition setup for your hard drive.

You can continue with this proposal based on YaST's ideas. Activate the second radio button in this case.

If YaST's suggestion does not fit your intentions, create your own partition setup starting with the partitions as currently present on the disks. Activate the third radio button in this case. This is also the option to choose for advanced options like RAID and LVM.

Suggested Partitioning

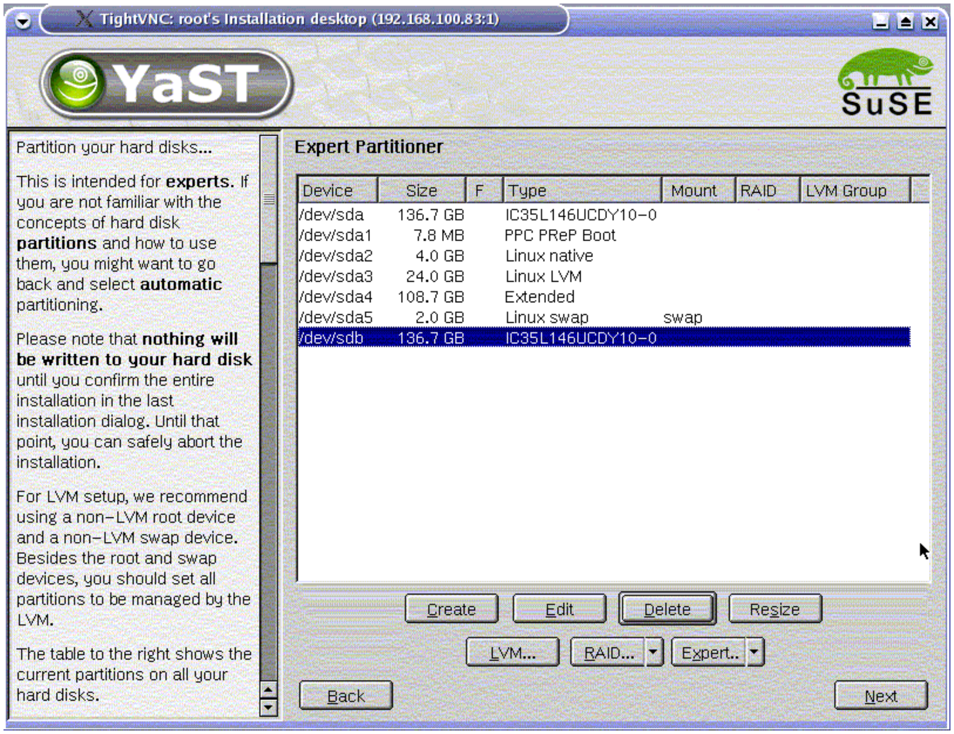
- Format partition /dev/sda5 2.0 GB (for swap)
- Create boot partition 7.8 MB on /dev/sda6
- Create root partition 106.7 GB (/dev/sda7 with reiser)

Choose

○ Accept proposal as-is
○ Base partition setup on this proposal
● Create custom partition setup

*Figure 2-14 Choose a custom partitioning scheme*

The “Expert partitioning” mode allows for the greatest flexibility. The initial screen is depicted in Figure 2-15 on page 31.



Partition your hard disks...

This is intended for **experts.** If you are not familiar with the concepts of hard disk **partitions** and how to use them, you might want to go back and select **automatic** partitioning.

Please note that **nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

For LVM setup, we recommend using a non-LVM root device and a non-LVM swap device. Besides the root and swap devices, you should set all partitions to be managed by the LVM.

The table to the right shows the current partitions on all your hard disks.

Expert Partitioner

| Device | Size | F | Type | Mount | RAID | LVM Group |
|---|---|---|---|---|---|---|
| /dev/sda | 136.7 GB | | IC35L146UCDY10-0 | | | |
| /dev/sda1 | 7.8 MB | | PPC PReP Boot | | | |
| /dev/sda2 | 4.0 GB | | Linux native | | | |
| /dev/sda3 | 24.0 GB | | Linux LVM | | | |
| /dev/sda4 | 108.7 GB | | Extended | | | |
| /dev/sda5 | 2.0 GB | | Linux swap | swap | | |
| /dev/sdb | 136.7 GB | | IC35L146UCDY10-0 | | | |

*Figure 2-15 Expert partitioning*

Once you decide on which disks you wish to create your partitions, you have to specify the partition type (swap, boot partition, Linux) and the type of file systems that you will define on this partition.

## Boot partition

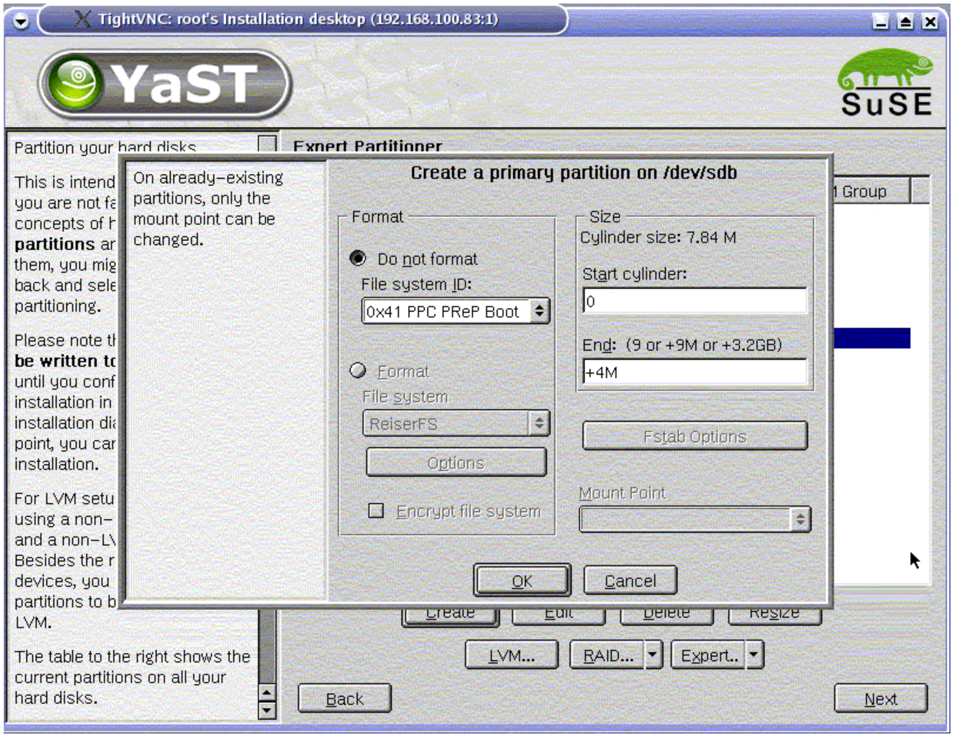
You must define a PReP (PowerPC Reference platform) boot partition; otherwise, you will not be able to boot the system off disk.

**Note**: As reported by some authors[4], we have found that the PReP boot partition must not be larger than 8 MB. We recommend using the default proposed by the SLES 8 installer.

The only file that will be written (raw) to this partition is the /boot/yaboot.chrp file, which is currently around 300 Kilobytes.

[4] Todd Inglett in: http://lists.terrasoftsolutions.com/yellowdog-general/January01/0082.html; Krisztian Mark Szentes inhttp://lists.debian.org/debian-powerpc/2003/debian-powerpc-200306/msg00309.html

Creating the PReP boot partition is shown in Figure 2-16. You will also need to define a swap partition.



TightVNC: root's Installation desktop (192.168.100.83:1)

Partition your hard disks

This is intend
you are not fa
concepts of h
**partitions** ar
them, you mig
back and sele
partitioning.

Please note th
**be written to**
until you conf
installation in
installation dia
point, you can
installation.

For LVM setu
using a non-
and a non-LV
Besides the r
devices, you
partitions to b
LVM.

The table to the right shows the current partitions on all your hard disks.

Expert Partitioner

On already-existing partitions, only the mount point can be changed.

Create a primary partition on /dev/sdb

Format

● Do not format

File system ID: 0x41 PPC PReP Boot

○ Format

File system: ReiserFS

□ Encrypt file system

Size

Cylinder size: 7.84 M

Start cylinder: 0

End: (9 or +9M or +3.2GB) +4M

Mount Point

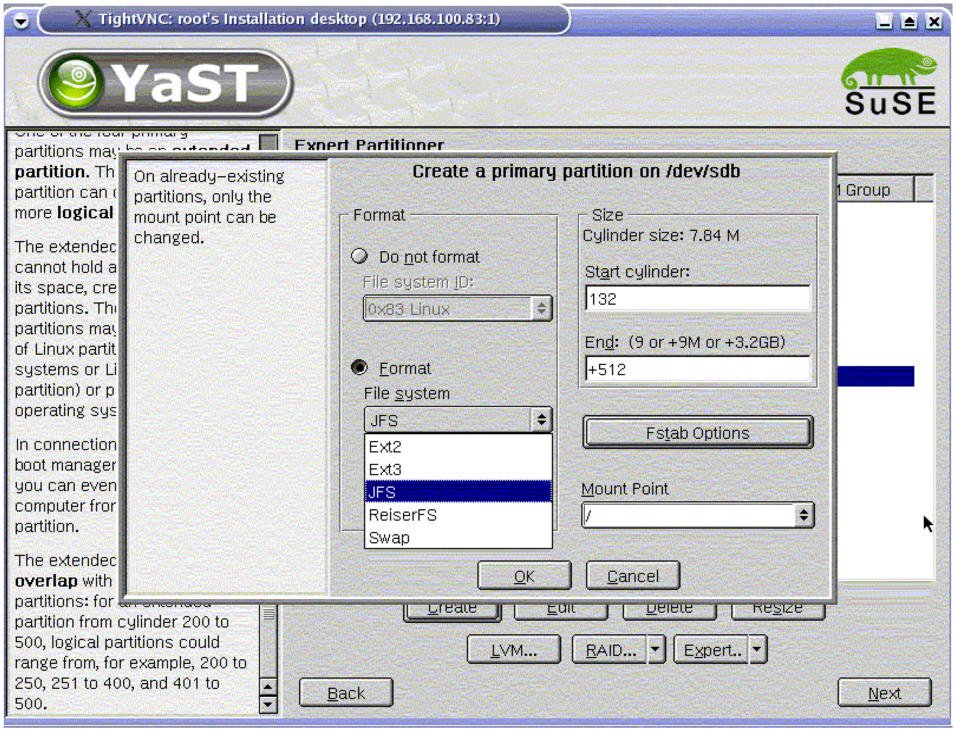
*Figure 2-16 Create a PReP boot partition*

## Linux partitions and file systems

Next, you need to define partitions for your data. You can choose to define a single root partition. You can also define more partitions for /usr, /home, /var, and so on, with or without the LVM. Refer to 3.6, "Logical Volume Manager (LVM)" on page 113 for a detailed discussion on LVM.

We do not recommend using LVM for the root partition, but it is fine to use it for any other partitions. It adds great flexibility to the way disk storage is managed.

The creation of a root partition with JFS as the file system type is shown in Figure 2-17. In our example, we decided to create a rather small partition for /, and we use LVM for all the other file systems. Generally speaking, it makes things more difficult to have the root partition as a logical volume.



partition from cylinder 200 to 500, logical partitions could range from, for example, 200 to 250, 251 to 400, and 401 to 500.

On already–existing partitions, only the mount point can be changed.

**Create a primary partition on /dev/sdb**

Format

○ Do not format

File system ID: 0x83 Linux

● Format

File system: JFS

Size

Cylinder size: 7.84 M

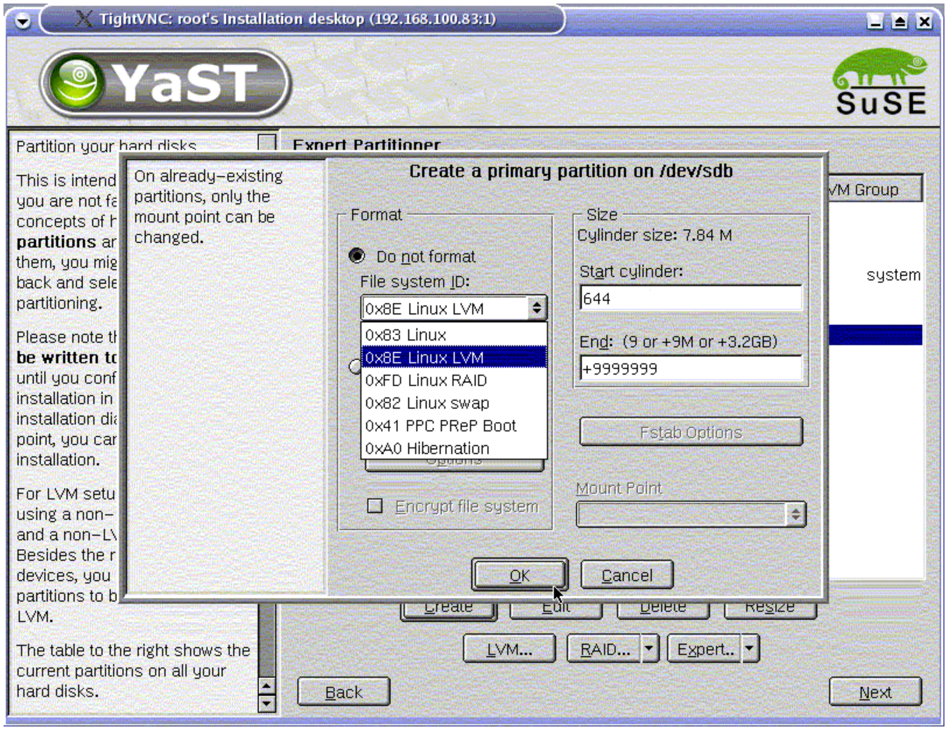
Start cylinder: 132

End: (9 or +9M or +3.2GB) +512

Mount Point: /

*Figure 2-17 Create a JFS root partition*

To use LVM, we need to create a volume group. We define an LVM partition that will occupy the rest of the disk. We proceed in the same way as for creating the root partition, except that we give it the type Linux LVM. This is shown in Figure 2-18 on page 34.



TightVNC: root's Installation desktop (192.168.100.83:1)

Partition your hard disks

This is intend
you are not fa
concepts of h
**partitions** ar
them, you mig
back and sele
partitioning.

Please note th
**be written to**
until you conf
installation in
installation dia
point, you car
installation.

For LVM setu
using a non-
and a non-LV
Besides the r
devices, you
partitions to b
LVM.

The table to the right shows the current partitions on all your hard disks.

Expert Partitioner

On already-existing partitions, only the mount point can be changed.

Create a primary partition on /dev/sdb

Format

● Do not format

File system ID: 0x8E Linux LVM

Size

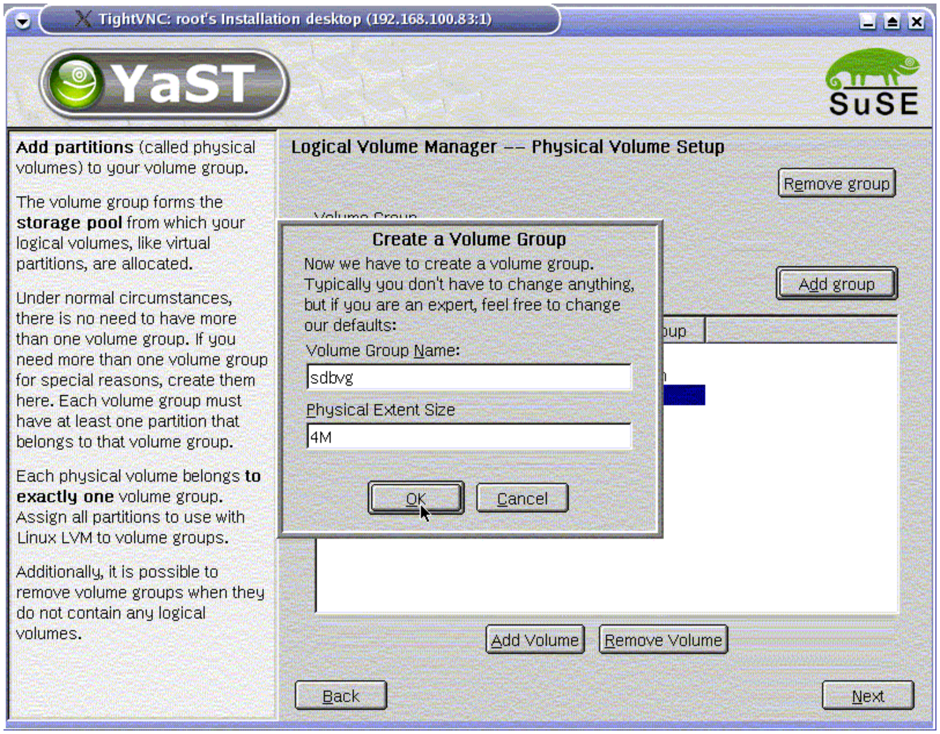
Cylinder size: 7.84 M

Start cylinder: 644

End: (9 or +9M or +3.2GB) +9999999

*Figure 2-18 Create a partition that fills up the disk*

This partition is then made into a volume group, and then logical volumes are defined on this volume group. Choose a name for the volume group and a physical extent size, as shown in Figure 2-19 on page 35.



**Add partitions** (called physical volumes) to your volume group.

The volume group forms the **storage pool** from which your logical volumes, like virtual partitions, are allocated.

Under normal circumstances, there is no need to have more than one volume group. If you need more than one volume group for special reasons, create them here. Each volume group must have at least one partition that belongs to that volume group.

Each physical volume belongs **to exactly one** volume group. Assign all partitions to use with Linux LVM to volume groups.

Additionally, it is possible to remove volume groups when they do not contain any logical volumes.

Logical Volume Manager -- Physical Volume Setup

Create a Volume Group

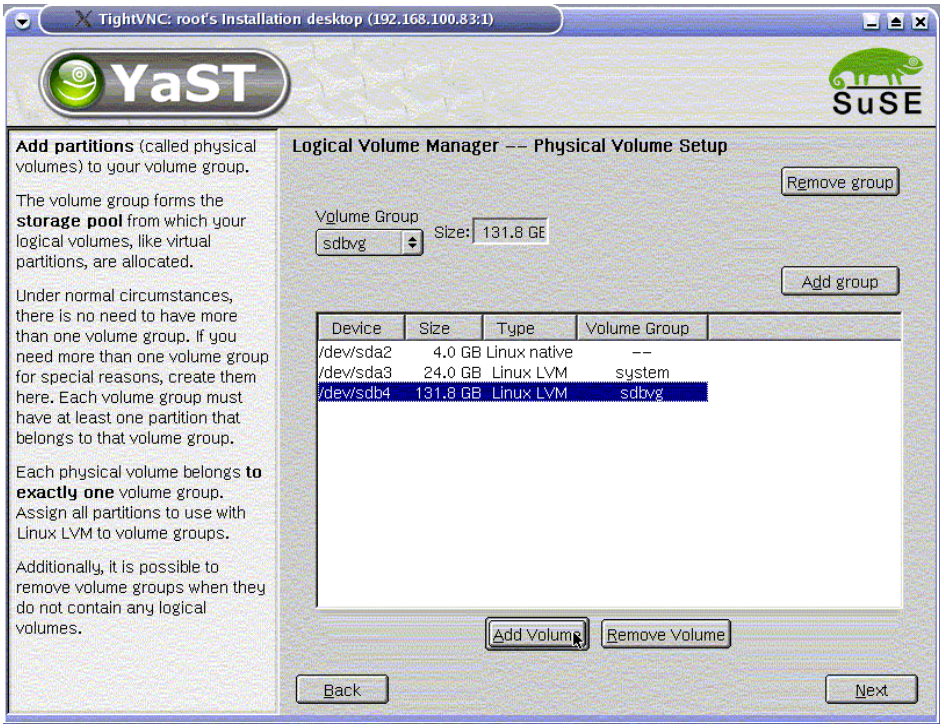
Now we have to create a volume group. Typically you don't have to change anything, but if you are an expert, feel free to change our defaults:

Volume Group Name: sdbvg

Physical Extent Size: 4M

*Figure 2-19 Volume group creation*

The new volume group is created using the /dev/sdb4 partition previously defined as an LVM partition. This is shown in Figure 2-20 on page 36.



**Add partitions** (called physical volumes) to your volume group.

The volume group forms the **storage pool** from which your logical volumes, like virtual partitions, are allocated.

Under normal circumstances, there is no need to have more than one volume group. If you need more than one volume group for special reasons, create them here. Each volume group must have at least one partition that belongs to that volume group.

Each physical volume belongs **to exactly one** volume group. Assign all partitions to use with Linux LVM to volume groups.

Additionally, it is possible to remove volume groups when they do not contain any logical volumes.

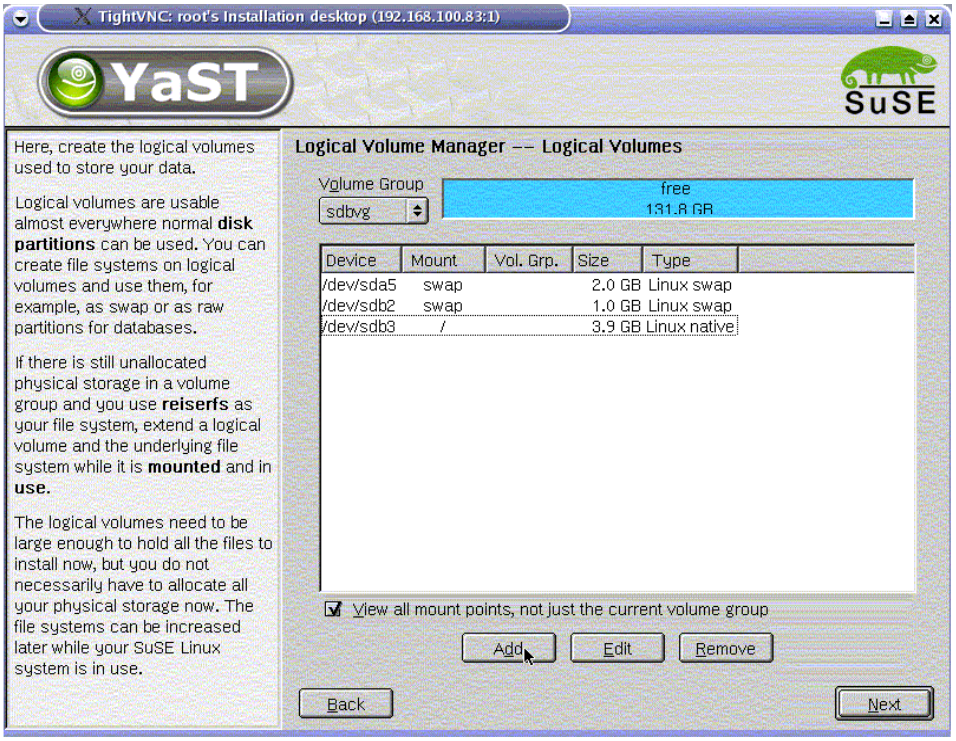
**Logical Volume Manager -- Physical Volume Setup**

Volume Group: sdbvg  Size: 131.8 GB

| Device | Size | Type | Volume Group |
|---|---|---|---|
| /dev/sda2 | 4.0 GB | Linux native | -- |
| /dev/sda3 | 24.0 GB | Linux LVM | system |
| /dev/sdb4 | 131.8 GB | Linux LVM | sdbvg |

*Figure 2-20 /dev/sda4 is added to the sdbvg volume group*

Our volume group is now ready, as shown in Figure 2-21 on page 37.



Here, create the logical volumes used to store your data.

Logical volumes are usable almost everywhere normal **disk partitions** can be used. You can create file systems on logical volumes and use them, for example, as swap or as raw partitions for databases.

If there is still unallocated physical storage in a volume group and you use **reiserfs** as your file system, extend a logical volume and the underlying file system while it is **mounted** and in **use.**

The logical volumes need to be large enough to hold all the files to install now, but you do not necessarily have to allocate all your physical storage now. The file systems can be increased later while your SuSE Linux system is in use.

**Logical Volume Manager -- Logical Volumes**

Volume Group: sdbvg

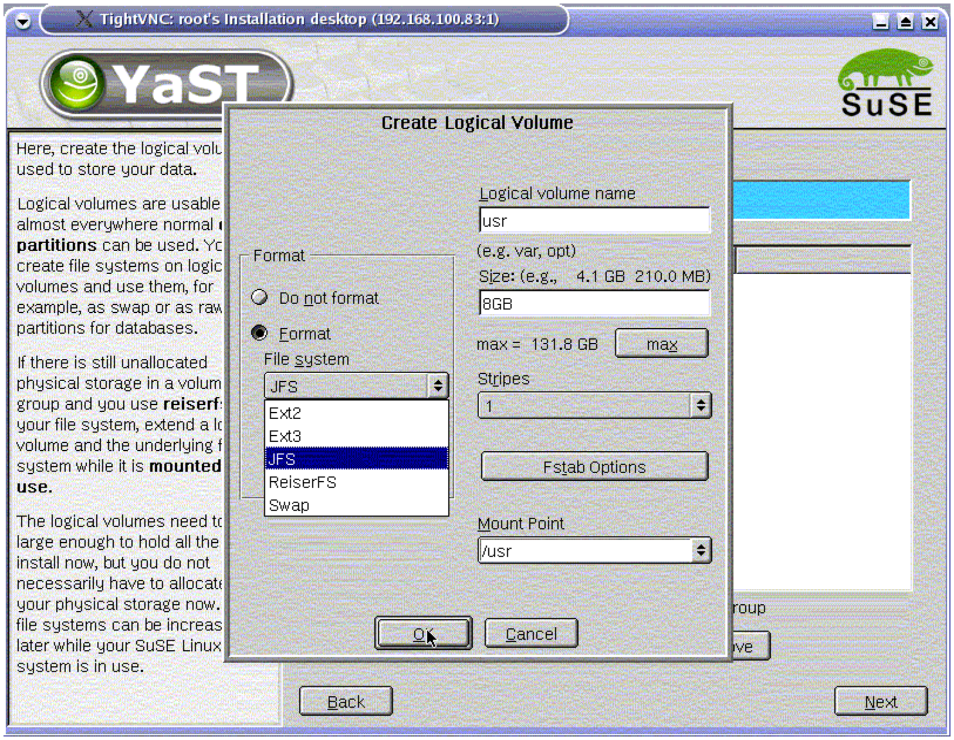
free 131.8 GB

| Device | Mount | Vol. Grp. | Size | Type |
|---|---|---|---|---|
| /dev/sda5 | swap | | 2.0 GB | Linux swap |
| /dev/sdb2 | swap | | 1.0 GB | Linux swap |
| /dev/sdb3 | / | | 3.9 GB | Linux native |

☑ View all mount points, not just the current volume group

*Figure 2-21 Volume group ready*

Next, we create logical partitions on which we mount /usr, /var, /opt, and so on. This is shown in Figure 2-22 on page 38.



Here, create the logical volu
used to store your data.

Logical volumes are usable
almost everywhere normal
**partitions** can be used. Yc
create file systems on logic
volumes and use them, for
example, as swap or as raw
partitions for databases.

If there is still unallocated
physical storage in a volum
group and you use **reiserf**
your file system, extend a lo
volume and the underlying f
system while it is **mounted**
**use.**

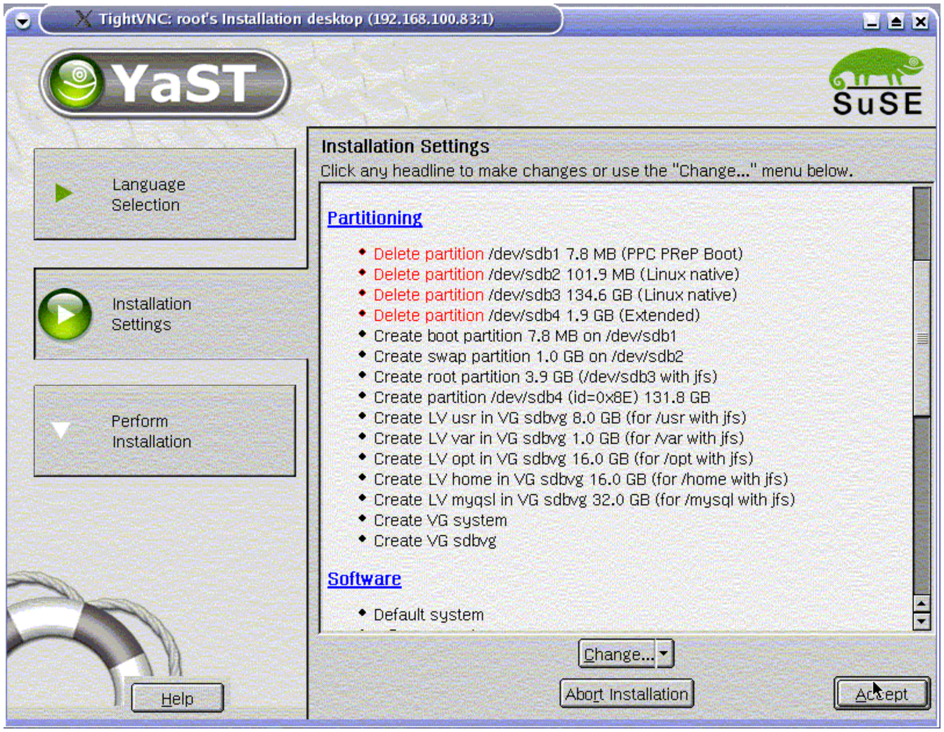
The logical volumes need to
large enough to hold all the
install now, but you do not
necessarily have to allocate
your physical storage now.
file systems can be increas
later while your SuSE Linux
system is in use.

Format
○ Do not format
● Format
File system: JFS

Logical volume name: usr
(e.g. var, opt)
Size: (e.g., 4.1 GB 210.0 MB): 8GB
max = 131.8 GB
Stripes: 1
Mount Point: /usr

*Figure 2-22 Logical volume creation (here /usr)*

Up to this point, nothing has been written to disk. Once your partitioning scheme is satisfactory, you need to commit the changes as shown in Figure 2-23 on page 39.



Installation Settings

Click any headline to make changes or use the "Change..." menu below.

Partitioning

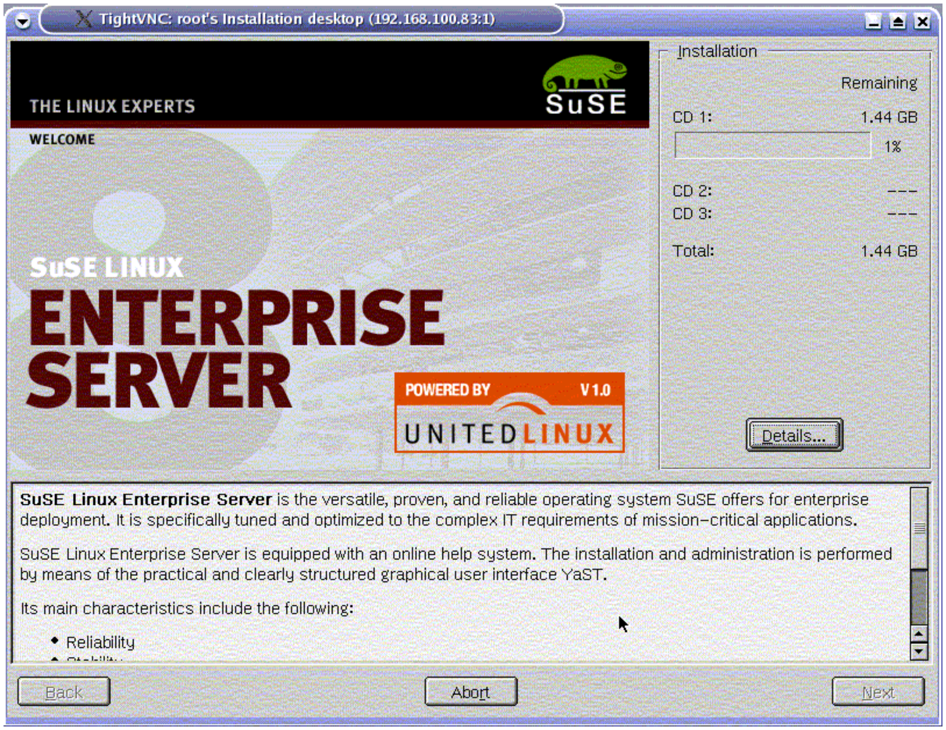
- Delete partition /dev/sdb1 7.8 MB (PPC PReP Boot)
- Delete partition /dev/sdb2 101.9 MB (Linux native)
- Delete partition /dev/sdb3 134.6 GB (Linux native)
- Delete partition /dev/sdb4 1.9 GB (Extended)
- Create boot partition 7.8 MB on /dev/sdb1
- Create swap partition 1.0 GB on /dev/sdb2
- Create root partition 3.9 GB (/dev/sdb3 with jfs)
- Create partition /dev/sdb4 (id=0x8E) 131.8 GB
- Create LV usr in VG sdbvg 8.0 GB (for /usr with jfs)
- Create LV var in VG sdbvg 1.0 GB (for /var with jfs)
- Create LV opt in VG sdbvg 16.0 GB (for /opt with jfs)
- Create LV home in VG sdbvg 16.0 GB (for /home with jfs)
- Create LV mysql in VG sdbvg 32.0 GB (for /mysql with jfs)
- Create VG system
- Create VG sdbvg

Software

- Default system

*Figure 2-23 Commit the new partitioning scheme*

If you decide not to change the default software selection, which is valid in most situations, the installation can start. The installer program formats the partitions and starts installing the software, as shown in Figure 2-24 on page 40.



Installation

| | Remaining |
|---|---|
| CD 1: | 1.44 GB |
| | 1% |
| CD 2: | --- |
| CD 3: | --- |
| Total: | 1.44 GB |

**SuSE Linux Enterprise Server** is the versatile, proven, and reliable operating system SuSE offers for enterprise deployment. It is specifically tuned and optimized to the complex IT requirements of mission–critical applications.

SuSE Linux Enterprise Server is equipped with an online help system. The installation and administration is performed by means of the practical and clearly structured graphical user interface YaST.

Its main characteristics include the following:

- Reliability

*Figure 2-24 Software installation started*

After all the selected packages are installed, you will be prompted to enter a password for the root userid. You are asked to configure the network interfaces. Once all the configuration data is entered, the system reboots once and you should be in business.

Refer to 2.5, “Post-installation tasks” on page 88 and Chapter 3, “System administration” on page 95 for information about how to further customize the system.

## 2.2.2 RHAS 3 installation

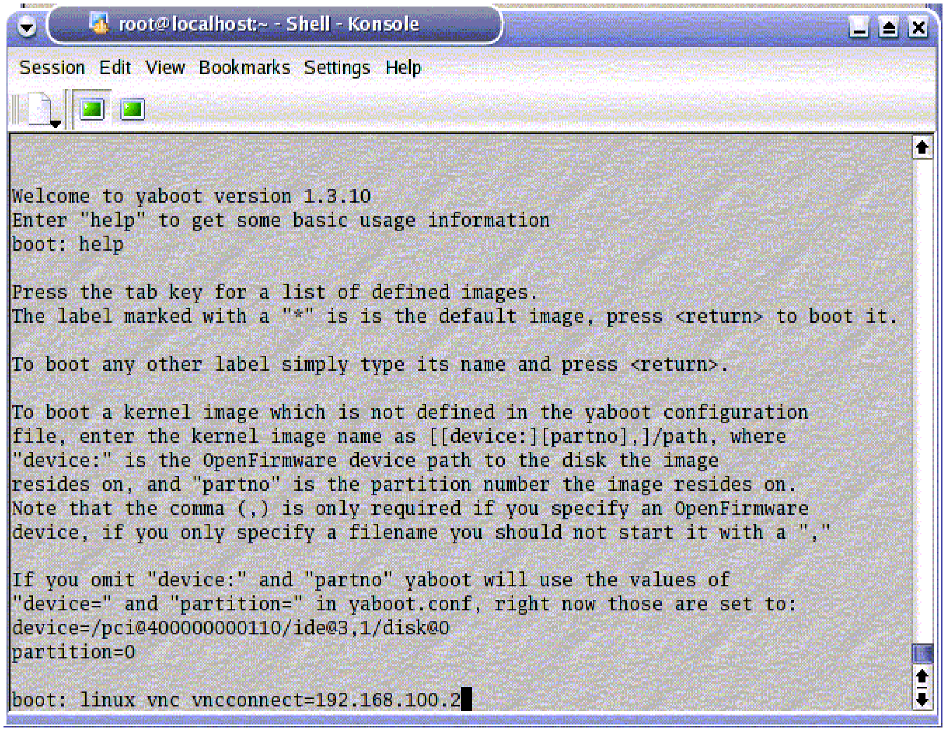
The installation of RHAS 3 follows the same steps as SLES 8. The VNC setup is a bit different if you decide to go for a VNC installation. The installer program requires that you give the name or IP address of the system from which you will manage the installation. On that system, you have to start the VNC client in listening mode as shown in Example 2-5 on page 41 for a UNIX system. The Windows version of the VNC client also has a listening mode. Here again we depict an installation with a VNC client.

*Example 2-5 Starting a VNC client in listening mode*

```
# vncviewer -listen
vncviewer -listen: Listening on port 5500 (flash port 5400)
vncviewer -listen: Command line errors are not reported until a connection
comes in.
```

In our configuration, the system has no graphics adapter and its first serial port is connected by a null modem cable to a Minicom client running on a Linux laptop.

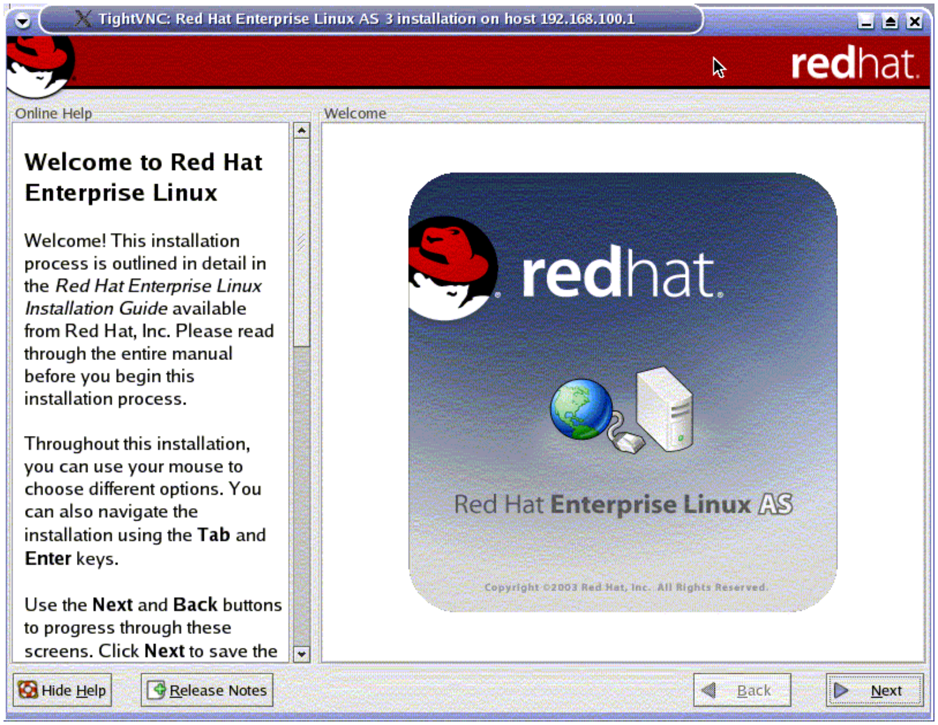
After booting the RHAS 3 CDROM, we receive the boot prompt in the Minicom window, as shown in Figure 2-25. This is where we specify the use of a VNC client.



```
Welcome to yaboot version 1.3.10
Enter "help" to get some basic usage information
boot: help

Press the tab key for a list of defined images.
The label marked with a "*" is is the default image, press <return> to boot it.

To boot any other label simply type its name and press <return>.

To boot a kernel image which is not defined in the yaboot configuration
file, enter the kernel image name as [[device:][partno],]/path, where
"device:" is the OpenFirmware device path to the disk the image
resides on, and "partno" is the partition number the image resides on.
Note that the comma (,) is only required if you specify an OpenFirmware
device, if you only specify a filename you should not start it with a ","

If you omit "device:" and "partno" yaboot will use the values of
"device=" and "partition=" in yaboot.conf, right now those are set to:
device=/pci@400000000110/ide@3,1/disk@0
partition=0

boot: linux vnc vncconnect=192.168.100.2
```

*Figure 2-25 boot: prompt options*

If you started the VNC client in listening mode, as described in Example 2-5, the first graphical screen from the Red Hat installer should pop up as shown in Figure 2-26 on page 42. The graphical installation program from Red Hat is called Anaconda.


TightVNC: Red Hat Enterprise Linux AS 3 installation on host 192.168.100.1
redhat.
Online Help
Welcome
redhat.
Red Hat Enterprise Linux AS
Copyright ©2003 Red Hat, Inc. All Rights Reserved.
Hide Help
Release Notes
Back
Next

**Welcome to Red Hat Enterprise Linux**

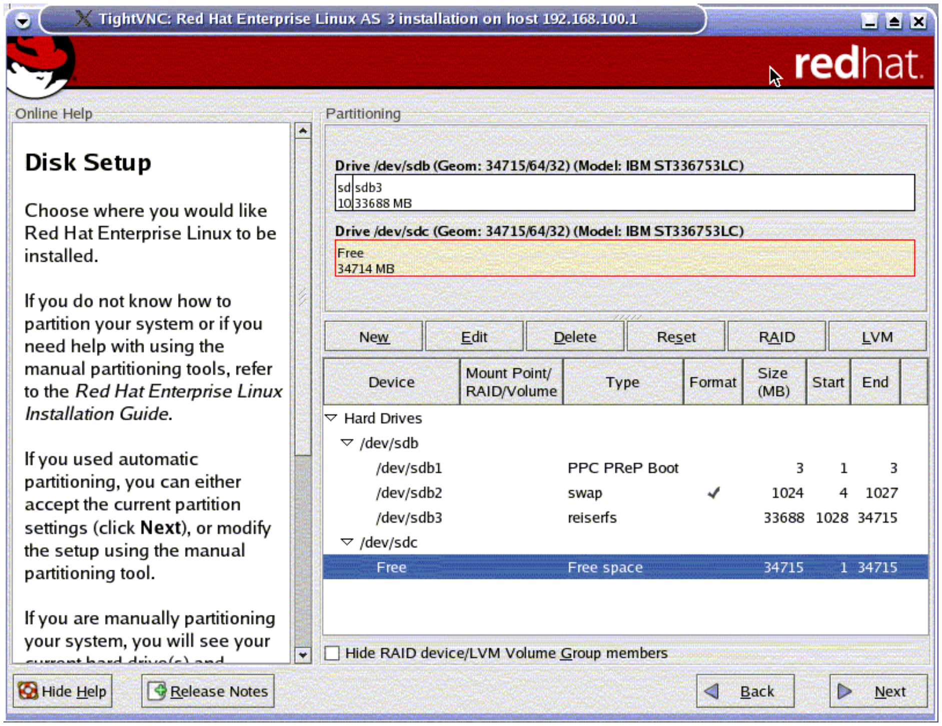
Welcome! This installation process is outlined in detail in the *Red Hat Enterprise Linux Installation Guide* available from Red Hat, Inc. Please read through the entire manual before you begin this installation process.

Throughout this installation, you can use your mouse to choose different options. You can also navigate the installation using the **Tab** and **Enter** keys.

Use the **Next** and **Back** buttons to progress through these screens. Click **Next** to save the

*Figure 2-26 Red Hat installer welcome display*

The installer takes you right to the disk partitioning section, as shown in Figure 2-27 on page 43. The tool is called DiskDruid.



**Disk Setup**

Choose where you would like Red Hat Enterprise Linux to be installed.

If you do not know how to partition your system or if you need help with using the manual partitioning tools, refer to the *Red Hat Enterprise Linux Installation Guide*.

If you used automatic partitioning, you can either accept the current partition settings (click **Next**), or modify the setup using the manual partitioning tool.

If you are manually partitioning your system, you will see your current hard drive(s) and

Drive /dev/sdb (Geom: 34715/64/32) (Model: IBM ST336753LC)
sd 10 | sdb3 33688 MB

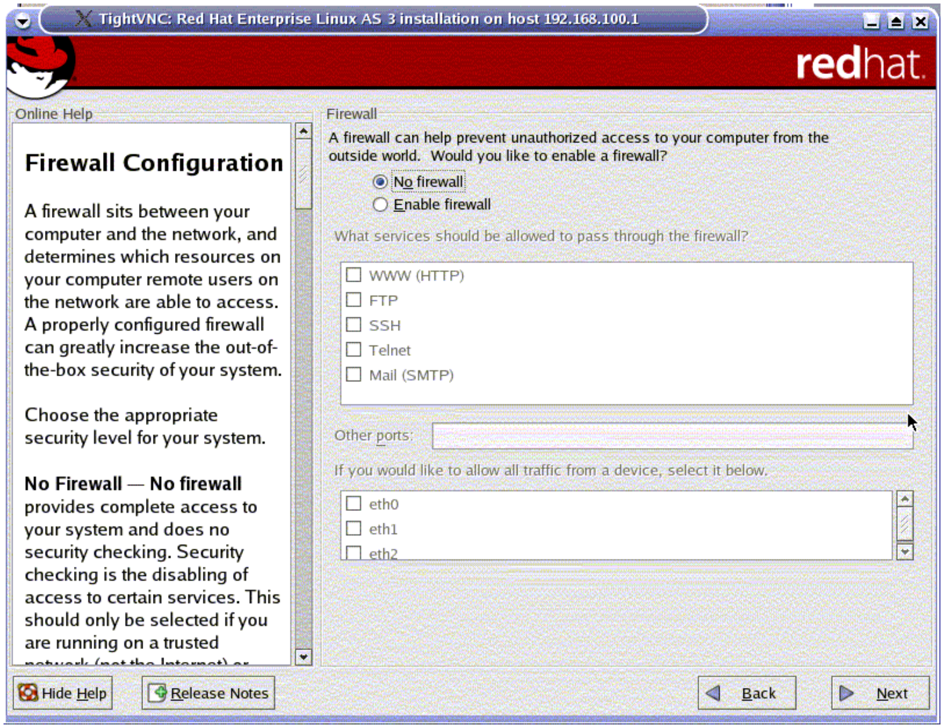
Drive /dev/sdc (Geom: 34715/64/32) (Model: IBM ST336753LC)
Free 34714 MB

| Device | Mount Point/ RAID/Volume | Type | Format | Size (MB) | Start | End |
|---|---|---|---|---|---|---|
| ▽ Hard Drives | | | | | | |
| ▽ /dev/sdb | | | | | | |
| /dev/sdb1 | | PPC PReP Boot | | 3 | 1 | 3 |
| /dev/sdb2 | | swap | ✓ | 1024 | 4 | 1027 |
| /dev/sdb3 | | reiserfs | | 33688 | 1028 | 34715 |
| ▽ /dev/sdc | | | | | | |
| Free | | Free space | | 34715 | 1 | 34715 |

□ Hide RAID device/LVM Volume Group members

*Figure 2-27 DiskDruid partitioning tool*

You can choose the initial level of security for your installation, as shown in Figure 2-28 on page 44.


TightVNC: Red Hat Enterprise Linux AS 3 installation on host 192.168.100.1
redhat
Online Help
Hide Help
Release Notes
Back
Next

Firewall Configuration

A firewall sits between your computer and the network, and determines which resources on your computer remote users on the network are able to access. A properly configured firewall can greatly increase the out-of-the-box security of your system.

Choose the appropriate security level for your system.

**No Firewall** — **No firewall** provides complete access to your system and does no security checking. Security checking is the disabling of access to certain services. This should only be selected if you are running on a trusted network (not the Internet) or

Firewall

A firewall can help prevent unauthorized access to your computer from the outside world. Would you like to enable a firewall?

● No firewall
○ Enable firewall

What services should be allowed to pass through the firewall?

□ WWW (HTTP)
□ FTP
□ SSH
□ Telnet
□ Mail (SMTP)

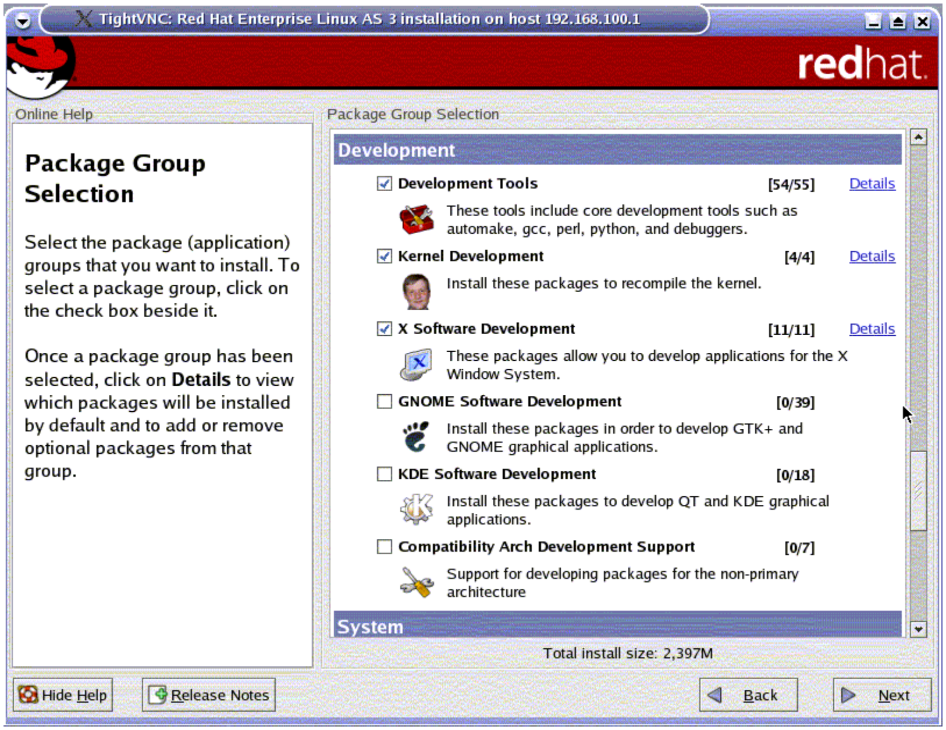
Other ports:

If you would like to allow all traffic from a device, select it below.

□ eth0
□ eth1
□ eth2

*Figure 2-28 Initial level of security*

Just as with SLES 8, you can also customize your software selection as illustrated in Figure 2-29 on page 45.


TightVNC: Red Hat Enterprise Linux AS 3 installation on host 192.168.100.1

Online Help

**Package Group Selection**

Select the package (application) groups that you want to install. To select a package group, click on the check box beside it.

Once a package group has been selected, click on **Details** to view which packages will be installed by default and to add or remove optional packages from that group.

Package Group Selection

**Development**

☑ **Development Tools** [54/55] Details
These tools include core development tools such as automake, gcc, perl, python, and debuggers.

☑ **Kernel Development** [4/4] Details
Install these packages to recompile the kernel.

☑ **X Software Development** [11/11] Details
These packages allow you to develop applications for the X Window System.

□ **GNOME Software Development** [0/39]
Install these packages in order to develop GTK+ and GNOME graphical applications.

□ **KDE Software Development** [0/18]
Install these packages to develop QT and KDE graphical applications.

□ **Compatibility Arch Development Support** [0/7]
Support for developing packages for the non-primary architecture

**System**

Total install size: 2,397M

*Figure 2-29 Software selection*

The installation proceeds with the actual installation of the packages selected, followed by a reboot.

## 2.2.3 Installation tips

Following are tips that we found useful during our CD-ROM installations.

### What happened to my disks

On a p650, the CD/DVD and the internal disks are on the same SCSI controller. They will therefore be assigned to a single partition.

Suppose you are installing from CD a first partition, called lpar1. You will add this SCSI controller to a profile and install the partition, using an internal disk.

Suppose you have two internal disks attached to this SCSI controller. They will be known to Linux as /dev/sda and /dev/sdb.

Now, if you need to install a second LPAR, lpar2, from CD later on, you will have to move this controller to the second LPAR to be able to use the CD. But because it is being used by lpar1, you must shut down lpar1 first and then move this SCSI controller to Linux to lpar2 so that you can boot and install from CD.

Suppose lpar2 has four disks attached. Upon booting lpar2, Linux will number the disks as follows:

- /dev/sda: first disk on the CD + disks SCSI controller; this is where lpar1 was installed, so do not touch it while installing lpar2
- /dev/sdb: second and last disk on the CD + disks SCSI controller
- /dev/sdc: first internal disk on lpar2; we will install lpar2 on this disk
- /dev/sdd: second internal disk on lpar2
- /dev/sde: third internal disk on lpar2
- /dev/sdf: fourth internal disk on lpar2

The second LPAR, lpar2, is installed on /dev/sdc. Then we shut down lpar2 to "give" back the SCSI controller with the CD and /dev/sda to lpar1. The lpar1 partition needs it, because Linux is installed in /dev/sda.

Now it is time to reboot lpar1, which finds its disk /dev/sda and comes up nicely. Now, boot up lpar2. Example 2-6 shows the result.

*Example 2-6 Where is my root disk?*

```
VFS: Cannot open root device "sdc3" or 08:43
Please append a correct "root=" boot option
Kernel panic: VFS: Unable to mount root fs on 08:43
 <0>Rebooting in 180 seconds..System Reset in kernel mode.
```

So what happened? When we installed lpar2, Linux had numbered the disks, starting with the disks which now belong to lpar1. Once lpar2 is rebooted with its own disks, then Linux starts numbering them this way:

- /dev/sda: first internal disk on lpar2; Linux is installed here
- /dev/sdb: second internal disk on lpar2
- /dev/sdc: third internal disk on lpar2
- /dev/sdd: fourth internal disk on lpar2

There are two places where the disk names need to match: in the bootloader configuration file /etc/lilo.conf, where we specify which partition of which disks owns the root file system, and in the mount table /etc/fstab, where we describe which partition is mounted where in the directory tree.

To recover from this situation, when lpar2 reboots, we specify at the boot prompt where the kernel should look for its root partition. This is shown in Example 2-7 on page 47.

*Example 2-7 Force the root partition to be /dev/sda3*

```
boot: linux root=/dev/sda3
```

The boot process should start and now comes the time to remount, in read-write mode, the root partition as described in the /etc/fstab file, still containing the reference to the “old’ root device: /dev/sdc3. Example 2-8 shows the result.

*Example 2-8 Not quite right yet*

```
Checking file systems...
fsck 1.28 (31-Aug-2002)

reiserfsck: could not open filesystem on "/dev/sdc3"


Warning... fsck.reiserfs for device /dev/sdc3 exited with signal 6.
fsck.reiserfs /dev/sdc3 failed (status 0x8). Run manually!          failed
Loading keymap qwerty/us.map.gz                                     done
Keyboard: kbdrate: Failed waiting for kbd controller!               failed
Loading console font lat1-16.psfu                                   failed
Loading screenmap none                                              done
Setting up console ttys                                             done

fsck failed.  Please repair manually and reboot. The root
file system is currently mounted read-only. To remount it
read-write do:

   bash# mount -n -o remount,rw /

Attention: Only CONTROL-D will reboot the system in this
maintenance mode. shutdown or reboot will not work.

Give root password to login:
```

After entering the root password, we issue the **mount** command, edit the files /etc/fstab and /etc/lilo.conf, then run lilo, exit, and reboot. This is depicted in Example 2-9.

*Example 2-9 Fix the /etc/lilo.conf and /etc/fstab*

```
(none):~ # mount -n -o remount,rw /
(none):~ # vi /etc/fstab
(none):~ # vi /etc/lilo.conf
(none):~ # lilo -v
running on chrp
install on /dev/sda
Installing /boot/yaboot.chrp onto /dev/sda1
```

```
519+1 records in
519+1 records out
check the image files
ERROR: image /boot/vmlinuz is not on bootdevice /dev/sda
`/tmp/ppc_lilo/yaboot.conf' -> `/etc/yaboot.conf'

(none):~ # exit
Unmounting file systems (ignore error messages)
/dev/sdc3 umounted
md: stopping all md devices.
flushing ide devices:
Restarting system.
```

2.3, “From network to fully unattended installation” on page 49 describes network installations that overcome completely these types of challenges.

### YaST2 in text mode can be fast

If you are installing rack-mounted LPARed systems, from CD-ROM or from the network, and if you cannot use a VNC install, you will end up opening terminal sessions on the HMC and interacting with the installer program from there. In text mode and through a serial connection, this is painfully slow.

Even worse, YaST2 in text mode is not intuitive and you very often have to use the <Tab> key to circulate through the options available on each screen in a sequential way. Pressing one <Tab> too many can result in having to start all over again. The trick here is use the <Esc> key together with the highlighted letter for each option, which takes you right to the desired action.

## 2.3 From network to fully unattended installation

Installing more than a few nodes from CD-ROM is not practical. The process of deploying SLES 8 or RHAS 3 on a large number of cluster nodes needs to be made faster and simpler, and you can accomplish this through network installation. There are many advantages to network installation:

- Speed

  With the advent of fast Ethernet networks (100Mbit/s, 1000Mbit/s), network installation can be much faster than CD or DVD installations.

- Efficiency

  With network installation, you can drive everything from the comfort of your office.

- Saving CDs

  There is no need to burn huge numbers of CDs (which are easily damaged).

- Troubleshooting

  It is useful to keep a few different kernels on an installation server, which can be used to boot remote clients.

- Automation

  Network installations are customizable and can be fully automated. Both SLES 8 and RHAS 3 provide mechanisms for this.

However, network installation requires additional setup time to configure and operate an installation server. It is also potentially more expensive, as you need to have a system at hand that plays the role of this installation server.

However, this system can be quite basic. Unlike AIX, which needs an AIX NIM install server, Linux can be network-installed from any UNIX or Linux system. We have installed Linux on LPARs from a laptop running Linux. The HMC itself has everything required to become a network installation server, but this is not supported.

CSM provides a very effective way of managing network installations. In the following sections, we describe what happens under the cover when CSM drives network installations. Also, even if you do not wish to use CSM, you may still wish to benefit from network installation.

## 2.3.1 Network installation fundamentals

The Linux Documentation Project[5] is an invaluable source of information for Linux. A few How-Tos relate to network boot and network installation. We recommend reading the following documentation:

- DHCP Howto[6]
- Network install HOWTO[7]
- Network boot HOWTO[8]
- KickStart HOWTO[9]
- Remote boot HOWTO[10]
- Remote-serial console HOWTO[11]

Under SLES 8, the HowTos are packaged in the howtoenh package and reside under the /usr/share/doc/howto directory.

Here we describe the steps needed to set up a network installation server to install SLES 8 or RHAS 3. Although we refer to “the” installation server, this function can be fulfilled jointly by three different systems. As we will see, we need a BOOTP/DHCP server, a TFTP server, and a NFS server to hold the packages to be installed.

This NFS server does not need to be the BOOTP/DHCP or the TFTP server, although this is common practice. If you install a large cluster (100+ nodes), it might be judicious to use a few NFS servers to share the load and speed up the overall installation process.

The network installation proceeds as follows:

- Upon remote IPL, the firmware issues a BOOTP request, sending along the MAC address of the installation adapter.
- A BOOTP/DHCP server responds with the IP address of the client node, the name of the boot file to use, and the IP address of the TFTP server for getting this boot file from.
- The firmware stores this information, downloads the boot file from the TFTP server, and starts it. The boot file is the installation kernel from the SLES 8 or RHAS 3 distribution. And although we describe the SLES 8 process here, the RHAS 3 process is similar.

[5] http://www.tldp.org/docs.html#howto
[6] Vladimir Vuksan <vuksan@veus.hr>
[7] Graham White <gwhite@uk.ibm.com>
[8] Brieuc Jeunhomme <bbp@via.ecp.fr>
[9] Martin Hamilton <martinh@gnu.org>
[10] Marc Vuilleumier Stuckelberg, David Clerc <David.Clerc@cui.unige.ch>
[11] Glen Turner <glen.tueer+howto@aarnet.edu.au>, Mark F. Komarinski <mkomarinski@wayga.org>

- Upon startup, this kernel will mount a mini-root file system in memory and start the /linuxrc script. This script is responsible for prompting the user for preliminary installation choices, such as load additional modules and choose the source for the packages to install (network, CD-ROM, disk) and the protocol to use (NFS, FTP, HTTP, TFTP).
- /linuxrc asks to load a kernel module for the network adapter that is used to install the packages. If you decide to use DHCP to configure the interface, the /linuxrc script also retrieves the directory corresponding to the root-path option that is set in /etc/dhcpd.conf for this node. This directory on the installation server is where we copied the CDs from the distribution.
- /linuxrc downloads a ramdisk from this directory (the /boot/root file, approximately 40 MB), mounts it, and starts YaST2 from there to perform the remaining part of the installation.
- From this point, the steps are identical to those in 2.2, “Step-by-step installation” on page 25.

## Enabling network boot on pSeries servers

A pSeries system can be instructed to boot from the network from the SMS menu. The menu for remote IPL is shown in Example 2-10.

*Example 2-10 Remote IPL menu*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
Main Menu
 1.  Select Language
 2.  Password Utilities NOT available in LPAR mode
 3.  View Error Log
 4.  Setup Remote IPL (Initial Program Load)
 5.  Change SCSI Settings
 6.  Select Console NOT available in LPAR mode
 7.  Select Boot Options






-------------------------------------------------------------------------------
Navigation Keys:

                                              X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key:4
```

Selecting 4 in the menu shown in Example 2-10 on page 51 gives the list of network interface cards available for netbooting (or remote IPL ,in pSeries terminology). This is a fast way to get the hardware addresses (MAC) of the adapters that you can use for network boot.

Make a note of the MAC addresses, as these will be needed to configure the BOOTP/DHCP server later. An example is given in Example 2-11.

*Example 2-11 Adapters available for network boot*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
NIC Adapters
           Device                            Slot          Hardware Address
 1.  10/100 Mbps Ethernet PCI Adapt      5:U0.1-P2-I5/E1         0002556f1fef
```

There are two other settings that we need to check for the adapter: the speed and duplex mode, and the spanning tree protocol. Select the adapter and you will be entering the network parameters for this network interface card as shown in Example 2-12.

*Example 2-12 Network parameters*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
Network Parameters
10/100 Mbps Ethernet PCI Adapter II: U0.1-P2-I5/E1
 1.  IP Parameters
 2.  Adapter Configuration
 3.  Ping Test








-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key:1
```

We selected option 2: Adapter Configuration, as shown in Example 2-12 on page 52. This gave us the screen shown in Example 2-13.

*Example 2-13 Adapter Configuration menu*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
Adapter Configuration
10/100 Mbps Ethernet PCI Adapter II: U0.1-P2-I5/E1
 1.  Speed,Duplex
 2.  Spanning Tree Enabled
 3.  Protocol




-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key:2
```

Set the speed and duplex as required. We chose 100 Mbit, full duplex in our case; see Example 2-14.

*Example 2-14 Speed and duplex settings*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
Speed,Duplex
10/100 Mbps Ethernet PCI Adapter II: U0.1-P2-I5/E1
 1.  auto,auto  ( rj45 )
 2.  10,half  ( rj45 )
 3.  10,full  ( rj45 )
 4.  100,half  ( rj45 )
 5.  100,full  ( rj45 )




-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key:5
```

Moving back to the Adapter Configuration menu, select the spanning tree option and disable it, as shown in Example 2-15. The spanning tree protocol is described at the following Web site:

```
http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cwsi2
/cwsiug2/vlan2/stpapp.htm
```

This is a link-level management protocol designed to avoid loops in Ethernet networks. With the Ethernet configurations that you are likely to use, this is not needed; if you do not disable it now, every network boot will take 60 seconds to build the spanning tree before actually going to the BOOTP/DHCP server.

*Example 2-15 Spanning tree currently enabled, so disable it*

```
Version RG030909_d65e03_sfw134
SMS 1.3 (c) Copyright IBM Corp. 2000,2003 All rights reserved.
-------------------------------------------------------------------------------
Spanning Tree Enabled
10/100 Mbps Ethernet PCI Adapter II: U0.1-P2-I5/E1
 1.  Yes  <===
 2.  No















-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key:2
```

## BOOTP/DHCP server

You need to define such a server on your network and it must be accessible either directly or through a gateway by the nodes you target for installation. If your installation server is a Linux server, you probably will end up defining a DHCP server. In the case of an AIX server, it is likely that you will create a BOOTP server.

Both work in approximately the same way, but their configurations differ. DHCP is backward-compatible with BOOTP, whereas pSeries systems emit BOOTP requests.

The BOOTP and DHCP protocols are described in:

http://www.faqs.org/rfcs/

### BOOTP configuration (for AIX)

Under AIX, the BOOTP configuration resides in /etc/bootptab. In our example, this file would contain a line like the one highlighted in Example 2-16.

*Example 2-16   Entry in the /etc/bootptab file*

```
# /etc/bootptab: database for bootp server (/usr/sbin/bootpd)
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#       first field -- hostname
#                       (may be full domain name and probably should be)
#
#      hd   -- home directory
#      bf   -- bootfile
#      sa   -- server IP address to tftp bootfile from
#      gw   -- gateways
#      ha   -- hardware address
#      ht   -- hardware type
#      ip   -- host IP address
#      sm   -- subnet mask
lpar8:hd=/tftpboot:bf=install:ip=192.168.100.84:ht=etherne:sa=192.168.100.110:g
w=192.168.100.110:sm=255.255.255.0:ha=0002556f1fef
```

What is important here is the hardware address (ha field). The hd and bf options are discussed in “The TFTP server” on page 57.

Remember that the bootp server is configured in the /etc/inetd.conf file. You should see a line similar to the one shown in Example 2-17.

*Example 2-17   bootp server entry in /etc/inetd.conf*

```
bootps  dgram udp wait root /usr/sbin/bootpd bootpd /etc/bootptab
```

The bootpd server can be started manually, outside of inetd, and in debug mode with the command shown in Example 2-18 on page 56. Make sure you remove bootp from the inetd configuration before running in standalone mode.

*Example 2-18 bootpd started manually*

```
# vi /etc/inetd.conf (to disable it in /etc/inetd.conf)
# refresh -s inetd (tell inetd to reread its config file)
0513-095 The request for subsystem refresh was completed successfully.
# /usr/sbin/bootpd -s -d -d -d -d -d /etc/bootptab
BOOTPD: bootptab mtime is Sat Nov  8 22:33:43 2003
BOOTPD: reading "/etc/bootptab"
BOOTPD: read 8 entries from "/etc/bootptab"
BOOTPD: dumped 8 entries to "/etc/bootpd.dump".
```

### DHCP configuration (for Linux)

DHCP has more features than BOOTP, and its configuration is slightly more complex. The configuration file for DHCP is /etc/dhcpd.conf. A description is given in Example 2-19.

*Example 2-19 Sample /etc/dhcpd.conf file*

```
always-reply-rfc1048 true;

#deny unknown-clients;
not authoritative;
ddns-update-style ad-hoc;
default-lease-time 60000;
max-lease-time 720000;

subnet 192.168.100.0 netmask 255.255.255.0 {
range 192.168.100.77 192.168.100.84;
option routers 192.168.100.60;

group {
   next-server 192.168.100.110;
   host lpar8 {
      fixed-address 192.168.100.84;
      hardware ethernet 00:02:55:6f:1f:ef;
      filename "install";
      option root-path “/install/sles”;
   }

}
}
```

The DHCP server in SLES 8 comes with the dhcp-server RPM. It is started on SLES 8 with the **rcdhcpd** command. The startup script has built-in syntax checking for the DHCP configuration file. The messages from the DHCP server are directed to the /var/log/messages file.

Sample output is given in Example 2-20.

*Example 2-20 DHCP server messages*

```
Nov  8 08:44:14 p630sles dhcpd: DHCPREQUEST for 192.168.100.84 from
00:02:55:6f:1f:ef via eth0
Nov  8 08:44:14 p630sles dhcpd: DHCPACK on 192.168.100.84 to 00:02:55:6f:1f:ef
via eth0
```

### The TFTP server

The next step is to set up a Trivial File Transfer Protocol (TFTP) server that will deliver the boot file to the requesting node. The default directory for serving TFTP files is /tftpboot.

For SLES 8, the boot file to copy to the /tftpboot directory is the install file at the root of the SLES 8 CD1. For RHAS 3, the file to be use is the netboot.img file found under images directory in the first CD of RHAS 3.

By default, the files served by the TFTP server need to be placed under the /tftpboot directory. This directory and the files underneath should belong to a regular user. We use the tftpd:tftpd user in SLES 8, and set the permissions to 440 for the boot file.

As reported[12] by Rolf Brudeseth at <rolfb@us.ibm.com>, you may experience problems during the TFTP download phase. As he suggests, the problems disappear when hardcoding the Address Resolution Protocol (ARP) address of the calling node on the installation server. This is done using the **arp** command, as shown in Example 2-21.

*Example 2-21 Forcing a static entry in the ARP table*

```
# arp -s lpar8 00:02:55:6f:1f:ef
# arp -a
lpar3 (192.168.100.79) at 00:02:55:3A:06:19 [ether] on eth0
lpar1 (192.168.100.77) at 00:02:55:3A:06:8C [ether] on eth0
hmc (192.168.100.109) at 00:02:55:E1:18:30 [ether] on eth0
lpar8 (192.168.100.84) at 00:02:55:6F:1F:EF [ether] PERM on eth0
lpar5 (192.168.100.81) at 00:02:55:3A:06:2C [ether] on eth0
bubu2 (192.168.100.60) at 00:02:55:E4:5A:F4 [ether] on eth0
lpar4 (192.168.100.80) at <incomplete> on eth0
lpar7 (192.168.100.83) at 00:02:55:3A:07:DB [ether] PERM on eth0
lpar6 (192.168.100.82) at 00:02:55:6F:1C:35 [ether] PERM on eth0
```

[12] See http://lists.debian.org/debian-powerpc/2002/debian-powerpc-200207/msg00858.html

ARP tables are usually of limited size. It is recommended that you remove the static entry from the ARP table, once the node has been installed.

Also, we learned that the bootfile in the /tftpboot directory should not be larger than 8 MBytes or it will not boot. This could be a firmware limitation that does not allow for enough memory to store and start the boot file.

On SLES 8, the tftp daemon is started from inetd. By default inetd is not enabled on a SLES 8 installation, so be sure to start inetd *before* attempting network installations.

To check the correct operation of the TFTP server, you can try to download files off the server from another location or even from the server itself. This is accomplished by using the **tftp** command, as shown in Example 2-22. If the TFTP server cannot start for some reason, the **get** command hangs.

*Example 2-22 Check the TFTP server*

```
root@p630sles:/root # tftp p630sles
tftp> bin
tftp> get install
Received 5194127 bytes in 0.4 seconds
tftp> quit
root@p630sles:/root # ls -l  /tftpboot/install
-r--r-----    1 tftpd    tftpd     5194127 Nov  2 15:47 /tftpboot/install
```

## The NFS server

Setting up an NFS server for installation is very easy. Prepare a directory on which to download the base level CDs of your distribution, copy over the contents of the CDs in the same tree, and then export the whole directory.

Make sure the permissions of the files and directories allow for anyone to read, and try mounting the exported directory on another system.

Here we describe the Linux way of setting up the NFS server. If using AIX, make the appropriate changes.

For SLES 8, we used the commands shown in Example 2-23 on page 59. We use downloaded ISO files of the SLES 8 distribution, and we use the Linux loop mount option.

Keep the update CDs from SLES 8 away from this directory; only the base level CDs should go there. We describe in 2.5.2, “Applying SLES 8 patches” on page 88, how to upgrade the system with patch CDs.

*Example 2-23 Creating the NFS source directory*

```
~# mkdir /install/sles
~# mkdir /mnt/sles8
~# for i in 1 2 3
do
mount -o loop -t iso9660 /tmp/SLES8-CD${i}.iso /mnt/sles8
cd /mnt/sles8
tar cf - . |(cd /install/sles;tar xf -)
cd
umount /mnt/sles8
done
```

We NFS export the directory by creating an entry in the /etc/exports file as shown in Example 2-24.

*Example 2-24 /etc/exports*

```
~# cat /etc/exports
/install        *(ro,root_squash,sync)
```

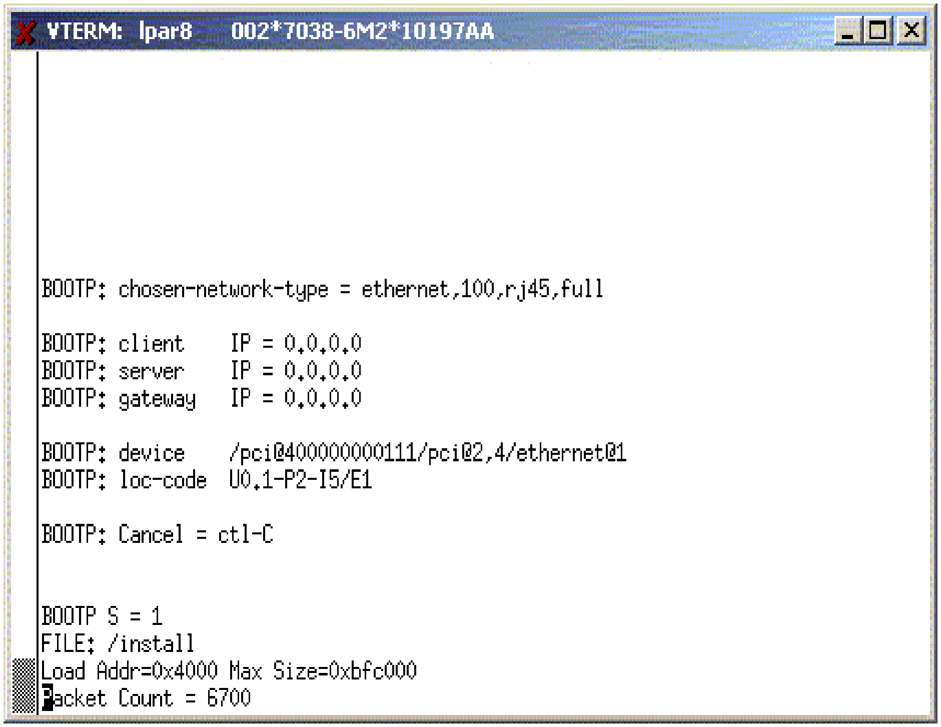
To start NFS on SLES 8, make sure the nfs-utils package is installed. The NFS server needs to be started with the nfsserver startup script under /etc/init.d. You may have to start the port mapper too, with the portmap startup script in the same directory. To check the correct operation of the NFS server, try to NFS-mount the target directory on another system, or at least on the server itself.

### 2.3.2 A practical example

We now perform our first network installation. We must first check that the DHCP server is running with the configuration file listed in Example 2-19 on page 56, and that the inetd superserver is started and configured to start the TFTP daemon upon request. We also have all the base level CDs dumped into a directory that is NFS-exported.

First the system is set to boot from network in SMS. We need to copy an install kernel from the distribution to the /tftpboot directory. The base distribution contains a kernel called install, but we will use the latest installation kernel at hand. We use the one that comes with the patch CD (SP3, in our case), and copy it to the /tftpboot directory. Why do we do that? Because it is very likely that the newer kernels will do a better job at detecting the hardware. And note that we only extract the kernel from the SP3 CD; we do not use the RPMs from SP3.

When the network boot starts, the terminal connection displays the BOOTP and TFTP progress as shown in Figure 2-30 on page 60.



```
BOOTP: chosen-network-type = ethernet,100,rj45,full

BOOTP: client    IP = 0.0.0.0
BOOTP: server    IP = 0.0.0.0
BOOTP: gateway   IP = 0.0.0.0

BOOTP: device    /pci@400000000111/pci@2,4/ethernet@1
BOOTP: loc-code  U0.1-P2-I5/E1

BOOTP: Cancel = ctl-C


BOOTP S = 1
FILE: /install
Load Addr=0x4000 Max Size=0xbfc000
Packet Count = 6700
```

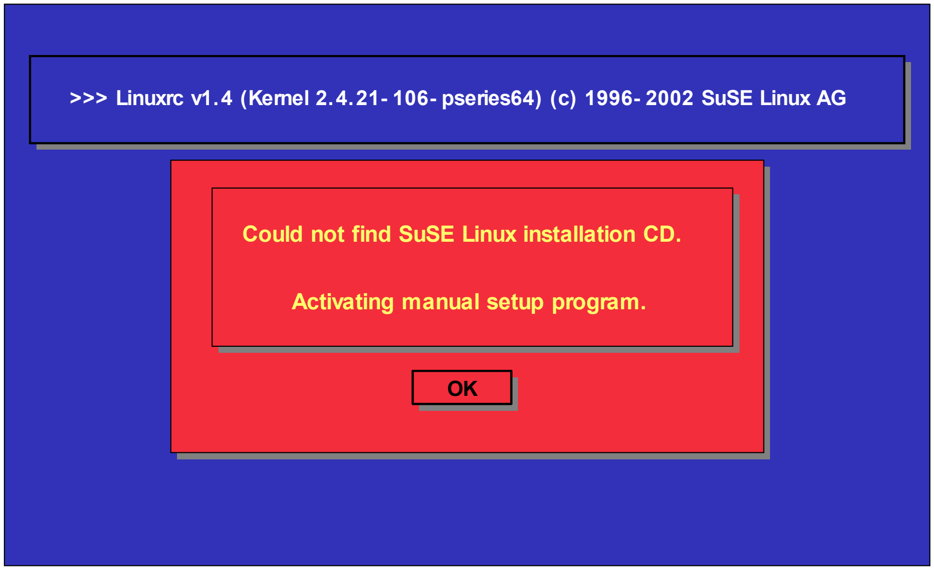
*Figure 2-30   BOOTP followed by the TFTP transfer of the boot file*

Meanwhile, on the installation server, the system log reports the BOOTP and the TFTP requests as shown in Example 2-25.

*Example 2-25   /var/log/messages*

```
Nov  8 22:12:32 p630sles dhcpd: BOOTREQUEST from 00:02:55:6f:1f:ef via eth0
Nov  8 22:12:32 p630sles dhcpd: BOOTREPLY for 192.168.100.84 to lpar8
(00:02:55:6f:1f:ef) via eth0
Nov  9 03:12:32 p630sles in.tftpd[29604]: RRQ from 192.168.100.84 filename
/install
```

Once the kernel transferred has been started, it passes control to the /linuxrc scripts to start interacting with the user. Figure 2-31 on page 61 shows the first screen.


>>> Linuxrc v1.4 (Kernel 2.4.21-106-pseries64) (c) 1996-2002 SuSE Linux AG

Could not find SuSE Linux installation CD.

Activating manual setup program.

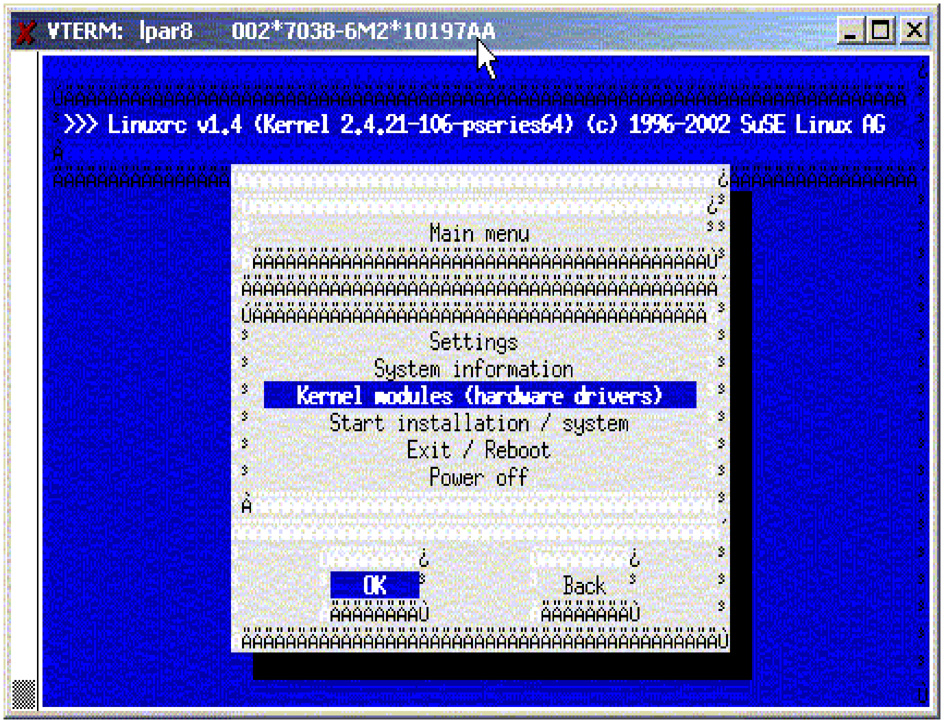
*Figure 2-31 linuxrc takes control*

So why can't linuxrc find the SLES 8 Installation CD? All three CDs have been downloaded in the /install/sles directory on the server and the dhcpd.conf file contains the root-path option to specify which root directory to use.

The problem is that the Open Firmware driving the BOOTP request does not know how to retrieve kernel parameters from DHCP, store them, retrieve the kernel from TFTP, and start it with the command line arguments saved previously. In our case, the minimum arguments to start the kernel would be the kernel module for the network adapter, the NFS server IP address, and the directory to mount from this NFS server.

When the installation kernel was started, it did not get this information, so linuxrc is prompting us to input the data. CSM solves this problem by entering the Open Firmware and passing the arguments directly to the kernel. Refer to "Passing the autoyast arguments to the kernel" on page 74 for an explanation of how to solve this problem in a different and very powerful way.

For the time being, we have to interact with linuxrc to get the installation going. Pressing OK in the screen shown in Figure 2-31 takes us back to the first linuxrc screen shown in Figure 2-32 on page 62. From this screen, we must first load the Ethernet kernel module. Select **Kernel modules (hardware drivers)** as shown, and then select **Load ppc_pseries64 modules**.

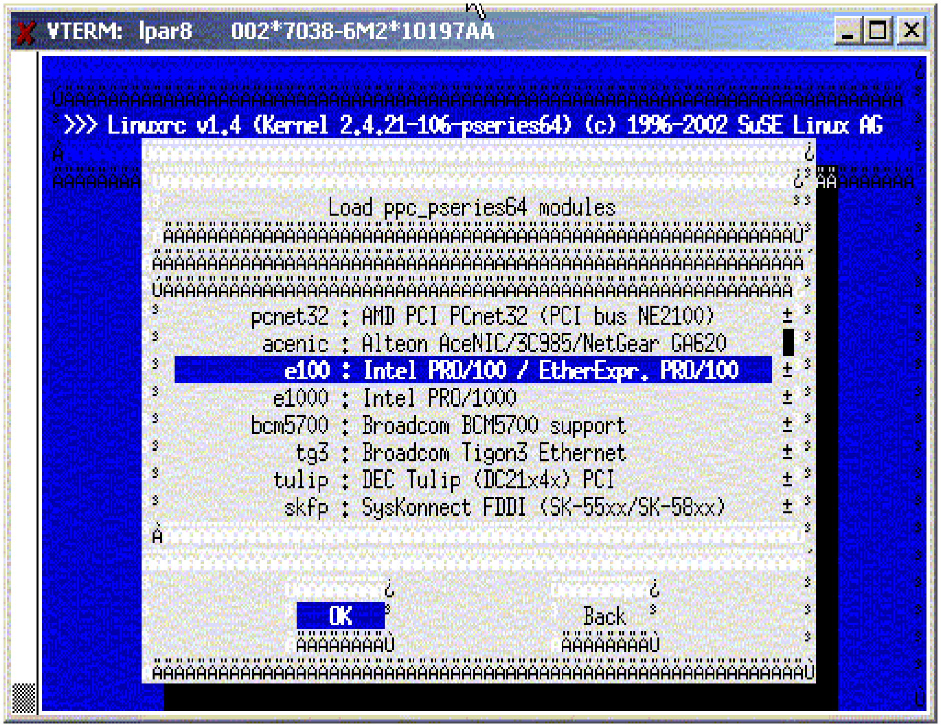


```
>>> Linuxrc v1.4 (Kernel 2.4.21-106-pseries64) (c) 1996-2002 SuSE Linux AG
```

VTERM: lpar8 002*7038-6M2*10197AA
Main menu
Settings
System information
Kernel modules (hardware drivers)
Start installation / system
Exit / Reboot
Power off
OK
Back

*Figure 2-32 Top level installation screen*

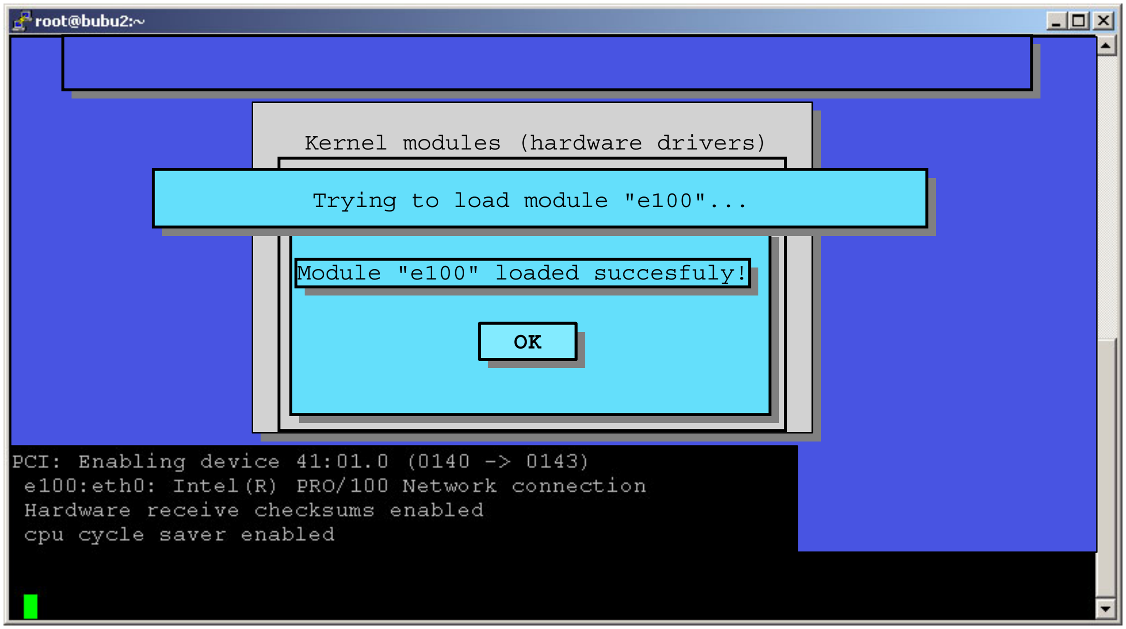
Choose the network module for your Ethernet adapter from the list shown in Figure 2-33 on page 63.



```
>>> Linuxrc v1.4 (Kernel 2.4.21-106-pseries64) (c) 1996-2002 SuSE Linux AG

            Load ppc_pseries64 modules

       pcnet32 : AMD PCI PCnet32 (PCI bus NE2100)
        acenic : Alteon AceNIC/3C985/NetGear GA620
          e100 : Intel PRO/100 / EtherExpr. PRO/100
         e1000 : Intel PRO/1000
       bcm5700 : Broadcom BCM5700 support
           tg3 : Broadcom Tigon3 Ethernet
         tulip : DEC Tulip (DC21x4x) PCI
          skfp : SysKonnect FDDI (SK-55xx/SK-58xx)

              OK                  Back
```

*Figure 2-33   List of available network modules*

The module is then loaded, as shown in Figure 2-34 on page 64.



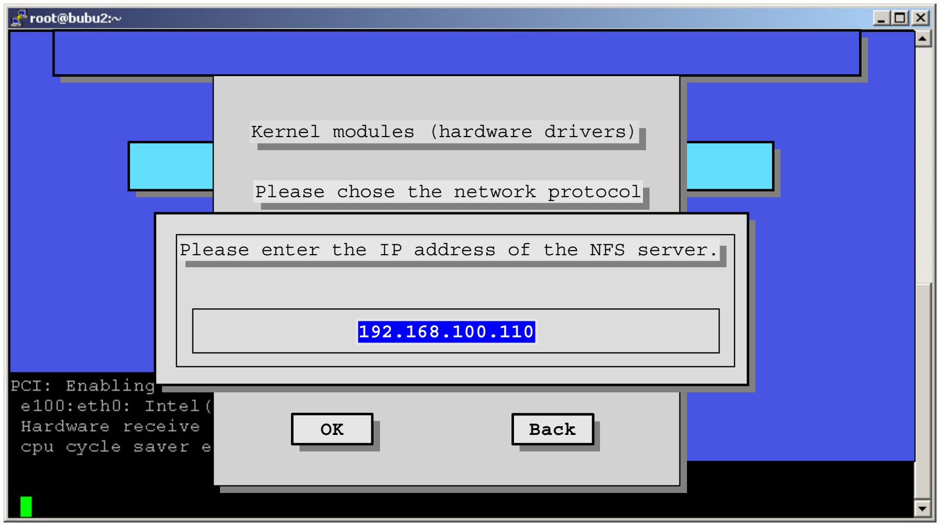
```
Kernel modules (hardware drivers)
Trying to load module "e100"...
Module "e100" loaded succesfuly!
```

```
PCI: Enabling device 41:01.0 (0140 -> 0143)
 e100:eth0: Intel(R) PRO/100 Network connection
 Hardware receive checksums enabled
 cpu cycle saver enabled
```

*Figure 2-34 Network module is loaded*

You have the choice for configuring the network adapter between fixed IP or DHCP. We used the DHCP method. linuxrc is cleverer than the Open Firmware, and it will retrieve not only the IP address but also the root-path option so that when we move on to configuring the source of the installation packages with NFS, linuxrc will fill in itself the fields with the information we entered in the /etc/dhcpd.conf file.

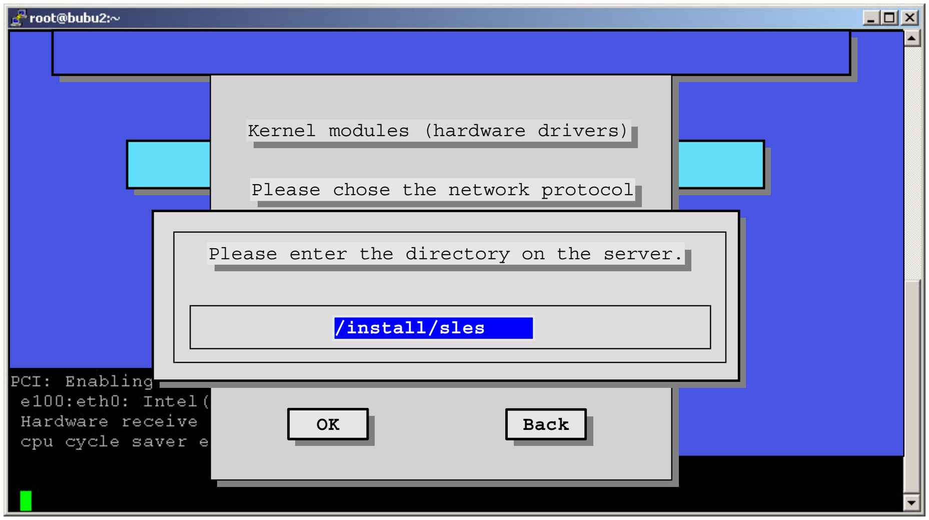
Back at the top level installation screen shown in Figure 2-32 on page 62, we can now select the **Start the installation/update** field. We select the network installation method with the NFS protocol, and we confirm the choice of the NFS server in Figure 2-35 on page 65.



```
Kernel modules (hardware drivers)

Please chose the network protocol

Please enter the IP address of the NFS server.

192.168.100.110

PCI: Enabling
 e100:eth0: Intel(
 Hardware receive
 cpu cycle saver e
```

*Figure 2-35 NFS server address*

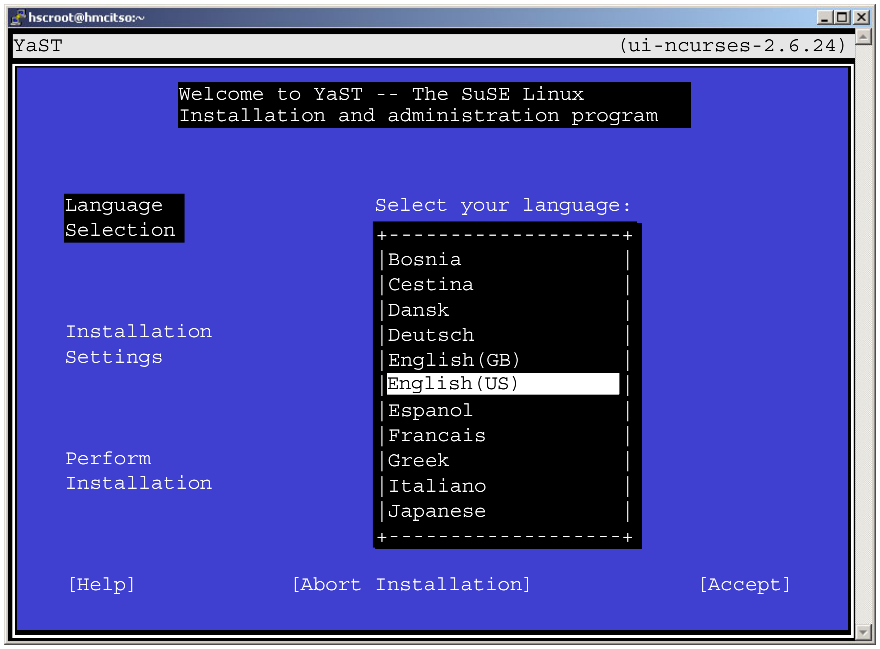
We confirm the name of the NFS server directory in Figure 2-36. This was filled at DHCP time, so we do not have to change it.



```
Kernel modules (hardware drivers)

Please chose the network protocol

Please enter the directory on the server.

/install/sles

PCI: Enabling
 e100:eth0: Intel(
 Hardware receive
 cpu cycle saver e
```

*Figure 2-36 NFS server directory*

After linuxrc mounts the root-path directory from the NFS server, it will load a large RAM disk (located in the boot/root file on the NFS server), switch to it, and start YaST2 off this new root directory, as shown in Figure 2-37.



```
YaST                                                    (ui-ncurses-2.6.24)

                Welcome to YaST -- The SuSE Linux
                Installation and administration program


    Language                      Select your language:
    Selection                     +-------------------+
                                  |Bosnia             |
                                  |Cestina            |
                                  |Dansk              |
    Installation                  |Deutsch            |
    Settings                      |English(GB)        |
                                  |English(US)        |
                                  |Espanol            |
                                  |Francais           |
    Perform                       |Greek              |
    Installation                  |Italiano           |
                                  |Japanese           |
                                  +-------------------+

    [Help]            [Abort Installation]             [Accept]
```

*Figure 2-37 YaST2 takes control*

From then on, the screens are identical to the ones we describe in 2.2.1, “SLES 8 installation” on page 26.

**Tips:**

- To speed up the network installation of the packages, close the terminal session. Writing to the console over the serial link slows down the installation.
- To monitor the network activity on the installation server, use the **`ifconfig`** command and watch the RX and TX fields.

Congratulations, at this point, you have performed you first network installation of SLES 8.

The process for RHAS 3 is very similar. The RHAS 3 does not use the information about the root-path option in /etc/dhcpd.conf, so you have to manually enter the IP address of the NFS server and the directory to mount.

## 2.3.3 Unattended installation

Now there are no more CDs to insert, because the installation proceeds using the network only. You can try different installer kernels very easily, since there is no need to burn bootable CDs; such is the power of network installation.

Still, we have to enter some information to drive the installation process, so the next step is to automate the network installation process: fill in the information one time, press Enter, and then let the systems install themselves.

There are basically two ways of doing this:

- By automating the key strokes
- By using autoyast2 for SLES 8 or kickstart for RHAS 3

In this document, we only discuss using autoyast2 and kickstart.

SuSE and RedHat each feature a specific tool to performed unattended network installation. Today, CSM for Linux on pSeries uses autoyast2; it will use kickstart when it supports RHAS 3.

With CSM, the user does not interact directly with the automatic installation tools. But CSM expects the user to provide a control file that is then modified, mainly to add the RSCT management part in the custom post-installation script.

In this section, we will use autoyast2 and kickstart directly to better understand how they work. This will be useful in helping you to troubleshoot any installation problems with CSM.

### Autoyast: the SuSE way

The reference document for autoyast2, written by Anas Nashif, is located at

http://www.suse.de/~nashif/autoinstall/8.1/autoyast2.pdf

The idea is to write a control file that describes all the choices for the installation. This file is given as an argument to the installer program, which no longer requires user input as this control file contains all the necessary information. Broadly speaking, the autoyast2 file can contain information about:

- General options like the language, the time zone or the type of mouse
- Reporting, the level of verbosity can be controlled this way
- Bootloader configuration
- Partitioning, including LVM
- Software package selection
- Network configuration
- Security settings
- Custom scripts

The custom scripts can be use to tailor the installation. CSM uses this feature extensively to incorporate the newly installed node in the CSM cluster, but you can do anything you want in these scripts. They can be written in shell or in perl script.

### *Control file format*

The autoyast2 control file is an XML file. This allows for easy parsing and syntax checking. You can use any editor to create the control file. Some syntactic editors (vim, emacs, kxmledit) can be used to check the proper structure of the file.

### *Creating a control file*

Each node to be installed can have its own control file. Example 2-26 shows an extract of a control file.

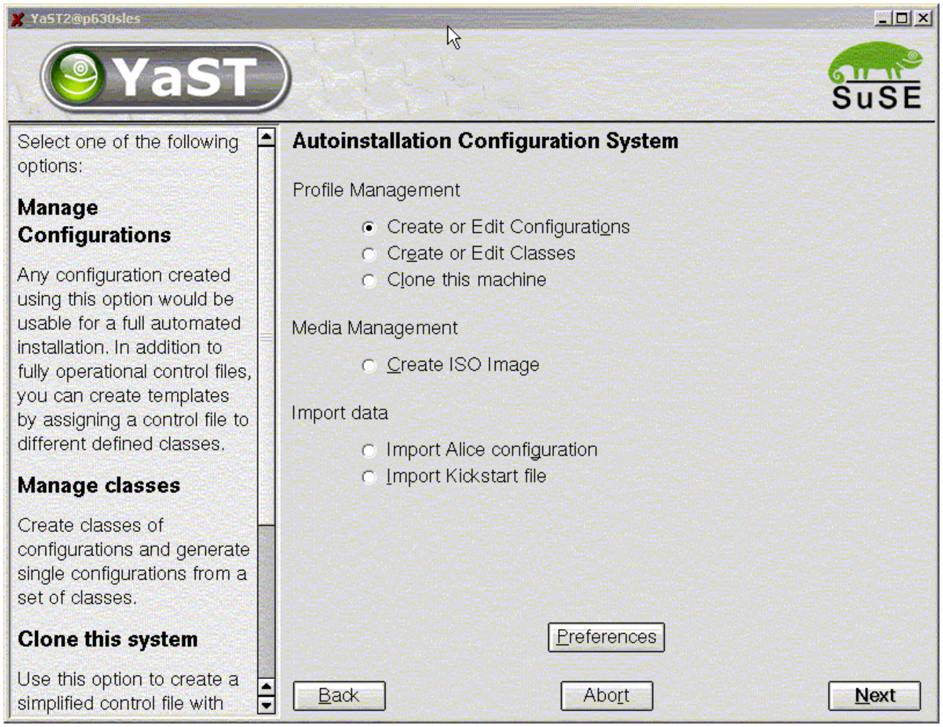
*Example 2-26 Excerpt of an autoyast2 control file*

```
<?xml version="1.0"?>
<!DOCTYPE profile SYSTEM "/usr/share/YaST2/include/autoinstall/profile.dtd">
<profile xmlns="http://www.suse.com/1.0/yast2ns"
xmlns:config="http://www.suse.com/1.0/configns">
  <configure>
    <inetd>
      <inetd_services config:type="list">
        <inetd_service>
          <service_name>ftp</service_name>
          <service_status>enable</service_status>
        </inetd_service>
        <inetd_service>
          <service_name>telnet</service_name>
          <service_status>enable</service_status>
        </inetd_service>
      </inetd_services>
      <start_inetd config:type="boolean">true</start_inetd>
    </inetd>
```

To create a control file, you can either write it “from scratch” with an editor, or use theYaST2 administration tool:

```
# yast2 autoyast
```

This can be done on any system running SLES 8. The initial screen for autoyast is shown in Figure 2-38 on page 69.



Select one of the following options:

**Manage Configurations**

Any configuration created using this option would be usable for a full automated installation. In addition to fully operational control files, you can create templates by assigning a control file to different defined classes.

**Manage classes**

Create classes of configurations and generate single configurations from a set of classes.

**Clone this system**

Use this option to create a simplified control file with

**Autoinstallation Configuration System**

Profile Management

- ● Create or Edit Configurations
- ○ Create or Edit Classes
- ○ Clone this machine

Media Management

- ○ Create ISO Image

Import data

- ○ Import Alice configuration
- ○ Import Kickstart file

*Figure 2-38 Autoyast configuration*

The cloning option is useful to quickly create a starting point. This option extracts the information from the running system and creates the corresponding autoyast2 file. If you have many identical systems, you could install one system manually, run **`yast2 autoyast`** on this node to create a control file, and then use this control file to install the remaining nodes.

If you choose the first option, as shown in Example 2-38, you are guided to enter your choices manually. The “Classes” option can be used to organize your control files by families, for example using a control file for your “Web servers” and another one for “interactive nodes”.

It is recommended that you start with a simple control file that works, and then improve it. The file shown in Example 2-27 on page 70 is a very basic control file that installs a minimal system on the first SCSI disk using an Ethernet adapter configured with DHCP. The root password and the admin password are both set to itsoadmin and are encrypted.

A very simplistic customization script is embedded in the file. It merely creates a file in /etc/profile.d directory to set the path to access the IBM XL Fortran compiler. In real situations, you would add more useful commands here, such as install additional software, create userids, set up LDAP, or whatever.

> **Important:** If you create your own autoyast2 control file, be sure to set the “confirm config” flag to false in the <install><general><mode> section. This is not the default.
>
> Without this setting, the installation will not run fully automated.

*Example 2-27 Sample autoyast2 control file*

```
<?xml version="1.0"?>
<!DOCTYPE profile SYSTEM "/usr/share/YaST2/include/autoinstall/profile.dtd">
<profile xmlns="http://www.suse.com/1.0/yast2ns"
xmlns:config="http://www.suse.com/1.0/configns">
  <configure>
    <networking>
      <dns>
        <dhcp_hostname config:type="boolean">false</dhcp_hostname>
        <dhcp_resolv config:type="boolean">false</dhcp_resolv>
        <domain>local</domain>
        <hostname>lpar8</hostname>
      </dns>
      <interfaces config:type="list">
        <interface>
          <bootproto>dhcp</bootproto>
          <device>eth0</device>
          <module>e100</module>
          <startmode>onboot</startmode>
          <wireless>no</wireless>
        </interface>
      </interfaces>
      <routing>
        <ip_forward config:type="boolean">false</ip_forward>
        <routes config:type="list">
          <route>
            <destination>default</destination>
            <device>-</device>
            <gateway>192.168.100.60</gateway>
            <netmask>-</netmask>
          </route>
        </routes>
      </routing>
    </networking>
    <scripts>
      <post-scripts config:type="list">
        <script>
          <filename>custom</filename>
          <interpreter>shell</interpreter>
          <source>
<![CDATA[# Very basic example of a custom script
echo "export PATH=$PATH:/opt/ibmcmp/xlf/8.1/bin" > /etc/profile.d/xlf.sh
]]>
          </source>
        </script>
      </post-scripts>
</scripts>
    <users config:type="list">
      <user>
        <encrypted config:type="boolean">true</encrypted>
        <user_password>hRoWvYp76SFiU</user_password>
        <username>root</username>
      </user>
      <user>
        <encrypted config:type="boolean">true</encrypted>
        <user_password>3VBnRzl.nZuO6</user_password>
        <username>admin</username>
      </user>
    </users>
  </configure>
  <install>
    <general>
      <clock>
        <timezone>US/Eastern</timezone>
      </clock>
      <keyboard>
        <keymap>english-us</keymap>
      </keyboard>
      <language>en_US</language>
      <mode>
        <confirm config:type="boolean">false</confirm>
        <forceboot config:type="boolean">false</forceboot>
        <interactive_boot config:type="boolean">false</interactive_boot>
        <reboot config:type="boolean">false</reboot>
      </mode>
      <mouse>
        <id>00_ps2</id>
      </mouse>
    </general>
    <partitioning config:type="list">
      <drive>
        <device>/dev/sda</device>
        <initialize config:type="boolean">false</initialize>
        <partitions config:type="list">
          <partition>
            <crypt_fs config:type="boolean">false</crypt_fs>
            <format config:type="boolean">false</format>
            <partition_id config:type="integer">65</partition_id>
            <partition_type>primary</partition_type>
            <size>4MB</size>
</partition>
          <partition>
            <crypt_fs config:type="boolean">false</crypt_fs>
            <crypt_key></crypt_key>
            <format config:type="boolean">false</format>
            <mount></mount>
            <partition_id config:type="integer">130</partition_id>
            <partition_type>primary</partition_type>
            <size>1GB</size>
          </partition>
          <partition>
            <crypt_fs config:type="boolean">false</crypt_fs>
            <crypt_key></crypt_key>
            <filesystem config:type="symbol">reiser</filesystem>
            <format config:type="boolean">true</format>
            <mount>/</mount>
            <partition_id config:type="integer">131</partition_id>
            <partition_type>primary</partition_type>
            <size>512MB</size>
          </partition>
          <partition>
            <crypt_fs config:type="boolean">false</crypt_fs>
            <crypt_key></crypt_key>
            <format config:type="boolean">false</format>
            <mount></mount>
            <partition_id config:type="integer">142</partition_id>
            <lvm_group>system</lvm_group>
            <partition_type>primary</partition_type>
            <size>24GB</size>
          </partition>
        </partitions>
        <use>all</use>
      </drive>
    </partitioning>
    <lvm config:type="list">
     <lvm_group>
      <lvm_name>system</lvm_name>
      <pesize>4M</pesize>
      <logical_volumes config:type="list">
        <lv>
              <lv_name>usrlv</lv_name>
              <lv_size>8GB</lv_size>
              <lv_fs>reiser</lv_fs>
<lv_mount>/usr</lv_mount>
        </lv>
        <lv>
```

```
            <lv_name>varlv</lv_name>
            <lv_size>512MB</lv_size>
            <lv_fs>reiser</lv_fs>
            <lv_mount>/var</lv_mount>
        </lv>
        <lv>
            <lv_name>homelv</lv_name>
            <lv_size>4GB</lv_size>
            <lv_fs>reiser</lv_fs>
            <lv_mount>/home</lv_mount>
        </lv>
        <lv>
            <lv_name>optlv</lv_name>
            <lv_size>1GB</lv_size>
            <lv_fs>reiser</lv_fs>
            <lv_mount>/opt</lv_mount>
        </lv>
      </logical_volumes>
     </lvm_group>
    </lvm>
    <software>
      <base>Minimal</base>
    </software>
  </install>
</profile>
```

At the time of writing, it is not possible to create LVM partitions with `yast2 autoyast`, and the control file must be edited manually to define the volume groups and the logical volumes. What we did is shown in Example 2-27 on page 70 in the <lvm></lvm> section.

Also, if you run the same installation two times, the second installation complains that the volume group you are trying to create already exists and the installation prompts you to remove the previous volume group.

To avoid this situation, Tobias Mucke[13] proposes using the script shown in Example 2-28 on page 74, to be inserted in the <scripts> section of the control file. Adjust the line giving the list of volume groups that could preexist on the system before installation starts.

Use the script shown in Figure 2-28 on page 74.

[13] Tobias_Mucke@mn.man.de http://lists.suse.com/archive/suse-autoinstall/2003-Jun/0069.html

*Example 2-28 Removing preexisting volume groups*

```
<pre-scripts config:type="list">
       <script>
        <filename>/vgremove.sh</filename>
        <interpreter>shell</interpreter>
        <source>
<![CDATA[
insmod lvm-mod
vgscan
# Adjust the list of VGs to be removed
vgs=`echo /dev/system`
for vg in $vgs
do
  if [ -d $vg ]
  then
   vgchange -a y $vg
   lvs=`ls $vg`
   for lv in $lvs
   do
     i=`basename $lv`
     if [ $i != "group" ]
     then
      lvremove -f $vg/$i
     fi
   done
   vgchange -a n $vg
   vgremove $vg
fi
done
]]>
       </source>
      </script>
     </pre-scripts>
```

### *Passing the autoyast arguments to the kernel*

To start the autoyast2 network installation, the installation kernel retrieved from the /tftpboot directory must be given a few arguments which describe the name of the control file and its location.

As discussed in 2.3.2, “A practical example” on page 59, TFTP cannot be used to pass arguments to the kernel; instead, we have to use the Open Firmware boot command. Open Firmware is described in greater detail in 2.4.3, “Open Firmware basics” on page 80 but at this point, we will use it as described in Example 2-29 on page 75.

To start the unattended installation, we set the system to boot to Open Firmware, start it, and open a terminal window. When we get to the Open Firmware prompt, we enter the boot command shown in Example 2-29, all in one single line.

This command tells the firmware to load a kernel from the network and start it with the command line starting at “`insmod=e100`”. This command line gives enough information to the kernel, to linuxrc in fact, to start the unattended installation. We specify the installation server and directory, as well as the location of the autoyast2 control file. From then on, the installation proceeds without user interaction.

*Example 2-29 Open Firmware boot command*

```
          1 = SMS Menu                            5 = Default Boot List
          6 = Stored Boot List                    8 = Open Firmware Prompt


     memory      keyboard     network     scsi     speaker  ok
0 > boot net insmod=e100 insmod=lvm-mod instmode=nfs
install=nfs://192.168.100.110/install/sles
autoyast=nfs://192.168.100.110/install/sles/trylvm.xml
```

**Tip:** SLES 8 SP3 comes with a very interesting tool that can “burn” command line arguments directly into a SuSE kernel, using a built-in static 512 characters. This is described in the ppc/netboot/ directory of the SP3 CD. To create a customized kernel for the unattended installation, copy the install kernel into the /tftpboot directory and issue:

```
/tftpboot:# ...../sp3/ppc/netboot/mkzimage_cmdline -a 1 -c\
-s “insmod=e100 insmod=lvm-mod instmode=nfs \
install=nfs://192.168.100.110/install/sles \
autoyast=nfs://192.168.100.110/instal/sles/trylvm.xml” install
```

This is very handy, because we do not have to go to Open Firmware any more. Just boot through TFTP and the kernel will start linuxrc with the right arguments. To recover a “neutral” kernel, simply run:

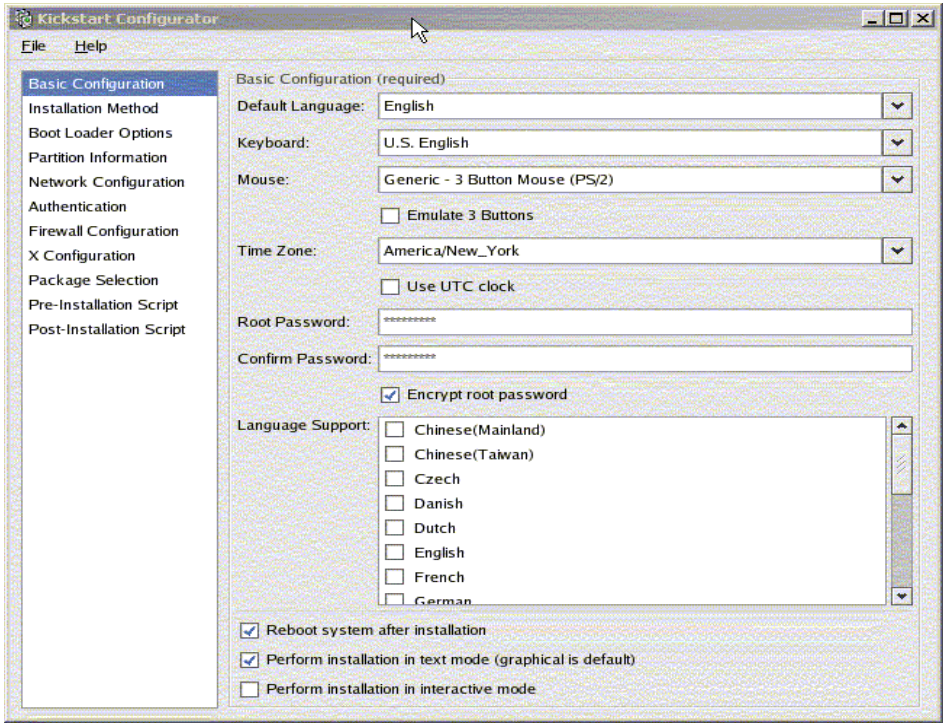
```
/tftpboot:# mkzimage_cmdline -a 0 -c install
```

This is a far-reaching utility, as it can be used to pass any kind of argument to the kernel. We also use it in “Disk setup” on page 364 to circumvent SCSI ID conflicts in HA environments. At the time of writing, RHAS 3 does not provide an equivalent utility.

## Kickstart: the Red Hat way

Red Hat's kickstart offers functionally similar to autoyast2. Here, the control file has no particular structure; it is simply a flat text file with directives. The easiest way to create a kickstart template file is to use redhat-config-kickstart. But although this is easy to use, it is not quite complete for kickstarting PowerPC systems; the control file, whose name defaults to ks.cfg, has to be manually edited.

The main redhat-config-kickstart window is shown in Figure 2-39. It cannot run in text mode, but you can run it on any recent Red Hat system.



Basic Configuration (required)

Default Language: English

Keyboard: U.S. English

Mouse: Generic - 3 Button Mouse (PS/2)

□ Emulate 3 Buttons

Time Zone: America/New_York

□ Use UTC clock

Root Password: **********

Confirm Password: **********

☑ Encrypt root password

Language Support:
□ Chinese(Mainland)
□ Chinese(Taiwan)
□ Czech
□ Danish
□ Dutch
□ English
□ French
□ German

☑ Reboot system after installation

☑ Perform installation in text mode (graphical is default)

□ Perform installation in interactive mode

*Figure 2-39 redhat-config-kickstart*

The control file generated this way does not work as is. The partition table that would be generated would not contain a PReP boot partition, which is needed, and an error is generated when configuring X windows. In fact, the installer reports an error when trying to skip the X configuration even though it detected no graphical device.

Fortunately, if you manually install an RHAS 3 system, a sample kickstart configuration file for the system that has just been installed is given in

/root/anaconda-ks.cfg. Using this file and the one generated by the redhat-config-kickstart command, plus advice from Jeremy Katz[14] for the partitioning problems, we put together a ks.cfg file that would work for us. It is listed in Example 2-30.

*Example 2-30 Sample ks.cfg*

```
#Generated by Kickstart Configurator
# And a good deal of hand editing too!!

#System  language
lang en_US
#Language modules to install
langsupport  --default=en_US
#System keyboard
keyboard us
#System mouse
mouse none
#Sytem timezone
timezone America/New_York
#Root password
rootpw --iscrypted $1$7rTArc8M$Un3VRAb5rS204seG4pEtY1
#Reboot after installation
reboot
#Use text mode install
text
#Install Red Hat Linux instead of upgrade
install
#Use NFS installation Media
nfs --server=192.168.100.110 --dir=/install/rhas
#System bootloader configuration
bootloader --location=partition --append console=hvc0
#Partition clearing information
clearpart --all --initlabel
#Disk partitioning information
autopart
#part None --fstype "PPC PReP Boot" --size 4 --asprimary
#part / --fstype ext3 --size 8192 --asprimary
#System authorization infomation
auth  --useshadow  --enablemd5
#Network information
network --bootproto=dhcp --device=eth0 --hostname lpar6
#Firewall configuration
firewall --disabled
#Do not configure XWindows
#skipx
```

[14] Jeremy Katz <katzj@redhat.com>, https://listman.redhat.com/archives/kickstart-list/2003-October/msg00152.htm

```
#Package install information
%packages
@ base-x
@ text-internet
@ server-cfg
@ dialup
@ admin-tools
@ graphical-internet
@ compat-arch-support
kernel
yaboot

%post
```

### Passing the kickstart argument to the kernel

Unlike SLES 8, RHAS 3 does not provide a trick for burning command line arguments in the kernel, so we have to boot to Open Firmware, retrieve the kernel stored (netboot.img file from the images directory on RHAS 3 CD1) on the server in the /tftpboot directory, and then start it with the appropriate arguments. This is shown in Example 2-31. Note that the syntax for the location of the control file on NFS is different from SLES 8.

*Example 2-31 Kickstart command line arguments*

```
          1 = SMS Menu                          5 = Default Boot List
          6 = Stored Boot List                  8 = Open Firmware Prompt


     memory      keyboard     network     scsi     speaker  ok
0 > boot net ks=nfs:192.168.100.110:/install/rhas/ks.cfg
```

## SIS: the Open Source way, but not for pSeries yet

System Installation Suite[15] (SIS) is an open source project that aims at providing a way to massively install Linux on similar nodes. Unlike autoyast2 or kickstart, where you specify the list of packages to be installed, SIS uses a concept very similar to that of mksysb in AIX.

One node, designated as a golden client, is primarily installed and customized. Some additional software could be installed or a kernel recompiled. Once the golden client is ready for replication, the client part of SIS is used to create a snapshot of the node. This snapshot (an AIX administrator would call it a *mksys*), is moved onto a SIS server.

[15] http://sisuite.org

On this server, a specific SIS kernel and initrd file are used to produce a bootable medium (CD-ROM) or set up an image to be used for network installation.

SIS is supported by CSM on xSeries®. Unfortunately, the network installation part is heavily dependent on the Pre eXecution Environment (PXE) from Intel®, which is not used on pSeries. PXE is responsible for handling the dialog between a network interface card and a DHCP server, and is commonplace on IA32 and IA64 servers.

There is more to SIS than the initial installation. It has useful features for quickly applying updates to a collection of nodes. However, the support for pSeries hardware is still embryonic. The client part is fairly hardware-independent and we were able to create an image of a pSeries node, but we could not get the server part to work properly. Stay tuned and watch for announcements on:

`http://sourceforge.net/projects/systemimager/`

## 2.4 Where is the BIOS

It is likely that our Linux readers are more accustomed to Intel PCs than to pSeries systems. There are obviously significant differences between them, and we describe them here. Apart from performance or scalability, the differences show at various levels:

- Processor - PowerPC versus Intel
- System architecture - CHRP versus PC
- Firmware - Open Firmware versus BIOS
- Booting, yaboot versus grub/lilo - a consequence of the difference in firmware

### 2.4.1 PowerPC

All pSeries systems use processors that implement the PowerPC architecture. Chips based on this architecture can also be found in the Apple Macintosh (G3, G4, G5) and in many embedded systems from various manufacturers. You can even find PowerPC chips in game consoles. More information on the family of PowerPC chips can be found at:

`http://www.ibm.com/chips/products/powerpc/`

IBM manufactures processors based on the PowerPC architecture for its own systems (POWER4 is the current declination), but it also manufactures PowerPC chips for other equipment manufacturers.

## 2.4.2 CHRP

All pSeries systems are built around PowerPC processors, and all obey the Common Hardware Reference Platform (CHRP) specification.

In the lifetime of the Power architecture, the first systems in 1990 were based on the Micro Channel® Architecture (MCA) bus and were named RS6K systems. These systems were built on the Power technology but not on the PowerPC architecture; that was instantiated for the first time with the desktop model 250 in 1993.

Later on, a first attempt was made to define a specification for building systems around the PowerPC processor. Called PReP (PowerPC Reference Platform®), it first came to life with the desktop model 43P in 1995, architected around the 604 PowerPC chip.

Since the introduction of the model S70, based on the PowerPC RS64 chip, in 1997 and the desktop model 43P-260, based on the POWER3 chip, all the systems obey the CHRP specification. The older PReP systems which were still supported with AIX 5.1 are no longer supported with AIX 5.2.

The CHRP specification is available from:

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

This specification comes with two parts: one describing the base architecture (NVRAM, RTAS, power management), and the other describing the I/O devices. The specification describes how the firmware needs to abstract the hardware and present a consistent interface to the operating systems that may run on a CHRP system.

## 2.4.3 Open Firmware basics

pSeries CHRP systems use a firmware that conforms to the open firmware IEEE Std. 1275-1994. Open Firmware is primarily a boot firmware that does not specify a particular system or processor. It also provides a machine independent devices tree interface that lists the properties of each IO device which can be displayed and altered using a command line interface.

There are Open Firmware bindings for a variety of systems architectures. The one that pertains to pSeries is described in:

http://playground.sun.com/1275/bindings/chrp/chrp1_8a.ps

The home of the Open Firmware working group is located at:

http://playground.sun.com/1275.

The Open Firmware standard derives from the Sun Microsystems OpenBoot and is currently being developed mainly by Sun, Apple and IBM. It complies with the ANSI Forth language as described at:

http://www.forth.org/

## Open Firmware prompt

pSeries systems can be booted to SMS or to the open firmware prompt, sometimes referred to as the ok prompt. You can reach the open firmware prompt by hitting the 8 key while booting. If under AIX, system administrators rarely have to use this.

For installing Linux, however, we sometimes had to boot to the OK prompt to pass arguments to the kernel, as described in Example 2-29 on page 75.

The Open Firmware prompt is also used extensively by the **netboot** command in CSM to boot/install from the network. SMS itself is an Open Firmware application.

Next, we give a very brief introduction to the Open Firmware prompt command line interface. A quick reference guide is available from:

http://www.firmworks.com/QuickRef.html

A thorough description can be found at these sites:

http://developer.apple.com/technotes/tn/tn1061.html

http://developer.apple.com/technotes/tn/tn1062.html

## Open Firmware commands

Booting the system in Open Firmware gives you the screen shown in Example 2-32.

*Example 2-32 Open Firmware prompt*

```
      1 = SMS Menu                          5 = Default Boot List
      6 = Stored Boot List                  8 = Open Firmware Prompt


   memory      keyboard     network     scsi     speaker  ok
0 >
```

You can navigate inside the device tree. Upon startup, you are at the top of the device tree. Use the **ls** command to display the device tree, a walk along all the buses and adapters that constitute the system under examination. It is shown in Example 2-33 on page 82.

*Example 2-33 ls lists the device tree*

```
0 > ls
000000c8fee0: /ibm,serial
000000c9c6c0: /chosen
000000ca0290: /cpus
000000ca39b8:   /PowerPC,POWER4@2
000000ca60e0:   /L2-cache@2001
000000ca65a8:   /L3-cache@3001
000000ca6c28: /memory@0
000000ca9100: /memory@10000000
000000ca93f0: /memory@20000000
000000ca96e0: /memory@30000000
000000ca9ac8: /memory-controller@7000000000000
000000caa698:   /IBM,memory-module@0
000000caaa48:   /IBM,memory-module@1
000000caadf8:   /IBM,memory-module@2
000000cab1a8:   /IBM,memory-module@3
000000cabd88: /memory-controller@7000000000010
000000cac958:   /IBM,memory-module@0
........
000000d1c690: /rtas
000000d31358: /vdevice
000000d317b0:   /vty@0
000000d321e0:   /IBM,sp@4000
000000d3a420:   /rtc@4001
000000d3bd70: /pci@400000000110
000000d43168: /pci@400000000111
000000d49b80:   /pci@2,4
000000d5ff60:     /ethernet@1
000000d511d8: /pci@400000000112
000000d57bd8:   /pci@2,2
000000d6b4c8:     /scsi@1
000000d722f8:       /sd
000000d737c0:       /st
000000d74f10:     /scsi@1,1
000000d7bd40:       /sd
000000d7d208:       /st
 ok
0 >
```

You can navigate back and forth in the device tree by using the **dev** command, and you can check where you are in the device tree by using the **pwd** command. This is shown in Example 2-34 on page 83, where we first point to the Ethernet adapter, then move backward, and finally return to the root of the device tree.

Do not type ok, as this is echoed by the Open Firmware.

*Example 2-34 dev and pwd to navigate the device tree*

```
0 > dev /pci@400000000111/pci@2,4/ethernet@1  ok
0 > pwd /pci@400000000111/pci@2,4/ethernet@1 ok
0 > dev .. ok
0 > pwd /pci@400000000111/pci@2,4 ok
0 > dev / ok
0 > pwd / ok
0 >
```

It can be interesting to show the properties attached to the devices. Example 2-35 displays the properties of our Ethernet adapter. Displaying the properties at the root of the device tree gives general information about the system.

*Example 2-35 .properties lists the adapter properties*

```
0 > dev /pci@400000000111/pci@2,4/ethernet@1
0 > .properties
ibm,loc-code           U0.1-P2-I5/E1
vendor-id              00008086
device-id              00001229
revision-id            0000000d
class-code             00020000
....
device_type            network
supported-network-types ethernet,auto,rj45,auto
                       ethernet,10,rj45,half
                       ethernet,10,rj45,full
                       ethernet,100,rj45,half
                       ethernet,100,rj45,full
max-frame-size         00000a28
address-bits           00000030
local-mac-address      0002556f 1fef
mac-address            0002556f 1fef
ibm,fw-adapter-name    10/100 Mbps Ethernet PCI Adapter II
ibm,fw-driver-version  00000150
chosen-network-type    ethernet,100,rj45,full
ibm,fw-revision-level  00000150
ok
0 > dev /
0 > .properties
ibm,model-class        G5
ibm,pci-full-cfg       00000001
clock-frequency        17d78400
device_type            chrp
#address-cells         00000002
#size-cells            00000002
ibm,max-boot-devices   00000005
```

```
ibm,aix-diagnostics
ibm,extended-address
ibm,lpar-capable
ibm,partition-no        00000002
ibm,partition-name      6c706172 3800
ibm,fw-sp-per-current   3000
ibm,fw-sp-per-default   3000
ibm,fw-sp-l3-current    3000
ibm,fw-sp-l3-default    3000
ibm,fw-sp-l3prefetch-current
                        3000
ibm,fw-sp-l3prefetch-default
                        3000

....
name                    IBM,7038-6M2
model                   IBM,7038-6M2
compatible              IBM,7038-6M2
system-id               IBM,0110197AA
 ok
0 >
```

Some devices have aliases which you can use to avoid typing long device names, especially when using the **boot** command. You can list these devices with the **devalias** command, as shown in Example 2-36.

In our example, net is an alias for the Ethernet adapter, and disk is an alias for the first SCSI disk.

*Example 2-36 devalias command*

```
0 > devalias
ibm,sp              /vdevice/IBM,sp@4000
disk                /pci@400000000112/pci@2,2/scsi@1/sd@0,0
scsi                /pci@400000000112/pci@2,2/scsi@1
screen              /vdevice/vty@0
net                 /pci@400000000111/pci@2,4/ethernet@1
network             /pci@400000000111/pci@2,4/ethernet@1
rtc                 /vdevice/rtc@4001
nvram               /nvram@3fdbd800000
 ok
0 >
```

Persistent configuration variables can be displayed with the **printenv** command, as shown in Example 2-37 on page 85.

*Example 2-37 printenv command*

```
0 > printenv
....
screen-#rows             28                  28
selftest-#megs           0                   0
boot-device              /pci@400000000112/pci@2,2/scsi@1/sd@0,0
boot-file
diag-device              /pci@400000000112/pci@2,2/scsi@1/sd@0,0
diag-file                diag                diag
output-device            /vdevice/vty        com1
input-device             /vdevice/vty        com1
oem-banner
oem-logo
nvramrc                  Defined : use NVEDIT related words to view
boot-command             boot                boot
reboot-command
security-#badlogins      0                   0
...
```

The command that we will most likely use is the **boot** command. This is the only way we found to give command line arguments to the Linux kernel when booting from the network.

We use the alias for the network to do that, passing along the arguments to the kernel. This is shown in Example 2-38, where we boot off the network (BOOTP) and pass along the two modules to load when booting.

*Example 2-38 boot command*

```
0 > boot net insmod=e100 insmod=lvm-mod
```

To leave the Open Firmware prompt and jump into SMS, use the command shown in Example 2-39, and press <1>.

*Example 2-39 Back to SMS*

```
0 > dev /packages/gui obe
```

## See the device tree from Linux

The device tree shows under Linux in the /proc/device-tree directory.

## 2.4.4 Yaboot

Yaboot is a disk boot loader for PowerPC-based systems. The way to boot CHRP systems was carried over from the old PReP systems, so that it uses a disk partition that is defined as a PReP boot partition.

The yaboot home page is located at:

http://penguinppc.org/projects/yaboot/doc/yaboot-howto.shtml/

SLES 8 provides a script called lilo that is more “natural” for Intel Linux users to use. Only the name of the script is the same; it has nothing to do with the Linux loader on PCs.

The lilo script stores the yaboot.chrp file raw inside the PReP boot partition and updates the /etc/yaboot.conf file based on the contents of the /etc/lilo.conf file. The simplest /etc/lilo.conf file specifies the location of the kernel to boot, as well as the root partition to mount. Example 2-40 gives a basic example.

*Example 2-40 Sample /etc/lilo.conf file*

```
# Generated by YaST2

default=test
timeout=100
boot=/dev/sda1
activate

image = /boot/vmlinuz
        label = linux
        root = /dev/sda2
        append = ""
```

To understand how yaboot operates, start it in debug mode. This can be done by copying the /boot/yaboot.chrp.debug file raw in the PReP boot partition with the command shown in Example 2-41.

**Note:** This is not needed in regular operation and is shown here only to demonstrate the process.

*Example 2-41 Run yaboot in debug mode*

```
# dd if=/boot/yaboot.chrp.debug of=/dev/sda1 bs=512
```

Then reboot the node and watch the output at the system console; see Example 2-42 on page 87.

*Example 2-42   Watching yaboot*

```
Adding OF methods...
    prom_init - OF interface initialized.
    yaboot_start - Malloc buffer allocated at 00300000 (1048576 bytes)
    yaboot_start - reloc_offset :  0         (should be 0)
    yaboot_start - test_bss     :  0         (should be 0)
    yaboot_start - test_data    :  0         (should be 0)
    yaboot_start - &test_data   :  00227780
    yaboot_start - &test_bss    :  00227f60
    yaboot_start - linked at    :  0x00200000
    yaboot_start - Running on _machine = 4
    yaboot_main - /chosen/bootpath = /pci@400000000111/pci@2/scsi@1/sd@8,0
...
file_name = /etc/yaboot.conf
partition = -1
    open_file - device is a block device
    partitions_lookup - block size of device is 512
partitions:
    file_block_open - number: 02, start: 0x00003ec1, length: 0x0080344b
--> ext2_open
....
ext2_open - ext2fs_open returned bad magic loading file 0291fca8
<-- ext2_open - -8
...
--> xfs_open
    xfs_open - dev=/pci@400000000111/pci@2/scsi@1/sd@8,0, part=0x00308068 (2),
file_name=/etc/yaboot.conf
...
xfs_mount - xfs_mount: Bad magic: 0
    xfs_open - Couldn't open XFS @
/pci@400000000111/pci@2/scsi@1/sd@8,0:0/8225280
<-- xfs_open - FILE_ERR_BAD_FSYS
--> reiserfs_open
    reiserfs_open - dev=/pci@400000000111/pci@2/scsi@1/sd@8,0, part=0x00308068
(2), file_name=/etc/yaboot.conf
...
Config file read, 160 bytes
    reiserfs_close - reiserfs_close called
    load_config_file - Config file successfully parsed, 160 bytes
Welcome to yaboot version 1.3.6.SuSE
Enter "help" to get some basic usage information
boot:
```

At startup time, yaboot will determine, from the firmware, which disk it was started from. Then, yaboot has to locate its /etc/yaboot.conf configuration file. But how can it locate a file in /etc when there is no root device mounted?

The answer is that yaboot has a built-in knowledge of three types of file systems: ext2/3, xfs, and reiserfs. It tries to open the partitions, looking for a valid /etc/yaboot.conf file.

In our example, yaboot tries ext2 first and fails, then tries xfs and fails, and finally succeeds opening a reiserfs file system with the /etc/yaboot.conf file. Once the configuration file is read, yaboot can then record the location of the kernel(s) referenced in the file and prompt the user to choose which kernel to use.

## 2.5 Post-installation tasks

At this point, we have installed the Linux operating system on our system and defined at least one network interface, and can access it remotely. Before customizing the system and making it available to our user community (developers, database administrators, Web administrators, computer scientists), we may need to perform a few other tasks such as checking the installation and applying patches. We describe these tasks in the following sections.

### 2.5.1 Checking the installation

SLES 8 provides a tool called **hwinfo** that will list all the hardware it finds. It is useful to check this list against your expectations, for example, the number and speed of the processors, memory configuration, and I/O adapters. The output of **hwinfo** is very detailed.

For I/O adapters, lspci can also be used; it is available under SLES 8 and RHAS 3.

### 2.5.2 Applying SLES 8 patches

SLES 8 and RHAS 3 each offer an automatic update facility, usually through an Internet connection to their respective support Web sites. This process is documented in 3.10, “System updates” on page 139. What we describe here are the steps to apply a patch CD to a newly installed SLES 8 distribution.

Service Packs (SPs) are patch CDs that consolidate in one media a whole set of individual patches known to work together. To apply these patches, use the **yast2** command. We assume that you have installed the default system. However, if you only installed the minimal system, only the curses version of yast2 is available. There is nothing wrong with this but the screens will obviously be different from the ones we show here.

Using the command **yast2**, you arrive at the screen shown in Figure 2-40 on page 89.

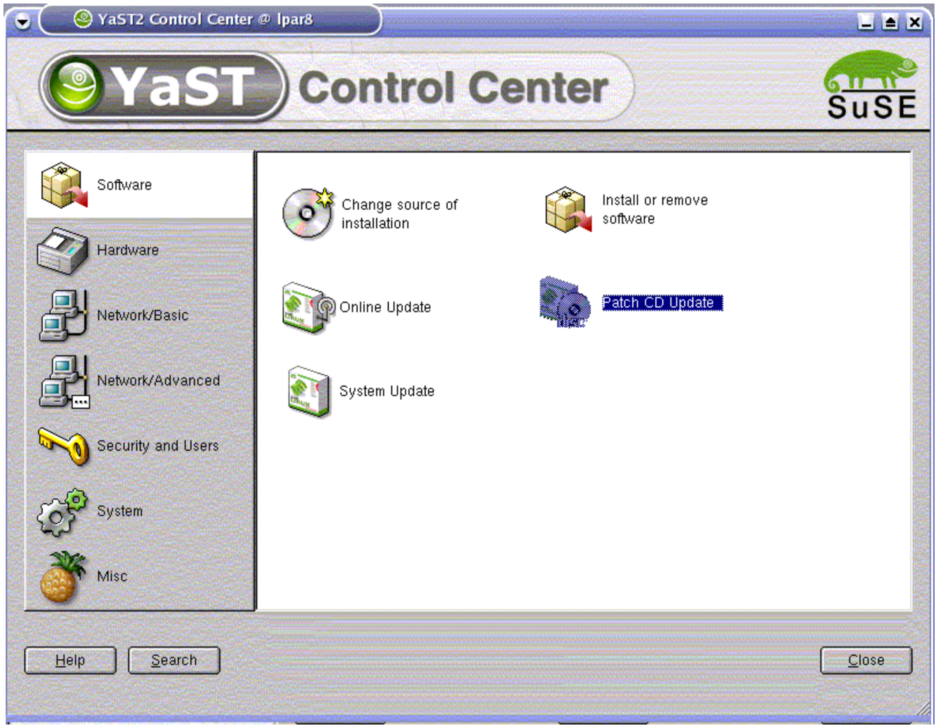
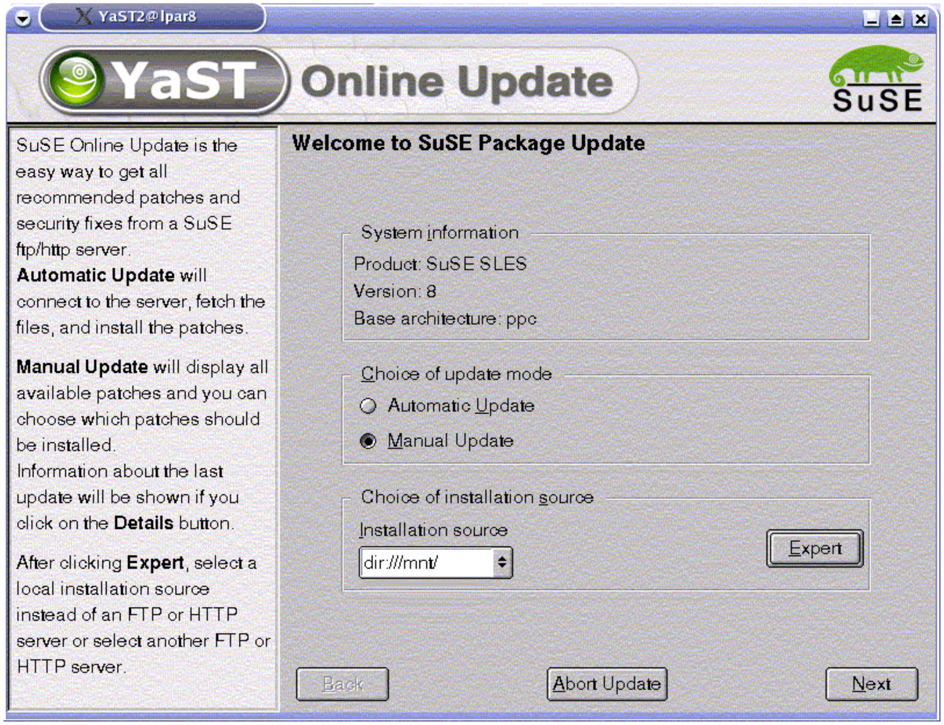


*Figure 2-40 SLES 8 software update*

Choose the **Patch CD update** option. You can then choose to back up the current system before proceeding. This “backup”, as well as the contents of the /etc/sysconfig directory, is saved under /var/adm/backup/rpmdb in the RPM database.

Once this backup is finished, you are presented with the update screen shown in Figure 2-41 on page 90. In our case, the SP3 CD was NFS-mounted under /mnt, thus the choice for the installation source.



SuSE Online Update is the easy way to get all recommended patches and security fixes from a SuSE ftp/http server.
**Automatic Update** will connect to the server, fetch the files, and install the patches.

**Manual Update** will display all available patches and you can choose which patches should be installed.
Information about the last update will be shown if you click on the **Details** button.

After clicking **Expert**, select a local installation source instead of an FTP or HTTP server or select another FTP or HTTP server.

**Welcome to SuSE Package Update**

System information
Product: SuSE SLES
Version: 8
Base architecture: ppc

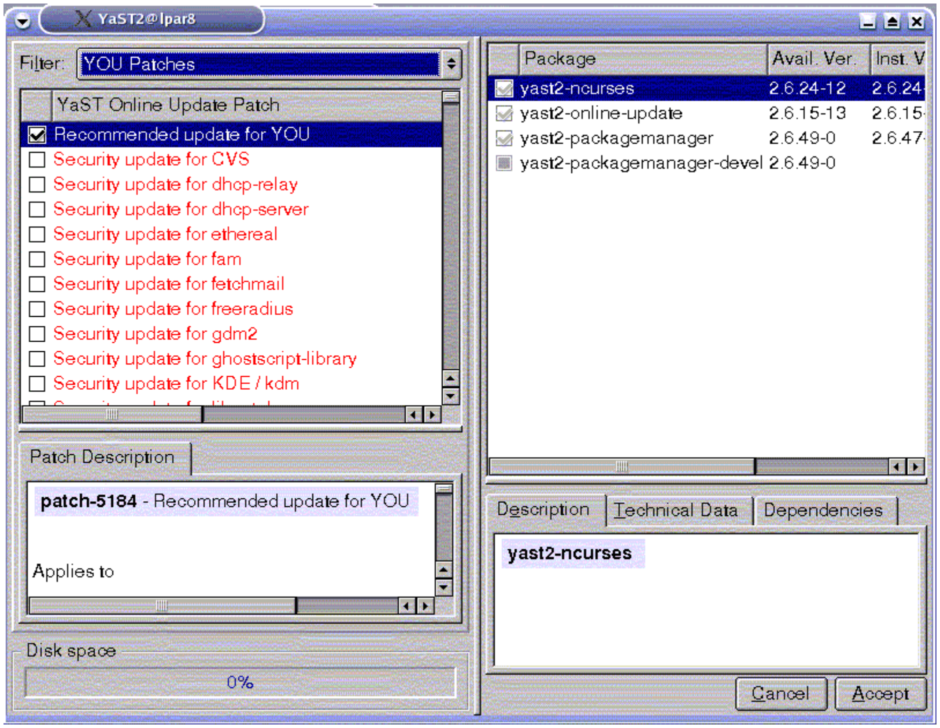
Choice of update mode
○ Automatic Update
● Manual Update

Choice of installation source
Installation source
dir:///mnt/

*Figure 2-41 Online update welcome screen*

The next step will update YaST itself and terminate the update process. You will be asked to restart YaST2 again to actually perform the rest of the update.

The first step is shown in Figure 2-42 on page 91.



Filter: YOU Patches

| | YaST Online Update Patch |
|---|---|
| ☑ | Recommended update for YOU |
| □ | Security update for CVS |
| □ | Security update for dhcp-relay |
| □ | Security update for dhcp-server |
| □ | Security update for ethereal |
| □ | Security update for fam |
| □ | Security update for fetchmail |
| □ | Security update for freeradius |
| □ | Security update for gdm2 |
| □ | Security update for ghostscript-library |
| □ | Security update for KDE / kdm |

| | Package | Avail. Ver. | Inst. V |
|---|---|---|---|
| ☑ | yast2-ncurses | 2.6.24-12 | 2.6.24 |
| ☑ | yast2-online-update | 2.6.15-13 | 2.6.15 |
| ☑ | yast2-packagemanager | 2.6.49-0 | 2.6.47 |
| ■ | yast2-packagemanager-devel | 2.6.49-0 | |

Patch Description

**patch-5184** - Recommended update for YOU

Applies to

Description | Technical Data | Dependencies

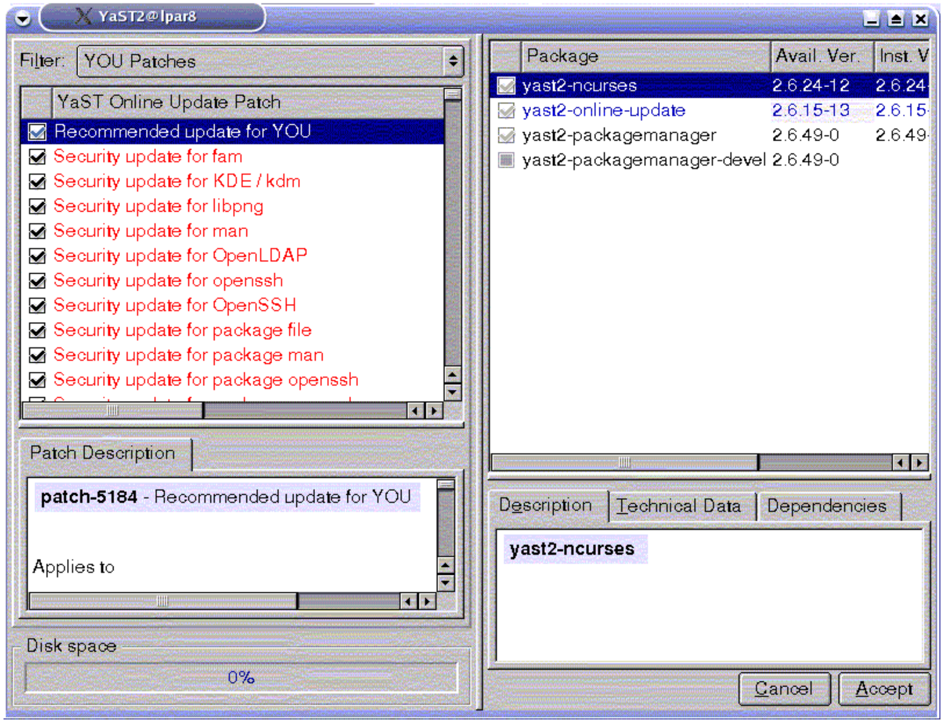
**yast2-ncurses**

Disk space

0%

*Figure 2-42 Update the updater first!*

Once this update finishes, we restart the whole process from the beginning. This time, we arrive at the screen showing all patches ready for installation; see Figure 2-43 on page 92.



Filter: YOU Patches

| | YaST Online Update Patch |
|---|---|
| ☑ | Recommended update for YOU |
| ☑ | Security update for fam |
| ☑ | Security update for KDE / kdm |
| ☑ | Security update for libpng |
| ☑ | Security update for man |
| ☑ | Security update for OpenLDAP |
| ☑ | Security update for openssh |
| ☑ | Security update for OpenSSH |
| ☑ | Security update for package file |
| ☑ | Security update for package man |
| ☑ | Security update for package openssh |

**patch-5184** - Recommended update for YOU

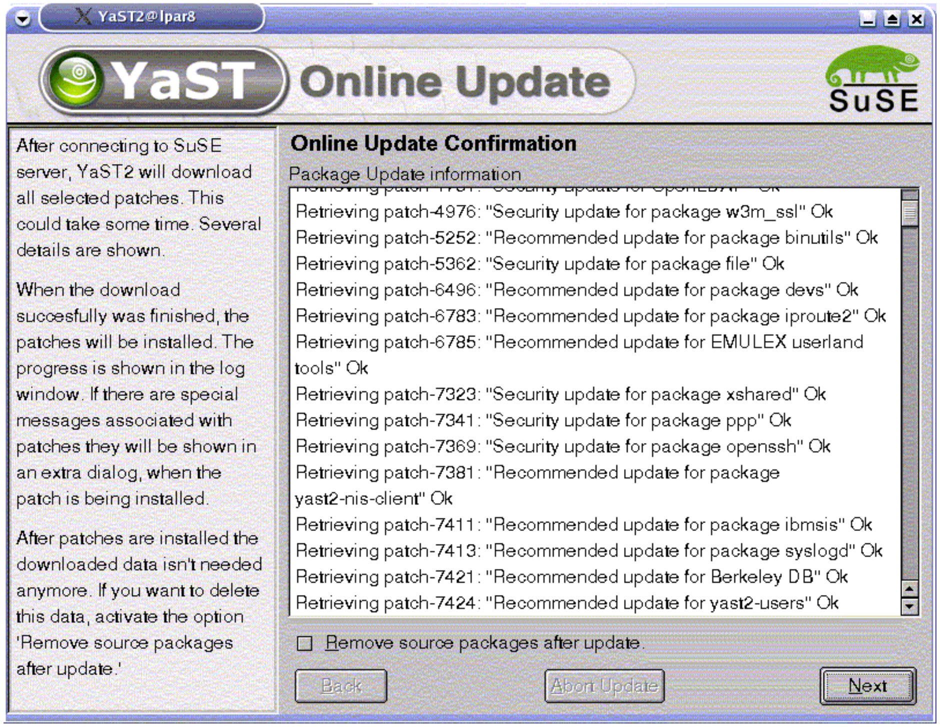
Applies to

| | Package | Avail. Ver. | Inst. V |
|---|---|---|---|
| ☑ | yast2-ncurses | 2.6.24-12 | 2.6.24- |
| ☑ | yast2-online-update | 2.6.15-13 | 2.6.15- |
| ☑ | yast2-packagemanager | 2.6.49-0 | 2.6.49- |
| ▨ | yast2-packagemanager-devel | 2.6.49-0 | |

**yast2-ncurses**

*Figure 2-43 List of software updates to apply*

Applying all the patches takes a while. You can follow the update progress as shown in Figure 2-44 on page 93.



After connecting to SuSE server, YaST2 will download all selected patches. This could take some time. Several details are shown.

When the download succesfully was finished, the patches will be installed. The progress is shown in the log window. If there are special messages associated with patches they will be shown in an extra dialog, when the patch is being installed.

After patches are installed the downloaded data isn't needed anymore. If you want to delete this data, activate the option 'Remove source packages after update.'

**Online Update Confirmation**

Package Update information

```
Retrieving patch-4976: "Security update for package w3m_ssl" Ok
Retrieving patch-5252: "Recommended update for package binutils" Ok
Retrieving patch-5362: "Security update for package file" Ok
Retrieving patch-6496: "Recommended update for package devs" Ok
Retrieving patch-6783: "Recommended update for package iproute2" Ok
Retrieving patch-6785: "Recommended update for EMULEX userland
tools" Ok
Retrieving patch-7323: "Security update for package xshared" Ok
Retrieving patch-7341: "Security update for package ppp" Ok
Retrieving patch-7369: "Security update for package openssh" Ok
Retrieving patch-7381: "Recommended update for package
yast2-nis-client" Ok
Retrieving patch-7411: "Recommended update for package ibmsis" Ok
Retrieving patch-7413: "Recommended update for package syslogd" Ok
Retrieving patch-7421: "Recommended update for Berkeley DB" Ok
Retrieving patch-7424: "Recommended update for yast2-users" Ok
```

□ Remove source packages after update.

*Figure 2-44 Monitoring the update progress*

You will need to reboot after the update if the kernel was updated too, which is very likely.

## 2.5.3 Where to get help

The primary source for support is the maintenance contract you get when buying SLES 8 or RHAS 3. However, there are also many resources on the Web that you can call upon.

### General Web sites

The primary sources of information for kernel-related issues are:

http://penguinppc.org
http://penguinppc64.org

## IRC channels

On irc.freenode.net, these channels are related to Linux on PowerPC:

- #ppc64: powerpc64 kernel development
- #ppclinux:
- #yaboot: the powerpc boot loader
- #mklinux: powerpc linux development channel
- #debianppc: unofficial debian powerpc channel
- #debianppc64 on irc.debian.org
- #yellowdog:
- #gentooppc

## Mailing lists

For all mailing lists related to Linux on PowerPC, see:

http://lists.linuxppc.org/

Lists are also available from:

http://lists.penguinppc.org

You may also wish to check:

http://lists.debian.org/debian-powerpc/

## News group

The primary news group for Linux on PowerPC is located at:

http://comp.os.linux.powerpc

# 3

# System administration

Recognizing that there are many Linux administration guides and most everyday administration tasks are covered by SuSE Administration Guide provided on the distribution CDs, in this chapter we concentrate on a brief introduction to local and cluster system administration. We describe the administration tools on a Linux system and discuss user administration issues, logical volume management and selected network topics. We also describe system update and system backup possibilities.

With or without CSM, a cluster or enterprise environment with more than a single Linux system will challenge an administrator. Administration of many network systems can be just a sum of single system administration tasks, but in this case there will be many perpetual and repeated steps. For effective enterprise system administration, you need tools that will focus these tasks at a single point of control and accomplish them once for the whole cluster.

We will use DHCPD for dynamic address configuration, BIND for name resolution, and LDAP or file distribution for user management. An environment built with these tools remains flexible, and adding new nodes is no additional effort.

We provide step-by-step instruction on how to get these tools running. All the configuration files were tested in our environment and should be able to serve your needs with only minor changes.

## 3.1 The bash shell

The most powerful and flexible administration tool on almost every UNIX-like system is the shell. The standard shell in a SuSE SLES for pSeries 8.0 is the Bourne again shell, or bash. It is an sh-compatible shell that incorporates useful features from the Korn shell (ksh) and C shell (csh). It is intended to conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools standard. Most sh scripts can be run by bash without modification.

### Command history

The standard history mode on bash is emacs; to view all of the current settings of the shell, issue the **`set -o`** command:

```
# set -o
```

If you are more comfortable using vi than the emacs mode, use the **`set -o vi`** command.

The default history size is 500 entries, but this can be changed by setting the HISTSIZE variable.

Command line history keys are mapped to the arrow keys by default. You can navigate the history by pressing arrow up and arrow down keys. Pressing Ctrl + R allows you to search in history; simply start typing your command and bash will display the first match in the history. You can then continue typing to get other results, or press Ctrl + R again to search backwards according to the characters already given.

### Command completion

The bash shell provides command completion, so typing the first letters from a command and pressing the Tab key will give you the choices from all executables in your path.

```
# cs TAB TAB
csh     csplit
```

Typing `TAB TAB` after the executable will show the files in the current directory:

```
lpar5:/etc/sysconfig/network # vi TAB TAB
config           ifcfg-eth0       ifcfg.template   wireless
dhcp             ifcfg-eth0.ORIG  providers
if-down.d        ifcfg-eth1       routes
if-up.d          ifcfg-lo         scripts
```

### The bash configuration files

When bash is invoked as an interactive login shell, or as a non-interactive shell with the `--login` option, it first reads and executes commands from the file /etc/profile, if that file exists. After reading that file, it looks for ~/.bash_profile, ~/.bash_login, and ~/.profile, in that order, and reads and executes commands from the first one that exists and is readable.

When an interactive shell that is not a login shell is started, bash reads and executes commands from ~/.bashrc, if that file exists.[1]

In SuSE SLES, the /etc/profile file should not be changed because it can be overwritten by an update. Global shell customization should be done by editing the /etc/profile.local instead. User scripts should be placed in ~/.bashrc in order to make sure it will be used even if a login shell is not started.

### Alternatives to bash

Open source software offers freedom of choice. If you prefer not to use bash, you can choose other shells, such as the following::

- **tcsh** - an enhanced but completely compatible version of the Berkeley UNIX C shell, csh.[2]
- **ksh** -a public domain Korn shell. PD-ksh is a clone of the AT&T Korn shell. It currently offers most of the ksh88 features, not much of the ksh93 features, and a number of its own features.[3]
- **zsh** - the Z shell. Of the standard shells, zsh most closely resembles ksh, but includes many enhancements.[4]
- **ash** - a version of sh with features similar to those of the System V shell.[5]
- **sash** - a standalone shell with built-in commands (used for system recovery and installation).

## 3.2 To YaST or not to YaSt

Yet another Setup Tool (YaST) is the main administration instrument on a SuSE Linux distribution. It combines most common administration tasks. YaST is not distributed under the General Public License (GPL), but has its own license; see:

```
http://www.suse.de/en/private/support/licenses/yast.html
```

[1] See man bash
[2] See man tcsh and http://www.tcsh.org for more information
[3] See man pdksh and http://web.cs.mun.ca/~michael/pdksh/ for more Information
[4] See man zsh and http://www.zsh.org/ for more information
[5] See man ash

YaST can be used in graphical mode or in a terminal window with ncurses. It is modular and opens new windows in a graphical mode for each module. You can get the list of available modules on your system (and shortpaths) by issuing the **yast2 -list** command, as shown in Example 3-1.

*Example 3-1 AvailableYaST modules and shortpaths*

```
# yast2 -list
Available modules:

autoyast
backup
bootfloppy
disk
dns
dsl
firewall
hwinfo
idedma
inetd
inst_source
isdn
joystick
keyboard
lan
language
ldap
lilo
lvm_config
mail
modem
mouse
nfs
nfs_server
nis
nis_server
online_update
powertweak
printer
profile-manager
proxy
restore
routing
runlevel
security
sound
sw_single
sysconfig
timezone
```

```
update
users
vendor
x11
```

YaST is easy to use and in graphical mode, it visualizes complicated tasks in a comprehensible way. We recommend using it, for example, for partitioning, for the first setup of Logical Volume Manager (LVM), and so on. YaST will prevent you from making certain errors and at least warn you in critical cases.

YaST makes changes to the system directly; it does not maintain its own configuration database (as was the case in older versions of SuSE). This means that we can combine administration by editing appropriate configuration files and through YaST, without producing system inconsistencies. For example, if we edit /etc/resolv.conf and run **`yast2 dns`** after that, it will show the current entries and allow us to edit them.

On the other hand, we do see some disadvantages to using YaST in a network environment:

- YaST does a great deal of logging, but it does not log which files are changed and which commands are called to achieve the particular task. You can find YaST logs in /var/log/YaST2/y2log. It differs from the AIX administration tool smit it that you cannot use the information in the logs for task automation through scripting.
- YaST is a single system administration tool, so you can only change the system you are using; you cannot run the same command on different systems.
- YaST does not provide a “bash mode”, so you cannot include the task that YaST should complete in the command line arguments or in a script.
- Some modules provided by YaST are incomplete; that is, they do not offer all possibilities and are restricted in functionality. For example, it is not possible to change RAID devices, to assign a whole physical disk to LVM, or to manage LDAP users through YaST.

**Tip:** We used tripwire to find out which files YaST is changing. To use tripwire, you need to generate a tripwire database first, then do anything with YaST and run tripwire again to see which files are changed

## 3.3 Web-based system administration (Webmin)

Webmin is a Web-based system administration tool for most UNIX systems. You need a frame-capable browser with Java support (this is only required for some modules).

Webmin has a modular structure it is easy to customize. Webmin has a built-in ability to delegate administration tasks to non-root users.

### 3.3.1 Webmin installation

To install Webmin, download the tar.gz file from:

`http://www.webmin.com`

Uncompress and untar it to the /opt directory.

Run the setup.sh script and answer the dialogs as shown in Figure 3-2.

> **Important:** In a production environment, we recommend that you enable ssl support and use secure password (keep in mind that the main webmin user has all the root privileges!).

*Example 3-2 Webmin installation*

```
lpar3:/opt/webmin-1.120 # ./setup.sh
************************************************************************
*            Welcome to the Webmin setup script, version 1.120         *
************************************************************************
Webmin is a web-based interface that allows Unix-like operating
systems and common Unix services to be easily administered.

Installing Webmin in /opt/webmin-1.120 ...

************************************************************************
Webmin uses separate directories for configuration files and log files.
Unless you want to run multiple versions of Webmin at the same time
you can just accept the defaults.

Config file directory [/etc/webmin]:
Log file directory [/var/webmin]:

************************************************************************
Webmin is written entirely in Perl. Please enter the full path to the
Perl 5 interpreter on your system.

Full path to perl (default /usr/bin/perl):
```

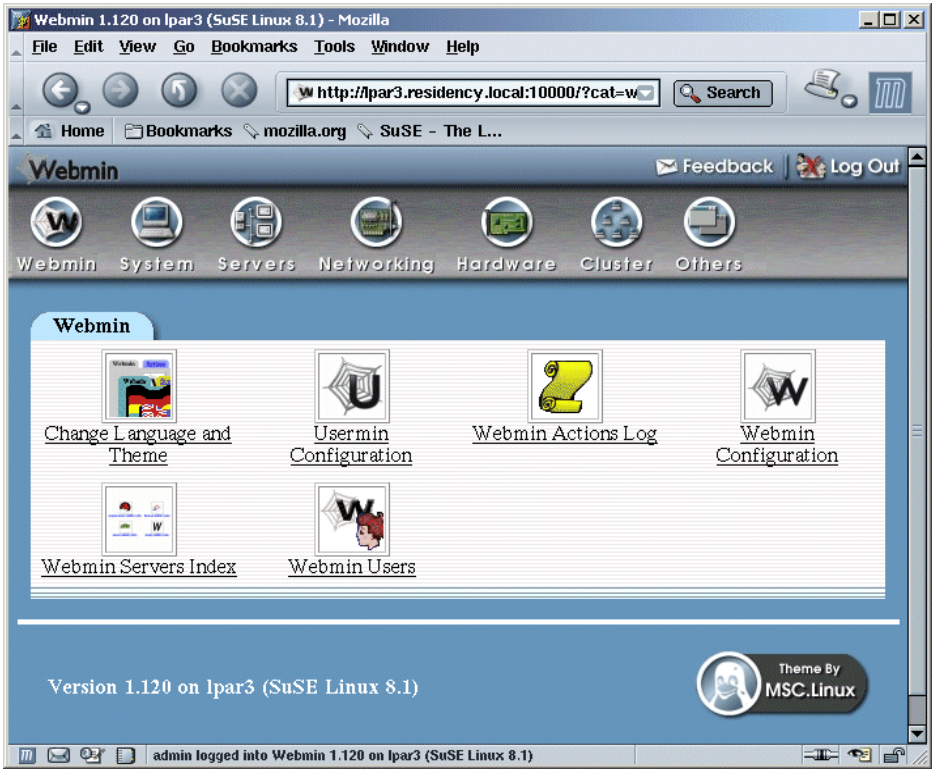
```
Testing Perl ...
Perl seems to be installed ok

***********************************************************************
Operating system name:    SuSE Linux
Operating system version: 8.1

***********************************************************************
Webmin uses its own password protected web server to provide access
to the administration programs. The setup script needs to know :
 - What port to run the web server on. There must not be another
   web server already using this port.
 - The login name required to access the web server.
 - The password required to access the web server.
 - If the webserver should use SSL (if your system supports it).
 - Whether to start webmin at boot time.

Web server port (default 10000):
Login name (default admin):
Login password:
Password again:
Use SSL (y/n): n
Start Webmin at boot time (y/n): y
***********************************************************************
Creating web server config files..
..done

Creating access control file..
..done

Inserting path to perl into scripts..
..done

Creating start and stop scripts..
..done

Copying config files..
..done

Configuring Webmin to start at boot time..
Created init script /etc/init.d/webmin
..done

Creating uninstall script /etc/webmin/uninstall.sh ..
..done

Changing ownership and permissions ..
..done
```

```
Running postinstall scripts ..
..done

Attempting to start Webmin mini web server..
Starting Webmin server in /opt/webmin-1.120
..done

***********************************************************************
Webmin has been installed and started successfully. Use your web
browser to go to

  http://lpar3:10000/

and login with the name and password you entered previously.
```

Now we point our Web browser to:

```
http://lpar3.residency.local:10000/
```

We log in as admin with the password we provided, as shown in Figure 3-1 on page 103.



Version 1.120 on lpar3 (SuSE Linux 8.1)

admin logged into Webmin 1.120 on lpar3 (SuSE Linux 8.1)

*Figure 3-1 Webmin administration screen*

## 3.3.2 Using Webmin

By selecting **Webmin Users**, we can create dedicated Webmin users and assign the Webmin modules to them. This allows us to have, for example, a user who is only allowed to manage printers through the Webmin interface.

Webmin is also cluster-aware, so we can install it on different nodes and run cluster commands from one central administration point, and run scripts or create users on all nodes in the cluster.

Not all Webmin modules worked properly on our system; some missed some prerequisites, and others failed to complete the task. We recommend that you test each module that you want to use on a test system. However, Webmin does provide action logs with detailed descriptions of which Webmin user has issued which command.

## 3.4 Runlevels

Init is a process started by the kernel during the boot, and it takes over the control of the start sequence. The main configuration file is /etc/inittab. In this file we can provide the default runlevel. The default runlevel for systems with graphical login is 5 without 3.

The first script started by init is /etc/init.d/boot, after it runs all startscripts in /etc/init.d/rc3.d (or rc5.d, for runlevel 5). If you want to add something to be run before the runlevel scripts, add it to /etc/init.d/boot.local.

If you need to add something in runlevel 3 or runlevel 5, then place your script in /etc/init.d and place a link starting with S+number and K+number in /etc/initrd/rc5.d.

> **Tip:** On SuSE SLES8, use /etc/init.d/skeleton as a template if you need to create your own start/stop scripts.

The /etc/init.d directory contains all start and stop scripts for all runlevels. Each runlevel has its own subdirectory containing symbolic links to the scripts in /etc/init.d. These are located under /etc/init.d/rcX.d (X stands for runlevel number 1-6).

These directories contain links beginning with the letter S, to be executed with the start option when the system *enters* the runlevel. Links to scripts beginning with the letter K are going to be executed when the system is *leaving* the runlevel, and only if the new runlevel does not contain the same script.

The number determines the execution order. For example, the symbolic link in /etc/init.d/rc3.d named S09sshd will be executed when entering runlevel3 after S05network; in fact, /etc/init.d/ssh will be called with the “start” option.

The runlevel description is given in /etc/inittab and is defined through the Linux Standard base specification:[6]

- runlevel 0 is System halt (Do not use this for initdefault!)
- runlevel 1 is Single user mode
- runlevel 2 is Local multiuser without remote network (for example, NFS)
- runlevel 3 is Full multiuser with network
- runlevel 4 is Not used, reserved for local use
- runlevel 5 is Full multiuser with network and xdm
- runlevel 6 is System reboot (Do not use this for initdefault!)

[6] See http://www.linuxbase.org/spec/refspecs/LSB_1.3.0/gLSB/gLSB/runlevels.html

We wondered why SuSE asked us for the root password even when booting in single user mode (init 1). The answer is contained in the following entry in the /etc/inittab file, asking what to do in single-user mode:

```
# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

### Using chkconfig

Both SuSE and Red Hat distributions provide a chkconfig script that allows us to manipulate and display the runlevel links. The chkconfig script relies on two comment lines in the start/stop scripts; for example, in Red Hat:

```
# chkconfig: 2345 20 80
# description: Saves and restores system entropy pool
```

SuSE[7] has a different format:

```
# Default-Start:              3 5
# Default-Stop:               0 1 2 6
# Short-Description:          Apache httpd
```

**chkconfig --list** will display all services and show in which runlevels these will be started.

We can easily change it by issuing following commands:

**chkconfig apache on** on SuSE, or **chkconfig httpd on** on Red Hat, will enable us to start Apache in the default runlevels defined in the /etc/init.d/apache script.

**chkconfig --list apache** will show the changes we have made. Be aware, however, that chkconfig does not start or stop the services; it simply sets them to be started or not started when entering the runlevel.[8]

## 3.5 Local user management

Local users can be managed either by using the command line or with YaST. Usingthe command line, we issued the following command in order to create a new user:

```
# useradd -m  gnu01
```

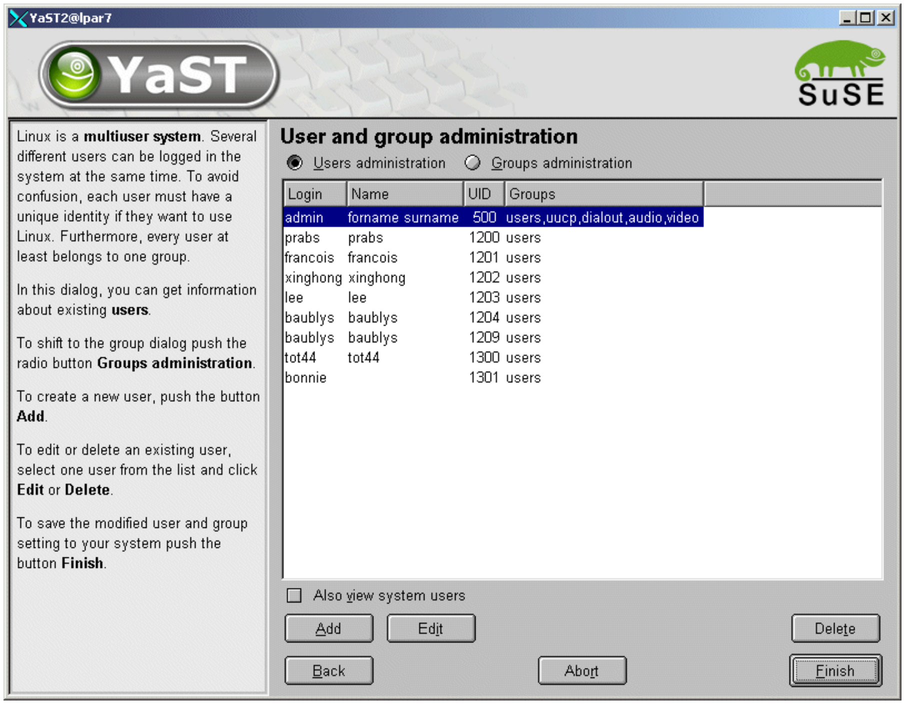
[7] The format used by SuSE is LSB compliant, see http://www.linuxbase.org/spec/refspecs/LSB_1.3.0/gLSB/gLSB.html#INITSCRCOMCONV

[8] See man chkconfig for more details

The option **-m** is used in order to create a default home directory. The contents of the default skeleton directory /etc/skel are copied to the newly created home. SuSE adds the new user to the group 100 users by default; there is no separate group created for every new user.

The file /etc/login.defs contains all the defaults for password setting, login retries, umask and path settings. The file /etc/shadow contains encrypted passwords.

Using YaST, we go to "Security Users" (or use the YaST shortcut "users", as shown in Figure 3-2). YaST automatically creates a home directory for a new user.



Linux is a **multiuser system**. Several different users can be logged in the system at the same time. To avoid confusion, each user must have a unique identity if they want to use Linux. Furthermore, every user at least belongs to one group.

In this dialog, you can get information about existing **users**.

To shift to the group dialog push the radio button **Groups administration**.

To create a new user, push the button **Add**.

To edit or delete an existing user, select one user from the list and click **Edit** or **Delete**.

To save the modified user and group setting to your system push the button **Finish**.

**User and group administration**

● Users administration ○ Groups administration

| Login | Name | UID | Groups |
|---|---|---|---|
| admin | forname surname | 500 | users,uucp,dialout,audio,video |
| prabs | prabs | 1200 | users |
| francois | francois | 1201 | users |
| xinghong | xinghong | 1202 | users |
| lee | lee | 1203 | users |
| baublys | baublys | 1204 | users |
| baublys | baublys | 1209 | users |
| tot44 | tot44 | 1300 | users |
| bonnie |  | 1301 | users |

□ Also view system users

*Figure 3-2 YaST user management*

If you want to define user roles (for example printer administrator, database administrator, or user administrator), use user groups. For more special privileges, use **sudo**, as described in 3.5.2, "Using sudo" on page 108.

## 3.5.1 User distribution

We discuss centralized user management with LDAP in 3.16, "OpenLDAP implementation" on page 153. However, another alternative for distributing users in the network is Network Information Service (NIS).[9]

But what if using LDAP and NIS is overkill for our needs? The simplest solution is to manage users locally and distribute the files to other machines.

Using **rsync** or **rdist** can be a very fast and reliable solution. The main disadvantage is that users should only change their passwords on the central server, because changing passwords on other nodes will either lead to inconsistencies, or the passwords will be overwritten during the next file distribution.

The main advantage is that there are no compatibility issues for any applications, because all settings are local. Once distributed, this solution stays functional even if the network connection to LDAP or the NIS server fails.

If you want to manage users locally and distribute them to other servers, you need at a minimum to distribute the following files:

- /etc/passwd
- /etc/shadow
- /etc/gshadow
- /etc/group

If you make any changes to the /etc/skel directory or to /etc/login.defs, you will need to distribute them, as well. In addition, home directories need to be exported through NFS or the automounter, or you can create them locally on each server.

### Using rdist for user distribution

The rdist program maintains identical copies of files over multiple hosts. It preserves the owner, group, mode, and mtime of files if possible, and can update programs that are executing.[10]

rdist is called on a master server (management server). It contacts the target nodes through ssh and starts a rdistd daemon on them. The master server and the nodes stats all the files in the configuration file (distfile). If the stat results are different, then files will be copied from the master server to the nodes and the rdistd daemon terminates.

[9] Find more information and how-tos for NIS at: http://www.linux-nis.org/nis/

[10] See man rdist

In our case, we create a very simple distfile, shown in Example 3-3, for distribution of user management-related files and the /etc/skel directory (on SuSE).

*Example 3-3 /root/distfile example*

```
HOSTS = ( lpar3 lpar4 lpar5 lpar6 )
FILES = ( /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/sudoers
/etc/resolv.conf)
${FILES} -> ${HOSTS}
                install -savetargets ;
```

To avoid being asked for the root password on every LPAR when we run rdist, we enable passwordless key authentication by exchanging ssh keys as described in 3.12.1, “Exchanging ssh keys” on page 144.

Any changes in this file on nodes will be overwritten. In our configuration, rdist will keep one .OLD copy of the overwritten files. If we are going to use rdist to distribute user management-related files, password changes should be only allowed on the management server. We run the **rdist** command on our management server, as shown in Example 3-4. The option **-P** is followed by the complete path of the **ssh** command:

*Example 3-4 Running the rdist command*

```
# rdist -v -P /usr/bin/ssh
lpar3: updating host lpar3
lpar3: /etc/passwd: need to install
lpar3: /etc/gshadow: need to update
lpar3: /etc/sudoers: need to install
lpar3: /etc/resolv.conf: need to update
lpar3: updating of lpar3 finished
```

We can run rdist as a cron job, or run it every time we make changes to any of the files in our distribution list.

For more information, see the homepage of the rdist project at:

`http://www.magnicomp.com/rdist/`

A useful introduction to rdist can be found at:

`http://www.benedikt-stockebrand.de/rdist-intro.html`

## 3.5.2 Using sudo

The sudoprogram allows normal users to execute predefined tasks which usually can only be executed by root. It can be used for improving overall system security; administration task delegation with **sudo** does not mean giving the root password away.

sudo is a setuid binary that is owned by root and can be executed by other users:

```
# ls -al /usr/bin/sudo
-rwsr-xr-x    1 root     root        95296 Nov 11  2002 /usr/bin/sudo
```

If users run **sudo**, they run it with root rights, or with the rights of a specified target user. The configuration for **sudo** is done in the /etc/sudoers file. Do not edit this file manually; instead, use **visudo** to edit it, as shown in Example 3-5 on page 109.

### Making user root equivalent

In our first example, we use a very simple scenario: granting all rights to a user we trust. This is more secure than giving away a root password.

*Example 3-5 Granting all root rights to user*

```
# visudo

# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers file.
#

# Host alias specification

# User alias specification

# Cmnd alias specification

# Defaults specification

# User privilege specification
root            ALL=(ALL) ALL
brockhaus       ALL=(ALL) ALL

# Uncomment to allow people in group wheel to run all commands
# %wheel        ALL=(ALL)       ALL

# Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
```

```
"/etc/sudoers.tmp" 29L, 605C                            17,6-16       All
```

We add one line:

```
   brockhaus       ALL=(ALL) ALL
```

We write and quit the visudo program with wq.

Now we can log in as user brockhaus and test our new granted rights, as shown in Example 3-6 on page 110. Note that user does not gets root path settings automatically, so we need to provide the full path to the command.

*Example 3-6   Using sudo*

```
brockhaus@p630sles:~> sudo /sbin/ifconfig eth0

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:

        #1) Respect the privacy of others.
        #2) Think before you type.

Password:
eth0      Link encap:Ethernet  HWaddr 00:02:55:4F:60:8A
          inet addr:192.168.100.110  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::202:55ff:fe4f:608a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5646864 errors:318 dropped:0 overruns:0 frame:318
          TX packets:8033920 errors:171 dropped:0 overruns:1 carrier:171
          collisions:824271 txqueuelen:100
          RX bytes:3259464842 (3108.4 Mb)  TX bytes:8050769484 (7677.8 Mb)
          Interrupt:101 Base address:0xec00 Memory:88030000-88030038
```

If we look at /var/log/messages, we see the following entry:

```
Nov 10 20:25:18 p630sles sudo: brockhau : TTY=pts/2 ; PWD=/home/brockhaus ;
USER=root ; COMMAND=/sbin/ifconfig eth0
```

The first time this user issues a sudo something command, he will be asked for a password (his own password, not root's). If he uses sudo again during the next 5 minutes (which is the default, and can be changed in the sudoers file), he will not be asked for the password.

Even so, every time the user needs to issue a command with root rights, the user will need to use sudo in front. .

In our case, we first change the default 5-minute password timeout to zero (0) in order force users to enter the password each time they use sudo. We also add the following line:

```
Defaults        timestamp_timeout=0
```

If we set timeout to -1, it will never expire, not even if the user logs out. If we use the following setting, then it will only apply to user brockhaus:

```
Defaults:brockhaus timestamp_timeout=0
```

**Tip:** Experienced Linux administrators never login or work as root. Use sudo !

### Delegating tasks

In our second example, we allow a group of database administrators DBADM to kill database processes, and a group of user administrators USRADM to add and delete users. We change the default policy for database administrators, so they do not need to supply a password when issuing the sudo command.

The result is equivalent to defining user roles in some other UNIX systems. In Example 3-7, we use variables in order to make our sudoer’s file easy extensible.

*Example 3-7 Delegating tasks with sudo*

```
#
# This file MUST be edited with the 'visudo' command as root.
#

# User alias specification
User_Alias      DBADMINS = marju, mathias
User_Alias      USRADMINS = karla, pablo

# Defaults specification
Defaults        timestamp_timeout=0
Defaults:DBADMINS timestamp_timeout=3

# Run as alias specification
Runas_Alias     DB = db2, dasusr1
Runas_Alias     OP = root

# Cmnd alias specification
Cmnd_Alias      KILL=/bin/kill
Cmnd_Alias      USRM=/usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/chage

# User privilege specification
root            ALL=(ALL) ALL
brockhaus       ALL=(ALL) ALL
```

```
DBADMINS        ALL=(DB) NOPASSWD: KILL
USRADMINS       ALL=(root) USRM
```

Now we can login as user marju and use the **sudo -l** command to see what rights we are granted, as shown in Example 3-8

*Example 3-8 sudo -l*

```
marju@p630sles:/root> sudo -l
User marju may run the following commands on this host:
    (db2, dasusr1) /bin/kill
```

As you see, we can use the **kill** command to kill jobs running with a db2 or dasusr1 id, as shown in Example 3-9. We do not need to provide any password, because we used the **NOPASSWD** option in /etc/sudoers.

*Example 3-9 Using sudo as user dasusr1*

```
marju@p630sles:/tmp> ps auxww|grep db2
dasusr1  16244  0.0  0.2 21932 4312 ?       S    23:24   0:00
/home/dasusr1/das/adm/db2dasrrm

marju@p630sles:/tmp> sudo -u dasusr1 kill -9 16245
marju@p630sles:/tmp>
```

More information on sudo and a more sophisticated example file can be found on the sudo mainpage at:

http://www.courtesan.com/sudo

You can also see the available parameters by issuing the **sudo -L** command as root. The command **sudo -V** issued as root will also show the version and which settings are in effect.

Using sudo for task delegation can also be achieved by allowing some users to execute all commands in, for example, the /usr/local/admin directory, and then putting the commands we want delegate in this directory.

> **Important:** Use the **ALL=(ALL) ALL** option only for yourself or someone you really trust. This user will always be able to become root by using su, start, or span root shell in vi mode.

Keeping the security issues in mind when using sudo is better than giving the root password away or working as root. Using sudo prevents commands from being executed as root by accident; every time you want to run something as root, you need to place sudo in front of the command and provide your password.

With sudo, if the root password needs to be changed, you do not need to inform all other administrators. The sudo logging allows you to determine who has issued which command, and when they issued it. Additional security can be achieved by using remote logging on another server.

# 3.6 Logical Volume Manager (LVM)

Linux traditionally handles storage by dividing the physical discs into partitions, formatting partitions in order to create file systems on them, and mounting those under the provided mount point.

A home file system mounted under /home resides, for example, on /dev/sda3, which is a device pointing to the third partition on the first scsi disk recognized by the system. Although this traditional storage handling works fine, it does not provide any flexibility. If you need to resize a partition or change the physical disk, all data must be backed up and restored after the change.

## 3.6.1 What is Logical Volume Manager

Logical Volume Manager (LVM) is an abstraction layer over the physical storage entities. It hides the storage physics from an operating system by putting them (or parts of them), together in a logical volume group.

Inside the logical volume group, LVM creates logical volumes (LVs), which can be used as if they were physical partitions. These logical volumes can be formatted and mounted as if they were partitions on the physical disks.[11]

## 3.6.2 How LVM works

Physical disks are divided in slices called *physical extents* (PE) (or *physical partitions*, in AIX terms). These slices are all equal in size inside one volume group and one physical volume. The default size is 4 MB, but if you are going to create logical volumes bigger than 256 GB, you should use a larger extent size. (For example, a physical extent size of 512 MB will limit a logical volume size to 32 Terabyte.)

A volume group (VG) contains a number of physical extents, which can mean it contains a number of physical disks or partitions or LUNs on remote storage—or

[11] For detailed information on LVM, see LVM How-to at: http://tldp.org/HOWTO/LVM-HOWTO, and the LVM white paper written by Michael Hasenstein at: http://www.suse.com/en/business/certifications/certified_software/oracle/docs/lvm_whitepaper.pdf . The home page of LVM is http://www.sistina.com/products_lvm.htm

even a mixture of all of them. A volume group can be considered as a logical drive.

A logical volume is a defined part of a volume group; it is equivalent to a partition on a single drive. A logical volume can be formatted and mounted by an operating system.

Compared to a traditional partition, a logical volume provides more flexibility; it can be resized or even moved to another volume group. A logical volume cannot span multiple volume groups. A logical volume consists of logical extents which are mapped 1:1 to the physical extents.

### 3.6.3 LVM terminology

Table 3-1 compares some LVM terminology to its traditional and AIX LVM counterparts.

*Table 3-1 LVM terminology comparison*

| Term | Traditional partitions | Logical Volume Manager | AiX LVM |
|---|---|---|---|
| Physical disks | Physical disks are referenced as devices and can be divided into partitions. | Physical disks are not visible; all partitions of the Type “Linux LVM” can be used as space for Volume Group, not dependent on their physical location or type. | pdisk. |
| Volume group | From the OS view, a VG is the equivalent of the physical disks. | VG contains physical extents and is divided in logical volumes. | Volume group contains pdisks. |
| Logical volume | Partition on the physical disk. | Logical volume resides in the VG and is visible as a standard block device. It can be formatted and mounted. | Logical volume. |

### 3.6.4 Setting up LVM

Setting up LVM always includes three main steps:

1. Initializing physical volumes (these can be disks, partitions, or RAID)
2. Defining volume groups.
3. Setting up logical volumes inside the volume croups.

### 3.6.5 Using the command line to create LVM

> **Tip:** Using whole disk for LVM requires that there is no partition table on the disk, not even an empty one. Since we cannot use YaST2 to assign a whole disk to the LVM, we use the command line.

First we need to overwrite the partition table:

```
# dd if=/dev/zero of=/dev/sdb bc=1k count=1
1+0 records in
1+0 records out
```

We reread the partition table in the kernel:

```
# blockdev --rereadpt /dev/sdb
```

We initialize the physical volume as shown in Example 3-10:

*Example 3-10 Initializing the physical volume*

```
# pvcreate /dev/sdb
pvcreate -- physical volume "/dev/sdb" successfully created

# lvmdiskscan
lvmdiskscan -- reading all disks / partitions (this may take a while...)
lvmdiskscan -- /dev/sda1   [       7.81 MB] Primary   [0x41]
lvmdiskscan -- /dev/sda2   [     101.97 MB] Primary  LINUX native partition
[0x83]
lvmdiskscan -- /dev/sda3   [       4.01 GB] Primary  LINUX native partition
[0x83]
lvmdiskscan -- /dev/sda5   [       2.01 GB] Extended LINUX swap partition
[0x82]
lvmdiskscan -- /dev/sda6   [     130.60 GB] Extended LVM partition [0x8E]
lvmdiskscan -- /dev/sdb    [     136.73 GB] new LVM whole disk
lvmdiskscan -- 2 disks
lvmdiskscan -- 1 whole disk
lvmdiskscan -- 0 loop devices
lvmdiskscan -- 0 multiple devices
lvmdiskscan -- 0 network block devices
lvmdiskscan -- 5 partitions
lvmdiskscan -- 1 LVM physical volume partition
```

Next, we create a volume group on /dev/sdb, as shown in Example 3-11.

*Example 3-11 Creating a volume group*

```
# vgcreate datavg /dev/sdb
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "datavg"
vgcreate -- volume group "datavg" successfully created and activated
```

We create a logical volume in the datavg:

```
# lvcreate -L 3000M -n data1-lv datavg
lvcreate -- doing automatic backup of "datavg"
lvcreate -- logical volume "/dev/datavg/data1-lv" successfully created
```

We format it:

```
# mkreiserfs /dev/datavg/data1-lv
```

And mount it:

```
# mkdir /data1
# mount /dev/datavg/data1-lv /data1
```

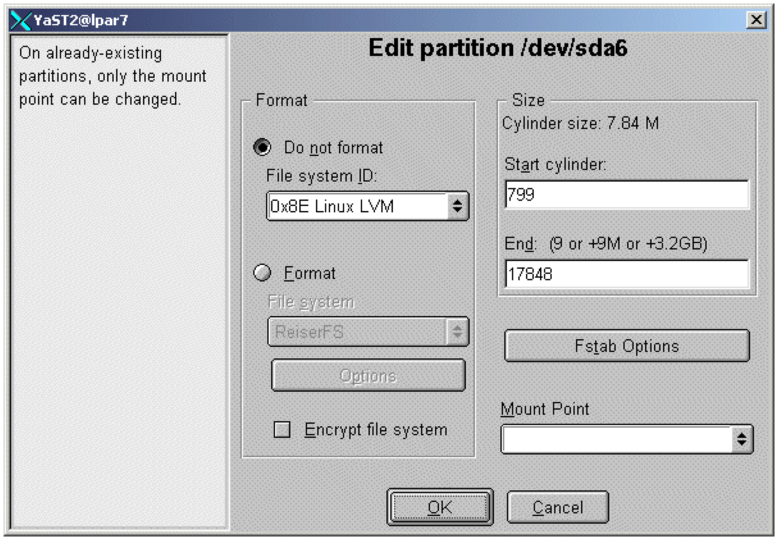
At this point, our datavg resides only on one physical volume, /dev/sdb. Since we have space on /dev/sda, we created partition type 8e and add this partition to our volume group:

```
# vgextend datavg /dev/sda6
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "datavg"
vgextend -- volume group "datavg" successfully extended
```

### 3.6.6 Using YaST for LVM

To create partitions that can be used for LVM, follow these steps:

1. Start YaST2, then go to **System** -> **Partitioning** -> **Create** -> **Create new partition**, Type LVM, and then select **Do not format**, as shown in Figure 3-3 on page 117.


YaST2@lpar7

Edit partition /dev/sda6

On already-existing partitions, only the mount point can be changed.

Format

● Do not format

File system ID: 0x8E Linux LVM

○ Format

File system: ReiserFS

□ Encrypt file system

Size

Cylinder size: 7.84 M

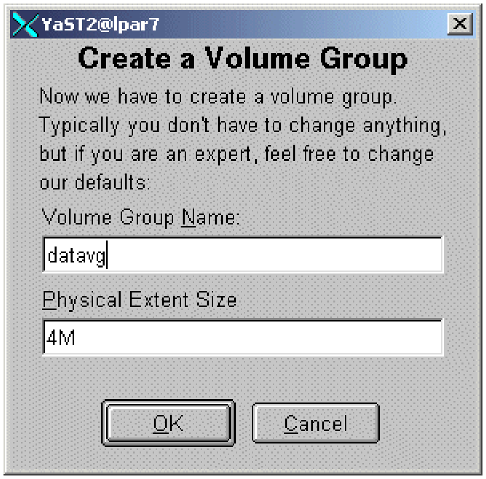
Start cylinder: 799

End: (9 or +9M or +3.2GB) 17848

Mount Point:

*Figure 3-3 Creating a partition for LVM*

2. Now we can press OK to create a volume group; see Figure 3-4.


YaST2@lpar7

Create a Volume Group

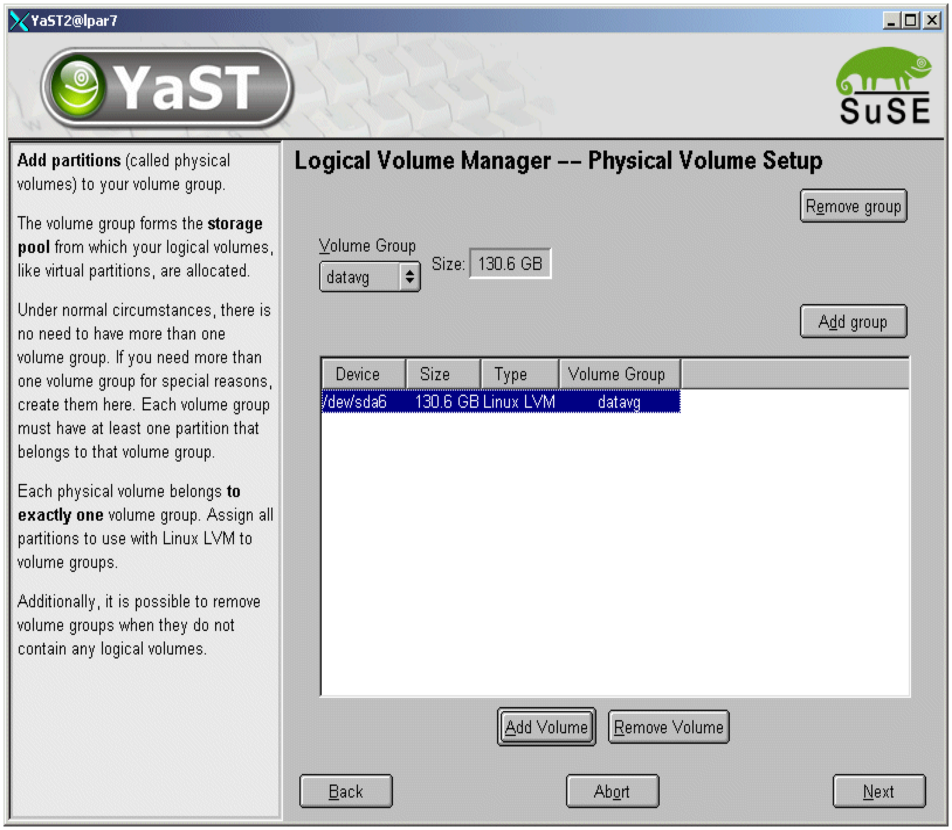
Now we have to create a volume group. Typically you don't have to change anything, but if you are an expert, feel free to change our defaults:

Volume Group Name: datavg

Physical Extent Size: 4M

*Figure 3-4 Creating a volume group*

Now we add our physical volume (previously defined partition) to the volume group pressing the **Add Volume** button, as shown in Figure 3-5 on page 118.


YaST2@lpar7

**Add partitions** (called physical volumes) to your volume group.

The volume group forms the **storage pool** from which your logical volumes, like virtual partitions, are allocated.

Under normal circumstances, there is no need to have more than one volume group. If you need more than one volume group for special reasons, create them here. Each volume group must have at least one partition that belongs to that volume group.

Each physical volume belongs **to exactly one** volume group. Assign all partitions to use with Linux LVM to volume groups.

Additionally, it is possible to remove volume groups when they do not contain any logical volumes.

Logical Volume Manager -- Physical Volume Setup

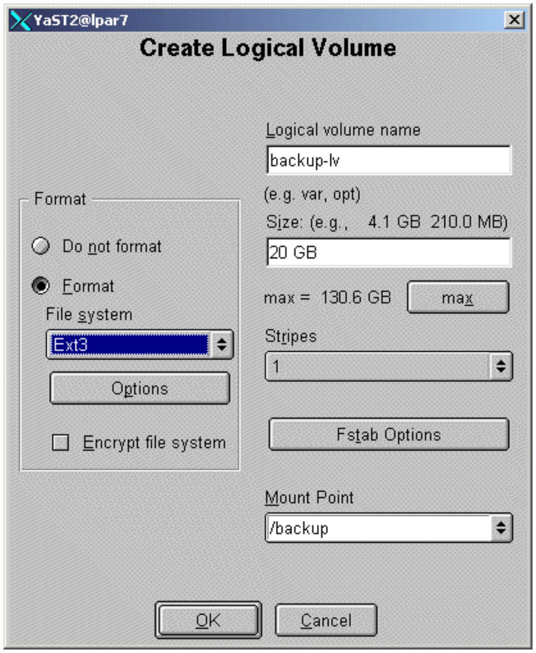
Volume Group: datavg

Size: 130.6 GB

| Device | Size | Type | Volume Group |
|---|---|---|---|
| /dev/sda6 | 130.6 GB | Linux LVM | datavg |

*Figure 3-5 Adding physical volume to the volume group*

The last step is to configure logical volumes in the volume group as shown in Figure 3-6



Logical volume name: backup-lv
(e.g. var, opt)
Size: (e.g., 4.1 GB 210.0 MB): 20 GB
max = 130.6 GB
Stripes: 1
Mount Point: /backup

Format
○ Do not format
● Format
File system: Ext3
□ Encrypt file system

*Figure 3-6 Creating a logical volume*

## 3.6.7 LVM features

In this section, we discuss the advantages of using Logical Volume Manager.

### Resizing

LVM volumes can be easily resized using **lvextend** and **lvreduce** commands. These commands change the size of the logical volume only, not the size of the file system in it.

n the following example we resize the volume, but the size of the file system does not increase:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/datavg/data1-lv  3.0G   33M  2.9G   2% /data1

# lvextend -L  +1000M /dev/datavg/data1-lv
lvextend -- extending logical volume "/dev/datavg/data1-lv" to 3.91 GB
lvextend -- doing automatic backup of volume group "datavg"
lvextend -- logical volume "/dev/datavg/data1-lv" successfully extended
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/datavg/data1-lv  3.0G   33M  2.9G   2% /data1
```

In order to increase the file system we need different tools for different file system, for reiserfs we use resize_reiserfs:

```
# resize_reiserfs -s +1000M /dev/datavg/data1-lv
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/datavg/data1-lv  4.0G   33M  3.9G   1% /data1
```

**Note:** Reiserfs and Ext2/3 file system both can be increased on the fly, but decreasing file system requires umount.

If you are using YaST2 to increase the size of the logical volume, it will increase the size of the file system automatically.

If you are using command line utilities, you always need to increase the logical volume size *prior* to increasing the file system size, and to decrease the file system size *prior* to decreasing logical volume size.

### Replacing hardware

In order to replace hardware, LVM is able to move all used physical extents (PE) to other disks. The command to do this is **pvmove**. If you provide only a disk which should be freed up as an argument, **pvmove** will decide by itself where to move data. In this way, exchanging hardware can be achieved without any interruption.

**Attention:** Back up the system before using **pvmove**, because if the system crashes during the process, it can be a problem. Moving striped logical volumes should not be interrupted.

### Making snapshots

A snapshot is a frozen, read only image of the logical volume. This is useful, for example, for providing a read only volume for backups. A snapshot is frozen at the moment of creation, but the real logical volume can be used and will be changed.

The snapshot logical volume must have enough space to keep the changes on the real LV during the lifetime of the snapshot.

```
# lvcreate -L 1024M -s -n snap-data1-lv /dev/datavg/data1-lv
lvcreate -- WARNING: the snapshot will be automatically disabled once it gets
full
lvcreate -- INFO: using default snapshot chunk size of 64 KB for
"/dev/datavg/snap-data1-lv"
lvcreate -- doing automatic backup of "datavg"
lvcreate -- logical volume "/dev/datavg/snap-data1-lv" successfully created
```

Now we mount the snapshot volume:

```
# mount /dev/datavg/snap-data1-lv /mnt/snapshot
mount: block device /dev/datavg/snap-data1-lv is write-protected, mounting
read-only
```

And after back up, we umount and delete it:

```
# umount /mnt/snapshot
# lvremove /dev/datavg/snap-data1-lv
lvremove -- do you really want to remove "/dev/datavg/snap-data1-lv"? [y/n]: y
lvremove -- doing automatic backup of volume group "datavg"
lvremove -- logical volume "/dev/datavg/snap-data1-lv" successfully removed
```

### Striping for performance

Although mirroring is not possible through LVM, it is possible to use software RAID devices. Striping or RAID 0 can be implemented through LVM using completely different storage systems. By default, LVM uses concatenation, which means it uses the next free place.

However, it is possible to change it for striping, which means it will use all drives in an alternating pattern, thus achieving better performance. In our testing environment we achieved nearly doubled the writing performance using two disks in LVM striping mode (Raid 0). See 3.7, “Software mirroring and LVM” on page 122 for a detailed description how to utilize mirroring with LVM.

### LVM commands

Table 3-2 lists LVM commands.

*Table 3-2 LVM command list*

| LVM command | Purpose | Example |
|---|---|---|
| `lvmdiskscan` | Displays all storage devices | |
| `vgscan` | Scans all physical devices, searches for VGs, and creates /etc/lvmtab and /etc/lvmtab.d | `vgscan` |

| LVM command | Purpose | Example |
|---|---|---|
| pvdata | Displays debugging information about PV, reads VGDA | pvdata /dev/sdb |
| pvscan | Scans PVs and displays active | #pvscan |
| pvcreate | Creates a PV from 8e type partition | |
| vgcreate | Creates VG using PVs | |
| pvmove | Moves data from one PV to another inside one VG | |
| vgreduce | Removes PV from VG | |
| vgdisplay, pvdisplay, lvdisplay | Displays information about VG, PV or LV | |
| vgchange | Activates or deactivates VG | |
| vgexport | Makes VGs unknown to the system, used prior to importing them on a different system | |
| vgimport | Imports VG from a different system | |
| vgsplit | Splits PV from existing VG into new one | |
| vgmerge | Merges two VGs | |
| lvcreate | Creates LV inside VG | |
| lvextend | Increases the size of LV | |
| lvreduce | Decreases the size of LV | |
| lvdisplay | Shows attributes of LV | |

## 3.7 Software mirroring and LVM

Linux LVM does not provide mirroring functionality by itself. However, it can use mirroring achieved through a hardware RAID controller or through software.

More information on Linux software RAID can be found at:

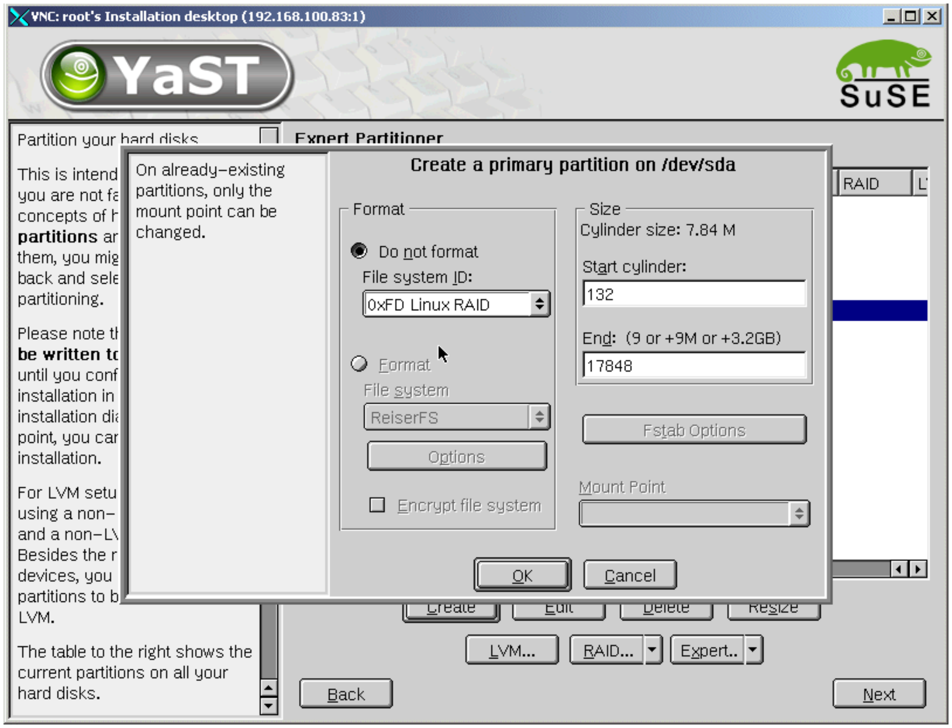
```
http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html
```

Currently, Linux supports linear md devices, RAID0 (striping), RAID1 (mirroring), and RAID4 and RAID5. Because we can use LVM for linear device and striping, and do not have enough disks for RAID5, we are going to create RAID1, a software mirror.

## 3.7.1 Using YaST to create RAID

From the LVM point of view, a RAID device (/dev/md0) is just storage, so we need to create mirrored disks or partitions first and place an LVM volume group on it. This can be achieved by YaST, as shown in Example 3-5.

First, we create two partitions which we will use for our RAID, as shown in Figure 3-7. We need at least two because we want to mirror them, which only makes sense when the partitions reside on different physical hard disks.



This is intend
you are not fa
concepts of h
**partitions** ar
them, you mig
back and sele
partitioning.

Please note th
**be written to**
until you conf
installation in
installation dia
point, you car
installation.

For LVM setu
using a non-
and a non-LV
Besides the r
devices, you
partitions to b
LVM.

The table to the right shows the current partitions on all your hard disks.

On already-existing partitions, only the mount point can be changed.

**Create a primary partition on /dev/sda**

Format
- ● Do not format
  - File system ID: 0xFD Linux RAID
- ○ Format
  - File system: ReiserFS
  - Options
  - □ Encrypt file system

Size
- Cylinder size: 7.84 M
- Start cylinder: 132
- End: (9 or +9M or +3.2GB) 17848

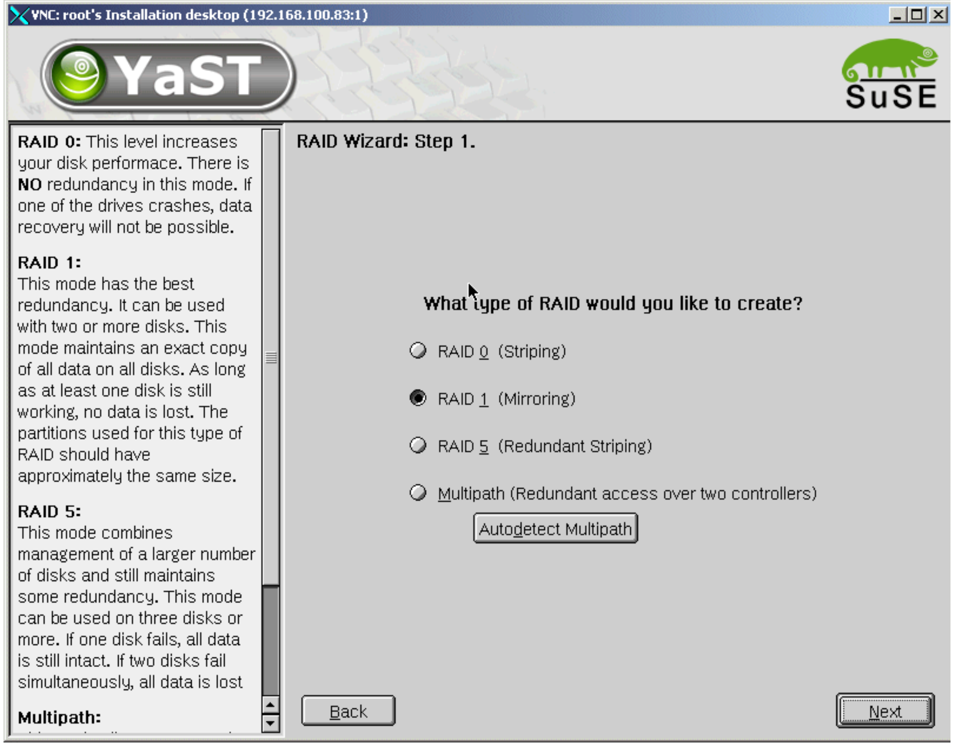
Fstab Options

Mount Point:

OK | Cancel

*Figure 3-7 Creating a RAID device with YaST, step1*

Next, we choose the RAID level we want to use, as shown in Figure 3-8. For mirroring, we choose RAID1. We do not recommend using striping through RAID, because you can achieve the same result by using LVM.



**RAID 0:** This level increases your disk performace. There is **NO** redundancy in this mode. If one of the drives crashes, data recovery will not be possible.

**RAID 1:**
This mode has the best redundancy. It can be used with two or more disks. This mode maintains an exact copy of all data on all disks. As long as at least one disk is still working, no data is lost. The partitions used for this type of RAID should have approximately the same size.

**RAID 5:**
This mode combines management of a larger number of disks and still maintains some redundancy. This mode can be used on three disks or more. If one disk fails, all data is still intact. If two disks fail simultaneously, all data is lost

**Multipath:**

**RAID Wizard: Step 1.**

**What type of RAID would you like to create?**

○ RAID 0 (Striping)

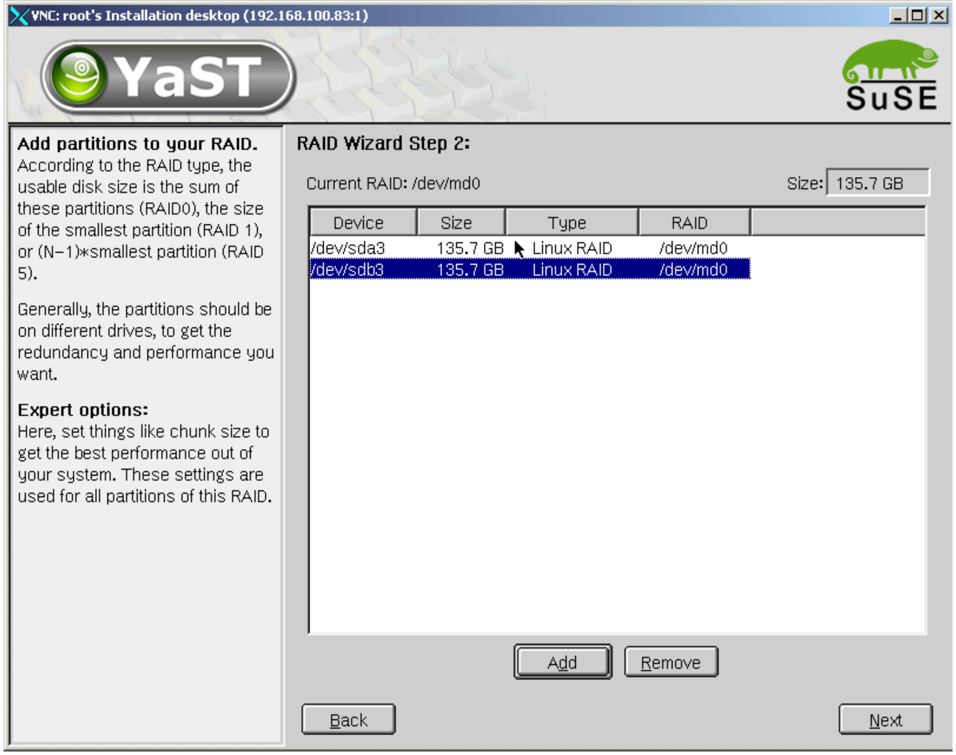
● RAID 1 (Mirroring)

○ RAID 5 (Redundant Striping)

○ Multipath (Redundant access over two controllers)

*Figure 3-8 Choosing desired RAID level*

Now we can assign our RAID partitions, /dev/sda3 and /dev/sdb3, to a newly created RAID device, /dev/md0 ,as shown in Figure 3-9 on page 125.


VNC: root's Installation desktop (192.168.100.83:1)

**Add partitions to your RAID.** According to the RAID type, the usable disk size is the sum of these partitions (RAID0), the size of the smallest partition (RAID 1), or (N−1)*smallest partition (RAID 5).

Generally, the partitions should be on different drives, to get the redundancy and performance you want.

**Expert options:** Here, set things like chunk size to get the best performance out of your system. These settings are used for all partitions of this RAID.

**RAID Wizard Step 2:**

Current RAID: /dev/md0

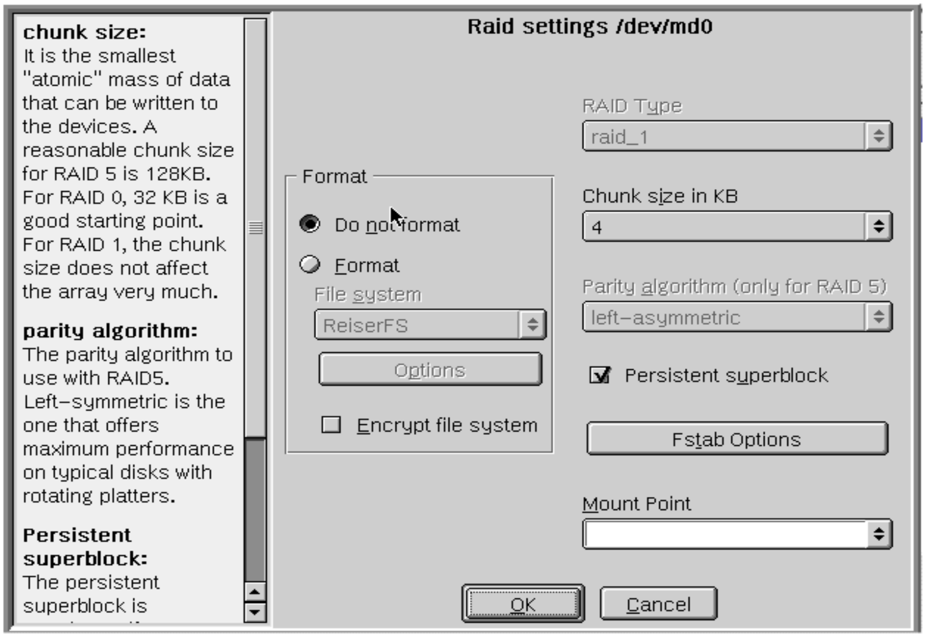
Size: 135.7 GB

| Device | Size | Type | RAID |
|---|---|---|---|
| /dev/sda3 | 135.7 GB | Linux RAID | /dev/md0 |
| /dev/sdb3 | 135.7 GB | Linux RAID | /dev/md0 |

*Figure 3-9 Assigning RAID type partitions in order to create a device*

In order to be able to use our mirrored device for LVM, we are not going to format it; see Figure 3-10.

However, it is possible to format and mount the mirrored device directly.



**chunk size:**
It is the smallest "atomic" mass of data that can be written to the devices. A reasonable chunk size for RAID 5 is 128KB. For RAID 0, 32 KB is a good starting point. For RAID 1, the chunk size does not affect the array very much.

**parity algorithm:**
The parity algorithm to use with RAID5. Left-symmetric is the one that offers maximum performance on typical disks with rotating platters.

**Persistent superblock:**
The persistent superblock is

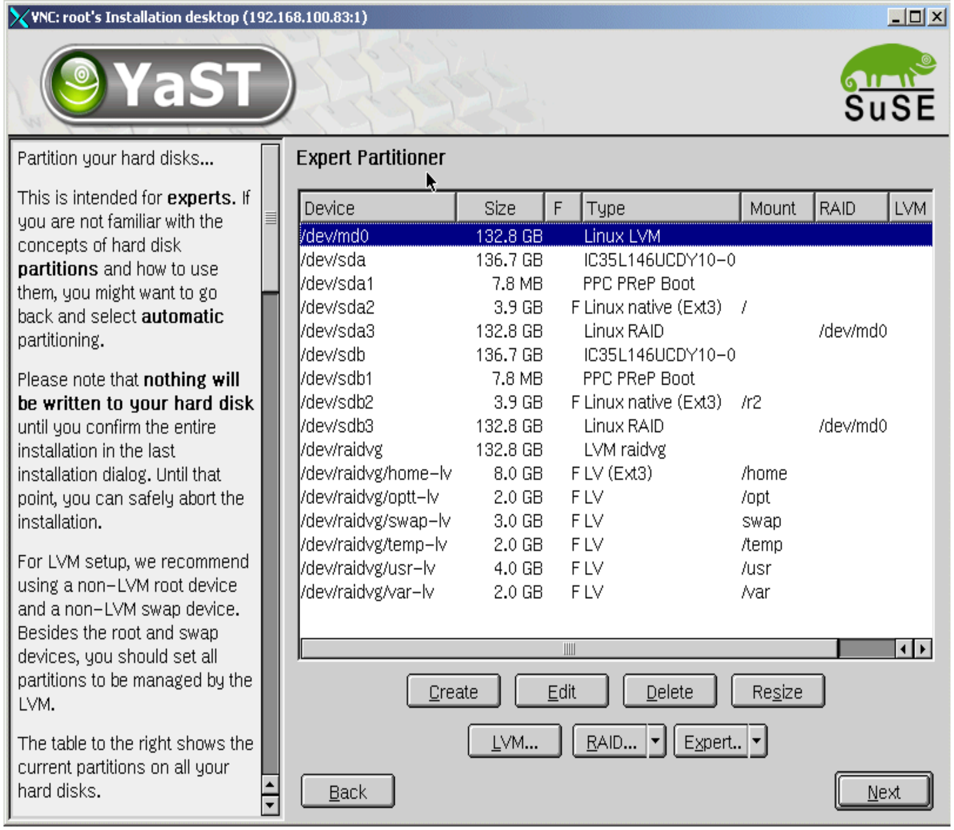
**Raid settings /dev/md0**

Format
● Do not format
○ Format
File system: ReiserFS
□ Encrypt file system

RAID Type: raid_1
Chunk size in KB: 4
Parity algorithm (only for RAID 5): left-asymmetric
☑ Persistent superblock
Mount Point:

*Figure 3-10 No format is chosen*

Now we can use /dev/md0 as a normal storage device. We can create an LVM volume group on it and LVM volumes in it, as shown in Figure 3-11 on page 127.



Partition your hard disks...

This is intended for **experts.** If you are not familiar with the concepts of hard disk **partitions** and how to use them, you might want to go back and select **automatic** partitioning.

Please note that **nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

For LVM setup, we recommend using a non-LVM root device and a non-LVM swap device. Besides the root and swap devices, you should set all partitions to be managed by the LVM.

The table to the right shows the current partitions on all your hard disks.

Expert Partitioner

| Device | Size | F | Type | Mount | RAID | LVM |
|---|---|---|---|---|---|---|
| /dev/md0 | 132.8 GB | | Linux LVM | | | |
| /dev/sda | 136.7 GB | | IC35L146UCDY10-0 | | | |
| /dev/sda1 | 7.8 MB | | PPC PReP Boot | | | |
| /dev/sda2 | 3.9 GB | F | Linux native (Ext3) | / | | |
| /dev/sda3 | 132.8 GB | | Linux RAID | | /dev/md0 | |
| /dev/sdb | 136.7 GB | | IC35L146UCDY10-0 | | | |
| /dev/sdb1 | 7.8 MB | | PPC PReP Boot | | | |
| /dev/sdb2 | 3.9 GB | F | Linux native (Ext3) | /r2 | | |
| /dev/sdb3 | 132.8 GB | | Linux RAID | | /dev/md0 | |
| /dev/raidvg | 132.8 GB | | LVM raidvg | | | |
| /dev/raidvg/home-lv | 8.0 GB | F | LV (Ext3) | /home | | |
| /dev/raidvg/optt-lv | 2.0 GB | F | LV | /opt | | |
| /dev/raidvg/swap-lv | 3.0 GB | F | LV | swap | | |
| /dev/raidvg/temp-lv | 2.0 GB | F | LV | /temp | | |
| /dev/raidvg/usr-lv | 4.0 GB | F | LV | /usr | | |
| /dev/raidvg/var-lv | 2.0 GB | F | LV | /var | | |

*Figure 3-11 Placing a volume group and volumes on raid device /dev/md0*

**Restriction:** We do not recommend using software RAID or LVM for the root and the boot partition. In our testing, although we were able to configure it, the system hung on the reboot.

## 3.7.2 Using the command line to create RAID

In Example 3-12 on page 128 we have two disks; a small part of the first is used for / partition and a swap device, and the second disk is empty.

We can create a logical partition on our first disk and mirror it to the partition on the second disk. For better compatibility and performance, we choose to span identical cylinders.

*Example 3-12 Starting point software RAID*

```
# fdisk -l

Disk /dev/sda: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1         1      8001   41  PPC PReP Boot
/dev/sda3            15       537   4200997+  83  Linux
/dev/sda4           538     17848 139050607+   5  Extended
/dev/sda5           538       799   2104483+  82  Linux swap

Disk /dev/sdb: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
```

First we create a RAID partition on the first disk (we type: `fdisk /dev/sda`, `n`, `l`, Enter, Enter, `t`, `6`, `fd`).

- n - new
- l - logical
- enter - use default starting cylinder
- enter - use default ending cylinder
- t - change type
- 6 - number of partition which type we want change
- fd - Type Linux Software RAID autodetect

*Example 3-13 Creating a RAID partition on the first disk*

```
fdisk /dev/sda

The number of cylinders for this disk is set to 17849.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (800-17848, default 800):
Using default value 800
Last cylinder or +size or +sizeM or +sizeK (800-17848, default 17848):
Using default value 17848
```

```
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): fd
Changed system type of partition 6 to fd (Linux raid autodetect)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

In Example 3-14, we create a RAID partition on the second disk and use 800 as a start cylinder in order to be analog to the first one. We will lose additional space anyway if we are going to mirror the partitions.

*Example 3-14 Creating a RAID partition on the second disk*

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 4
First cylinder (1-17849, default 1): 800
Last cylinder or +size or +sizeM or +sizeK (800-17849, default 17849):
Using default value 17849

Command (m for help): t
Partition number (1-4): 4
Hex code (type L to list codes): fd
Changed system type of partition 4 to fd (Linux raid autodetect)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

Now we have partition sda6 and sdb4 ready for RAID. In order to define what kind of RAID we want to create, we edit /etc/raidtab as shown in Example 3-15.

*Example 3-15 /etc/raidtab file*

```
raiddev /dev/md0
        raid-level   raid1
        nr-raid-disks   2
        chunk-size      32
        persistent-superblock 1
        device          /dev/sda6
        raid-disk       0
        device          /dev/sdb4
        raid-disk       1
```

Now we run the **mkraid** command in order to create a RAID device defined in /etc/raidtab, as shown in Example 3-16.

*Example 3-16 mkraid*

```
#  mkraid /dev/md0
handling MD device /dev/md0
analyzing super-block
disk 0: /dev/sda6, 136946061kB, raid superblock at 136945984kB
disk 1: /dev/sdb3, 136954125kB, raid superblock at 136954048kB
```

We can watch the status of our RAID device by issuing the **cat /proc/mdstat** command, as shown in Example 3-17.

*Example 3-17 Checking RAID status*

```
# cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 sdb3[1] sda6[0]
      136945984 blocks [2/2] [UU]
      [>....................]  resync =  1.8% (2538560/136945984)
finish=140.7min speed=15917K/sec
unused devices: <none>
```

**Note:** We do not need to wait for reconstruction to finish in order to use the RAID device; the synchronization is done using idle I/O bandwidth. The process is transparent, so we can use the device (place LVM on it, partition and mount) although the disks are not synchronized yet. If one disk fails during the synchronization, we will need our backup tape.

Now we can create a volume group and add /dev/md0 to it, as shown in Example 3-18.

*Example 3-18 Creating VG on RAID device*

```
# vgcreate raidvg /dev/md0
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "raidvg"
vgcreate -- volume group "raidvg" successfully created and activated
```

And a logical volume in this volume group, as shown in Example 3-19.

*Example 3-19 Creating LV in raidvg*

```
lvcreate  -L 20G -n mirrordata1 raidvg
lvcreate -- doing automatic backup of "raidvg"
lvcreate -- logical volume "/dev/raidvg/mirrordata1" successfully created
```

This newly created volume can be formatted and mounted.

## 3.8 File systems

In this section, we introduce the choice of file systems available using Linux on pseries and their basic characteristics. For more detailed information, you can refer to “Advanced file system implementor's guide” written by Daniel Robbins and published at:

`http://www.ibm.com/developerworks/linux/library/l-fs.html`

You can also refer to an article in Linux Magazine about journaling file systems written by Steve Best at:

`http://www.linux-mag.com/2002-10/jfs_01.html`

An introduction to Linux file systems is also provided in “Filesystems in Linux” in “SuSELinux EnterpriseServer 8 for IBM iSeries and IBM pSeries Installation”, available on the CD or in the United Linux white paper:

`http://www.suse.com/en/business/products/server/sles/misc/ul_whitepaper.pdf`

A single Linux machine can use all file systems at once. On SuSE, we can set up all of them (except xfs) through YaST. Red Hat AS 3.0 only supports ext2, ext3, and JFS.

Some benchmarks comparing the performance of different file systems can be found at:

`http://oregonstate.edu/~kveton/fs/`

## 3.8.1 Second extended file system (ext2)

Extended file system version 2 is a classic Linux file system that offers good performance and reliability, but is not a journaling file system.

A power failure with a system mounted, or a system crash, will cause a dirty bit set and a running file system consistency check (fsck) after reboot. The system cannot be mounted until the consistency check is successfully finished. Fsck on the ext2 file system needs to check all file system structures; the time it takes to accomplish this task depends on the size of the file system and the number of files in it.

For information about limits of the ext2 file system, refer to:

`http://www.unitedlinux.com/pdfs/whitepaper4.pdf`

For information about the design and history of ext2, see:

`http://e2fsprogs.sourceforge.net/ext2intro.html`

## 3.8.2 Third extended file system (ext3)

Extended 3 file system (ext3) is built on an ext2 framework, adding the journaling feature to it. In contrast to ext2, it does not require a file system check after every unclean shutdown.

Although ext3 is a journaling file system, it remains backward-compatible. An ext3 file system can always be explicitly mounted as an ext2 file system. An ext2 can be easily promoted on the fly to the ext3 file system, as shown in Example 3-20, and no reformatting is required.

*Example 3-20 promoting ext2 to ext3*

```
# tune2fs -j -c0 -i0 /dev/datavg/bench-lv
tune2fs 1.28 (31-Aug-2002)
Setting maximal mount count to -1
Setting interval between check 0 seconds
Creating journal inode: done
This filesystem will be automatically checked every -1 mounts or
0 days, whichever comes first.  Use tune2fs -c or -i to override.
```

In order to use journaling, we need to remount the file system with -t ext3 or the auto option. Surprisingly, the performance increases in most cases after promoting ext2 to ext3. Ext3 has higher throughput, because its journaling optimizes hard drive head motion.[12]

Ext3 has multiple journaling modes:

- data=journal > journal all file data and metadata
- data=ordered > journal metadata only, but data integrity ensured
- data=writeback > journal metadata only

The data=journal and data=ordered (default) modes[13] in ext3 provide even better reliability then reiserfs or jfs because it is designed to ensure the integrity of both metadata and data.[14] Journal can be placed on other devices. The limits applying for ext3 are the same as ext2.

ext3 file system is the default file system on the Red Hat AS 3.

**Tip:** We recommend that you use the ext3 file system for system partitions. Because of its combination of journaling feature, recovery tools, and backward compatibility, it is the safest choice.

### 3.8.3 Reiser file system (reiserfs)

Reiserfs, developed and designed by Hans Reiser and his team at Namesys (http://www.namesys.com), was the first journaling file system available for Linux and is the default file system on SuSE SLES.

It uses optimized b* balanced tree, dynamic inode allocation and tail packing. Reiserfs has excellent small file performance and space utilization. Reiserfs is around 8 to 15 times faster than ext2 when handling files smaller than one k.[15]

ReiserFS does meta-data journaling, enabling fast crash recovery without the expense of full data journaling.

For a look at the specifications of reiserfs, see:

```
http://www.namesys.com/faq.html#reiserfsspecs
```

[12] http://www.redhat.com/support/wpapers/redhat/ext3/index.html

[13] SuSE Installation Guide: “A relatively new approach is to use the data=ordered mode, which ensures both data and metadata integrity, but uses journaling only for metadata. The file system driver collects all data blocks that correspond to one metadata update. These blocks are grouped as a “transaction” and will be written to disk before the metadata is updated. As a result, consistency is achieved for metadata and data without sacrificing performance.”

[14] Daniel Robbins article at: http://www-106.ibm.com/developerworks/library/l-fs7.html

[15] http://www-106.ibm.com/developerworks/library/l-fs.html

**Tip:** For a performance tweak, use **`noatime`** and **`notail`** options when mounting a reiser file system.

### 3.8.4 Journaling File system (jfs)

JFS for Linux is a port and further development from the OS/2® jfs file system. It has an OS/2 compatibility option and is also closely related to the JFS2 file system on AIX 5L (but not to the JFS (1) file system on AIX)[16] JFS for linux is distributed under General Public License (GPL).

All system structure fields in JFS are 64-bits in size. JFS does journaling of meta-data only, it can immediately restart after a crash, and uses B+Tree algorithms for large directory structures.

JFS supports dynamic disc inode allocation, extended file attributes (EA) and access control lists (ACL).

JFS is a log-based, byte-level file system with great scalability and restore capabilities. Some JFS limits are shown in Table 3-3.

*Table 3-3 JFS limits*

| JFS | Limits |
|---|---|
| Maximum file size | 4 PB with 4k/block size; 4 TB with 512 block size |
| Maximum file system size | 4 PB with 4k/block size; 4 TB with 512 block size |
| MInimum file system size | 16 Megabyte |
| Number of subdirectories in one directory | 65 K |

For more information on JFS, visit:

```
http://www-124.ibm.com/developerworks/opensource/jfs/
http://www-106.ibm.com/developerworks/library/l-jfs.html
```

## 3.9 RPM management

Red Hat Packet Manager (RPM) is a standard software packaging format in SuSE and all Red Hat-based distributions.[17]

[16] See interview with Steve Best at: http://www.osnews.com/story.php?news_id=69

## 3.9.1 Naming convention

The RPM naming convention has the following format: name-version-release.arch.rpm. If we take screen-3.9.13-23.ppc.rpm as an example, we see the name of the software package: screen, version:3.9.13 and release:23. The last section, ppc, indicates that this RPM is build for the PowerPC plattform.

## 3.9.2 Querying RPM

We have two RPM querying possibilities: querying a package we have not installed yet, and querying an installed software package. Most options are the same for both queries, but the command syntax is different. Some options can only be used for one of the queries.

To query a package that is not installed (an RPM file):

```
rpm -q OPTIONS -p name-version-release.arch.rpm
```

Always add the **-p** option for a package that is not installed, as shown in Example 3-21.

*Example 3-21 rpm querying rpm-file*

```
# rpm -qi -p screen-3.9.13-23.ppc.rpm
Name        : screen                       Relocations: (not relocateable)
Version     : 3.9.13                            Vendor: SuSE Linux AG,
Nuernberg, Germany
Release     : 23                            Build Date: Tue Nov 12 01:38:51
2002
Install date: (not installed)               Build Host: sweetie.suse.de
Group       : System/Console                Source RPM:
screen-3.9.13-23.src.rpm
Size        : 633585                           License: GPL
Packager    : http://www.suse.de/feedback
Summary     : Multi Screen on a VT100/ANSI Terminal
Description :
With this program you can take advantage of the multitasking abilities
of your Linux system by opening several sessions over one terminal.
The sessions can also be detached and resumed from another login
terminal.

Documentation: man page

Authors:
```

[17] For more information on building your own RPMs, see: http://www-106.ibm.com/developerworks/linux/library/l-rpm1/ or Maximum RPM

```
--------
    Oliver Lauman
    Juergen Weigert   <jnweiger@immd4.informatik.uni-erlangen.de>
    Michael Schroeder <mlschroe@immd4.informatik.uni-erlangen.de>
```

To gather information about installed packages we do not provide the full RPM name with version, release and architecture; instead, we only use the name of the software:

```
rpm -q OPTIONS name
```

This is shown in Example 3-22.

*Example 3-22 rpm: querying installed rpm package*

```
# rpm -qi rsync
Name        : rsync                        Relocations: /usr
Version     : 2.5.6                             Vendor: Red Hat, Inc.
Release     : 20                            Build Date: Tue 05 Aug 2003
03:52:53 PM EDT
Install Date: Thu 06 Nov 2003 05:00:36 PM EST      Build Host:
cure81.devel.redhat.com
Group       : Applications/Internet         Source RPM: rsync-2.5.6-20.src.rpm
Size        : 263593                           License: GPL
Signature   : DSA/SHA1, Wed 24 Sep 2003 01:24:57 PM EDT, Key ID
219180cddb42a60e
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : A program for synchronizing files over a network.
Description :
Rsync uses a reliable algorithm to bring remote and host files into
sync very quickly. Rsync is fast because it just sends the differences
in the files over the network instead of sending the complete
files. Rsync is often used as a very powerful mirroring process or
just as a more capable replacement for the rcp command. A technical
report which describes the rsync algorithm is included in this
package.
```

Table 3-4 lists options for querying RPMs; refer to the rpm man page for more information.

*Table 3-4 RPM query options*

| RPM query option | Purpose |
|---|---|
| rpm -q -i | Information: package description, packager, build and install date |
| rpm -q -l | File list |

| RPM query option | Purpose |
|---|---|
| rpm -q -s | File list with state: the state of each file is either normal,  not installed, or replaced. |
| rpm -q -d | List of documentation files. |
| rpm -q -c | List of configuration files. |
| rpm -q --requires | List of packages or files on which this package depends. |
| rpm -q --provides | List what this package provides. |
| --checksig | Check signature (only for package). |
| rpm -qf /filename | Which package installed the provided file. |
| rpm -qilf /filename | Which package installed the provided file, what it does, and which files belong to it. |
| rpm -qdf /filename | Show all documentation installed on the provided file or RPM it belongs to. |
| rpm -qa | List all rpms installed. |
| rpm -Va | Which files in my system are modified missing or replaced (this takes a long time). |
| rpm -Vv name | Which files from the given package are modified or missing? |
| rpm --rebuilddb | Everything is broken, please rebuild the rpm database (backup the /var/lib/rpm directory first!) |

RPM maintains a database in /var/lib/rpm, and this database grows, especially after system updates. It might be helpful to rebuild the database (backup the /var/lib/rpm directory first). On our test system, rpm --rebuilddb decreased the size of the /var/lib/rpm directory from 70 to 22 Megabytes.

To verify and monitor changes in the system, **`rpm -V package`** or **`a`** can be used. It uses the database and displays all files changed since installation for the given package, or for the entire system, if the **`-a`** option is provided, as shown in Example 3-23 on page 138.

*Example 3-23 using rpm -V samba*

```
# rpm -V samba
S.5....T c /etc/pam.d/samba
S.5....T c /etc/samba/smbpasswd
```

RPM displays 8 characters in front of the changed file name with the following meanins:

- S size
- 5 MD5-checksum
- L symbolic link
- D device
- U user
- G group
- M modus (permissions)
- T modification time

## 3.9.3 Installing and uninstalling RPMs

If you want to install some third party RPMs, first query them and make sure they apply to your architecture (ppc) or are architecture-independent (noarch), then verify that they can be installed on your distribution.

As for querying, we need a full RPM file name (name-version-release.arch.rpm) if we are going to install it, but we only need the package name (name) if we are uninstalling it. Table 3-5 lists examples for RPM installation and uninstallation.

*Table 3-5 RPM installation*

| Command | Purpose |
|---|---|
| rpm -ivh packagename.rpm | Install package, be verbose and show hash |
| rpm -Uvh | Install package that is already installed (upgrade) |
| rpm -Fvh *rpm | Install only those newer rpms that are already present on the system (freshen) |
| rpm -e packagename | Uninstall rpm |

RPMs are always distributed as a full package, not as an update package. If for some reason you need to downgrade an RPM to the level you used before upgrading, you need to uninstall it and reinstall the older RPM. If you are not able to uninstall an RPM because of other packages depending on it, use the **`--nodeps`** option when uninstalling, and reinstall the older version immediately.

Surprisingly, on SuSE SLES, YaST can be used to install RPMs on the command line, and under some circumstances it resolves dependencies automatically. If you call **`yast -i shortrpmname`** it will try to resolve dependencies of the package.

If your system has any unresolved dependencies (not necessarily related to this package), then the command will become interactive and ask what to do with the unresolved dependency conflict.

If the system is in good condition and the installation CD (or other directories provided in **`yast inst_source`**) is accessible, then YaST will install the package requested and even resolve its dependencies if they are resolvable through other packages present in the installation source.

If we call **`yast -i fullpackagename`**, then YaST will just install the RPM. It acts like **`rpm -U`**. It will fail just like rpmif the dependencies of the package are unresolved.

## 3.10 System updates

SuSE Linux provides two different ways to keep your server up-to-date: interactive updates through YaST2, and **`online_update`** for unattended updates. We can use maintenance CDs or any directory, NFS-share, http or ftp server. The server list is automatically updated.

> **Tip:** If you are behind a firewall and need to use passive ftp for online updates set “passive_ftp = on” in /etc/wgetrc, as YaST uses wget to retrieve patches.

Online update is a useful feature, but in a production environment, we do not always have Internet access from every node or LPAR. In order to keep these nodes up-to-date, we need to set up our own update server.

### 3.10.1 Setting up our own SuSE update server

In order to create our own update server in a private network, we need to replicate the directory structure used by SuSE on our network share. We can use NFS, FTP, HTTP.

1. In our case, we decided to use NFS. We created a new logical volume, update-lv, and mounted it under /update. We shared this directory through NFS to all nodes in the network.
2. Next we create a directory structure expected by online_update:

   ```
   ../architecture/updat/Product-Name/Version
   ```

In our case, for SuSE SLES 8 for pseries it means:

```
../ppc/update/SuSE-SLES/8/patches
../ppc/update/SuSE-SLES/8/rpm
```

3. Now we can download the patches and RPMs or copy them from the maintanence CD (Service Pack CD) in to the created directories.
4. On the client, we need to edit /etc/sysconfig/onlineupdate and set:

   ```
   YAST2_LOADFTPSERVER="no"
   ```

   This will prevent overwriting of our custom setting in /etc/suseservers.
5. We edit /etc/suseservers:

   ```
   #
   # sample /etc/suseservers
   #
   #ftp://ftp.suselinux.hu/pub/suse
   # Our local update server:
   nfs://p630sles/update
   ```

6. At this point, we can try to run online updates with -V (verbose) -d (dry run) and -D (debug) (see online_update --help for more options):

   ```
   # online_update -V -d -D
   ```

   We can choose to update a particular package; for more online_update options, see Example 3-24:

*Example 3-24 online_update options*

```
Usage: online-update [-u url] [-p product] [-v version] [-a arch] [-d] [-s]
[-n] [-g] [-i] [security] [recommended] [document] [optional]

-u url      Base URL of directory tree used to get patches from.

-g          Only download patches, don't install.
-i          Install downloaded patches, don't download.

-p product  Name of product to get patches for.
-v version  Version of product to get patches for.
-a arch     Base architecture of product to get patches for.

-d          Dry run. Only get patches, don't install them.
-n          No signature check of downloaded files.

-s          Show list of patches.
-V          Be verbose.
-D          Debug output.
```

## 3.10.2 Online update on Red Hat AS

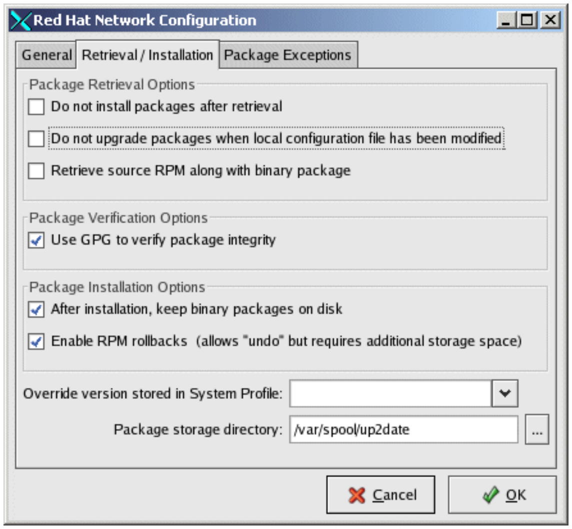
Red Hat provides a tool for online updates through Red Hat Network: up2date. This tool works in X Windows or from the command line. First you need to issue:

```
rpm --import /usr/share/rhn/RPM-GPG-KEY
```

Next, run **up2date** or **rhn_register**. The first time you run the command, it will ask you to provide your Red Hat Network user name and password, and you will need to register your system. It will display a list of available channels and the list of available updates. For more information, see:

```
http://www.redhat.com/software/rhn/update/
```

You can also start **rhn_register** with the **--configure** option to change the default settings or to configure a proxy server, as shown in Figure 3-12.



Red Hat Network Configuration

Package Retrieval Options

□ Do not install packages after retrieval

□ Do not upgrade packages when local configuration file has been modified

□ Retrieve source RPM along with binary package

Package Verification Options

☑ Use GPG to verify package integrity

Package Installation Options

☑ After installation, keep binary packages on disk

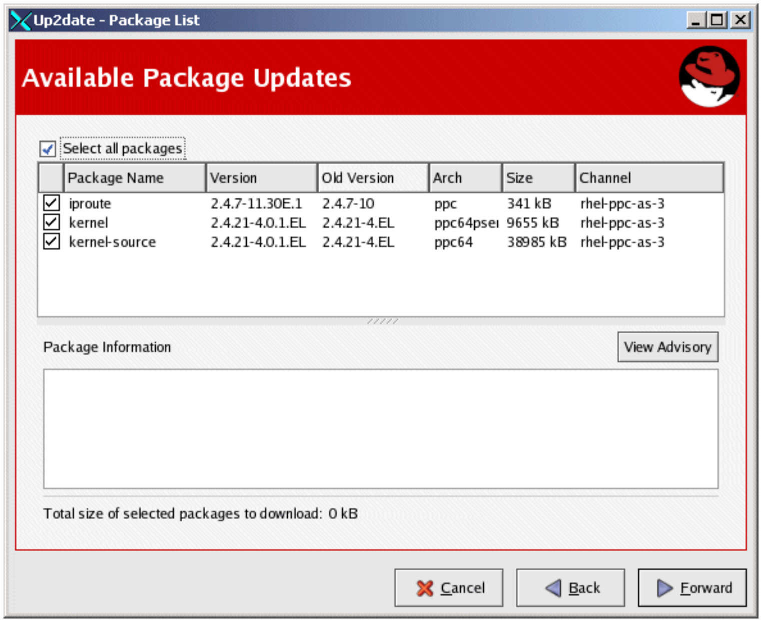
☑ Enable RPM rollbacks (allows "undo" but requires additional storage space)

Override version stored in System Profile:

Package storage directory: /var/spool/up2date

*Figure 3-12 Red Hat Network configuration*

Up2date gathers information about system hardware and software and sends this to Red Hat Network in order to determine available software updates. It runs from the command line or in graphical mode if the DISPLAY variable is set. Figure 3-13 on page 142 shows a list of available updates.



Available Package Updates

☑ Select all packages

| | Package Name | Version | Old Version | Arch | Size | Channel |
|---|---|---|---|---|---|---|
| ☑ | iproute | 2.4.7-11.30E.1 | 2.4.7-10 | ppc | 341 kB | rhel-ppc-as-3 |
| ☑ | kernel | 2.4.21-4.0.1.EL | 2.4.21-4.EL | ppc64psei | 9655 kB | rhel-ppc-as-3 |
| ☑ | kernel-source | 2.4.21-4.0.1.EL | 2.4.21-4.EL | ppc64 | 38985 kB | rhel-ppc-as-3 |

Package Information

Total size of selected packages to download: 0 kB

*Figure 3-13 Up2date available updates for our system*

## 3.11 System backup

In this section, we discuss backup solutions available for Linux on pseries. We describe the installation of Tivoli® Storage Manager (TSM) in 8.3, “Tivoli Storage Manager (TSM)” on page 406. In addition to commercial solutions, there are many UNIX built-ins, such as tar, cpio or pax, which can be used for local, tape or network backups.

For network backups, we can also use **`rsync`**. It transfers data compressed and through ssh, it can be used to make incremental and rotating backups. Some generic examples for using rsync as a backup tool can be found at:

`http://rsync.samba.org/examples.html`

There are also many scripts that utilize rsync to do more comfortable and sophisticated network backups:

**`RIBS`** is available at:

`http://rustyparts.com/scripts.php`

**`rsync-backup`** is available at:

```
http://freshmeat.net/projects/rsync-backup/?topic_id=137%2C861
```

**duplicity** is available at:

```
http://www.nongnu.org/duplicity/
```

## Using amanda for backup

The amanda (or Advanced Maryland Automatic Network Disk Archiver) backup system, allows the administrator of a LAN to set up a single master backup server to back up multiple hosts to a single large capacity tape drive. amanda is open source software distributed under BSD license.

The home page of amanda is:

```
http://www.amanda.org
```

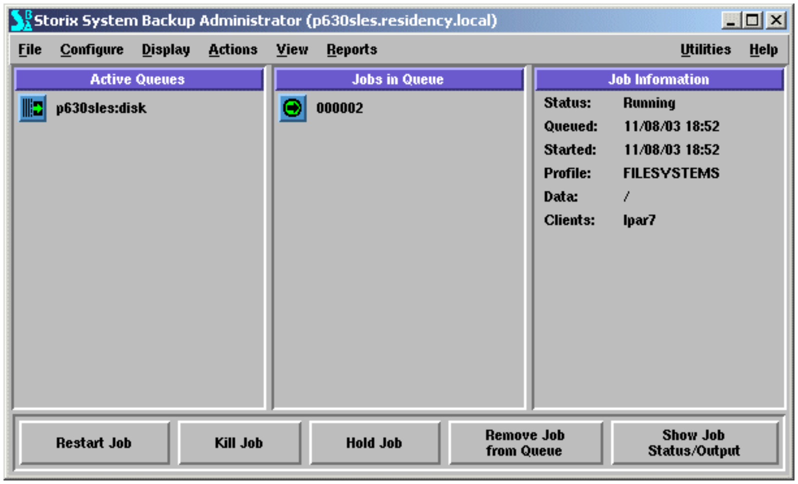
Despite the lack of a sophisticated graphical interface, amanda is a competitive product widely used in Linux environments.

## Using storix for backup

One of the commercial backup solutions available for 64-bit PowerPC Linux is storix. You can visit the storix homepage and download an evaluation version of the software:

```
http://www.storix.com
```

It worked fine on our test system, except that we were not able to create bootable rescue CDs, which is an outstanding feature from this software.



p630sles:disk

000002

Status: Running
Queued: 11/08/03 18:52
Started: 11/08/03 18:52
Profile: FILESYSTEMS
Data: /
Clients: lpar7

*Figure 3-14 storix running a backup job*

# 3.12 ssh

After default installation of any SuSE or Red Hat system, the only way to access it remotely is by using ssh. The telnet port is closed by default and we strongly recommend, for security reasons, that you do not change it.

If you are accessing the system from another Linux box, you will usually have the openssh client installed.

For AIX 5L, you can install it from:

http://www-124.ibm.com/developerworks/projects/opensshi

For AIX 4.3, you can install it from:

http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html

On Windows, you can use one of three ssh clients, for example, putty:

http://www.putty.nl/download.html

For the same security reason, we did not want to use .rhosts file and rsh. In order to be able to run unattended remote commands or file copies, we need to set up ssh for passwordless login. This can be achieved through ssh key exchange.

In a test environment, this can be done in both ways, but in a production environment, we recommend that you enable login from a central management server to the nodes only, not from every node to every node.

## 3.12.1 Exchanging ssh keys

In this example, we enable passwordless login from serverA (lpar7) to serverB (lpar1). This requires two steps: generating a public key on serverB, and placing this key in .ssh/authorized_keys2 file on serverA.

In order to generate a public key on serverB, we run following command:

```
lpar7:~# ssh-keygen -t rsa -b 1024 -N ""
```

As a result, two files are created in /root/.ssh directory: id_rsa  and id_rsa.pub.

In the next step, we copy the file to the serverA:

```
lpar7:~# scp id_rsa.pub lpar1:/tmp
```

We add its content to the /root/.ssh/authorized_keys2:

```
lpar1:~ # cat /tmp/id_rsa.pub >> /root/.ssh/authorized_keys2
```

Now we can go back to serverA (lpar7) and try ssh login to serverB (lpar1). If everything is correct, we will not be prompted for a password:

```
lpar7:~/.ssh # ssh lpar1
Last login: Mon Nov  3 18:17:41 2003 from lpar7
lpar1:~ #
```

If your public key authentication does not work, then look in /var/log/messages for the reason and check the permissions (0655 for public key and 600 for id.rsa). A world readable home directory will prevent this authentication from working, as well.

> **Important:** If you are going to implement CSM, then do not exchange root keys manually; CSM will take care of this. You can exchange keys for other users .

## 3.13 DHCP server configuration

Dynamic Host Configuration Protocol (DHCP) can be used for automatically assigning IP addresses and other information to the hosts in the network. The configuration contains only one file. However, starting the dhcpd server in the enterprise network with one dhcpd server already running can result in some confusion or damage if the clients configured to use dhcp suddenly receive other addresses.

So be sure to have your server in a separate network, or use statical IP configuration for all nodes and servers that you do not want to serve. Clients usually take the first address offer they get, and often it is the wrong one.

*Example 3-25 /etc/dhcpd.conf*

```
not authoritative;
ddns-update-style ad-hoc;
default-lease-time 60000;
max-lease-time 720000;

option myoption code 129 = text ;
# dynamically assigned adress range
# every client can get one

subnet 192.168.100.0 netmask 255.255.255.0 {
range 192.168.100.211 192.168.100.212;
option routers 192.168.100.60;
option nameservers 192.168.100.110;
option domain residency.local;
# statically assigned adresses, only clients
# with designated MAC adress will get it
group {
        next-server 192.168.100.110;

       host lpar1 {
             fixed-address 192.168.100.77;
             hardware ethernet 00:02:55:3A:06:8C;
            filename "install";
       }
       host lpar2 {
             fixed-address 192.168.100.78;
             hardware ethernet 00:02:55:6f:1f:e3;
            filename "yaboot";
            option root-path "/tftpboot/";
}

        host lpar3 {
             fixed-address 192.168.100.79;
             hardware ethernet 00:02:55:3a:06:19;
            filename "yaboot";
       }
}
}
```

After editing /etc/dhcpd.conf, we start our dhcpd server:

```
# /etc/init.d/dhcpd start
```

Check /var/log/messages:

*Example 3-26 /ect/var/log/messages excerpt:*

```
....
dhcpd: Internet Software Consortium DHCP Server V3.0.1rc9
Oct 22 15:40:50 p630sles dhcpd: Copyright 1995-2001 Internet Software
Consortium.
Oct 22 15:40:50 p630sles dhcpd: All rights reserved.
Oct 22 15:40:50 p630sles dhcpd: For info, please visit
http://www.isc.org/products/DHCP
Oct 22 15:40:50 p630sles dhcpd: Wrote 0 deleted host decls to leases file.
Oct 22 15:40:50 p630sles dhcpd: Wrote 0 new dynamic host decls to leases file.
Oct 22 15:40:50 p630sles dhcpd: Wrote 0 leases to leases file.
Oct 22 15:40:50 p630sles dhcpd: Listening on LPF/eth0/00:02:55:4f:60:8a/CSM
Oct 22 15:40:50 p630sles dhcpd: Sending on   LPF/eth0/00:02:55:4f:60:8a/CSM
Oct 22 15:40:50 p630sles dhcpd: Sending on   Socket/fallback/fallback-net
....
```

If you have left the dynamically assigned address range in the dhcpd.conf file, then every client configured to be a dhcpd client will get an address from this range. For network installations or network boot, we strongly recommend that you use statically assigned addresses in order to prevent installing servers or clients you do not intend.

## 3.14 DNS server configuration

As a starting point we use the default caching only nameserver configuration provided by SuSE. The caching only name server forwards requests to other nameservers provided by keyword forwarders and caches the answers for better performance. This configuration is good enough if you only need to resolve names and addresses in the Internet.

In order to use local name resolution too, we add two new zone records: residency.local (name to IP address), and 100.168.192.rev for reverse name resolution (IP address to name). For debugging, we add the logging section because it forces the name server daemon to log all requests in /var/log/messages.[18]

*Example 3-27 /etc/named.conf*

```
options {
directory "/var/named";
        forwarders {
                9.12.6.7;
                };
listen-on { any;};
notify no;
        forward first;
};

zone "localhost" in {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "127.0.0.zone";
};

zone "." in {
type hint;
```

[18] For more information see BIND 9 Administrator Reference Manual: http://www.bind9.net/Bv9ARM.html

```
        file "root.hint";
};

# You can insert further zone records for your own domains below.


logging {
        category queries {
                default_syslog;
                };
        category update {
                default_syslog;
                };
        };
zone "residency.local" {
        type master;
        file "/var/named/residency.local.hosts";
        };

zone "100.168.192.in-addr.arpa" {
type master;
        file "/var/named/100.168.192.rev";
        };
```

Now we create our own local zone files: residency.local, and for reverse lookup, 100.168.192.rev. We need a residency.local file in order to be able to resolve names to IP addresses.

*Example 3-28 /var/named/residency.local*

```
$ttl 38400 residency.local.   IN   SOA     192.168.100.110. root.p630sles. (
                         0310221736  ; serial
                        10800        ; refresh
                        3600         ; retry
                        604800       ; expiry
                        38400 )      ; minimum

residency.local.        IN      NS      192.168.100.81
$GENERATE 111-254      node-$   A        192.168.100.$
lpar8.residency.local.  IN      A       192.168.100.84
lpar7.residency.local.  IN      A       192.168.100.83
lpar6.residency.local.  IN      A       192.168.100.82
lpar5.residency.local.  IN      A       192.168.100.81
lpar4.residency.local.  IN      A       192.168.100.80
lpar3.residency.local.  IN      A       192.168.100.79
lpar2.residency.local.  IN      A       192.168.100.78
lpar1.residency.local.  IN      A       192.168.100.77
p630sles.residency.local.       IN      A       192.168.100.110
```

**Important:** Every time you change something in the zone file, you need to change the serial number in the second line.

In order to resolve addresses to names, we need a file for reverse name resolution. In this file, addresses are written in opposite order: 77.100.168.192, instead of 192.168.100.77.

*Example 3-29 /var/named/100.168.192.rev*

```
$ttl 38400100.168.192.in-addr.arpa. IN    SOA     p630sles. root.p630sles. (
                        9998899999
                        10800
                        3600
                        604800
                        38400 )

100.168.192.in-addr.arpa.      IN      NS      p630sles

77.100.168.192.in-addr.arpa.   IN      PTR     lpar1.
78.100.168.192.in-addr.arpa.   IN      PTR     lpar2.
79.100.168.192.in-addr.arpa.   IN      PTR     lpar3.
80.100.168.192.in-addr.arpa.   IN      PTR     lpar4.
81.100.168.192.in-addr.arpa.   IN      PTR     lpar5.
82.100.168.192.in-addr.arpa.   IN      PTR     lpar6.
83.100.168.192.in-addr.arpa.   IN      PTR     lpar7.
84.100.168.192.in-addr.arpa.   IN      PTR     lpar8.
110.100.168.192.in-addr.arpa.  IN      PTR     p630sles.
$GENERATE 111-254 $           PTR     node-$.
```

**Tip:** The trick in both zone files is in the line starting with $GENERATE: it will assign all nodes above 111 names; that is, node-111, node-112 and so on. This is a useful feature in combination with a dhcpd server for a larger environment or cluster, because you do not need to add nodes one by one anymore.

## 3.15 Using iptables for security

The netfilter/iptables project is the Linux 2.4.x / 2.5.x firewalling subsystem.It delivers you the functionality of packet filtering (stateless or stateful), many different kinds of Network Address Translation (NAT) and packet managing. The homepage of Linux iptables is:

http://www.netfilter.org

Iptables is a part of the Linux kernel or more precisely: “netfilter is a set of hooks inside the linux 2.4.x kernel's network stack which allows kernel modules to register callback functions called every time a network packet traverses one of those hooks”.[19]

Here we illustrate the use of iptables in a simple example: we have a management server in our environment and several nodes. We want to allow one node to reach the management server and to allow the management server reach the node, but to deny any kind of IP communication from one node to other nodes. This star-formed structure applies to CSM design: the CSM management server communicates with nodes, but nodes do not need to communicate with each other.

In Example 3-30, we limit the communication from a node to one IP address, the management server, allowing any kind of IP traffic between them:

```
/sbin/iptables -I INPUT -s ADDRESS -j ACCEPT
```

In detail, insert rule (-I) in the chain INPUT anything with source address (-s) management server will be accepted. The same command for the OUTPUT chain with the **-d** option determines outgoing traffic to the destination. We use default policy DROP - this means the packets not matching the allowed packet rules will be dropped and no reply will be sent. The alternate policy REJECT would mean the packets are rejected and a reply is sent to the initiator saying that the packet was rejected.

**Attention:** Before experimenting with iptables, ensure you will be able to log in locally or through HMC, or you may lock yourself out. Do not play with servers without having a direct, non-networked login possibility, or at a minimum, set a job to remove the firewall in 30 minutes.

*Example 3-30 ip-isolate script*

```
# This is the location of the iptables command
IPTABLES="/sbin/iptables"
# Address of our  Management Server
MGMT=192.168.100.110

## Flush everything, start from scratch
#
# Incoming packets from the outside network
$IPTABLES -F INPUT
# Outgoing packets from the internal network
$IPTABLES -F OUTPUT
# Forwarding/masquerading
```

[19] see http://www.netfilter.org/documentation/

```
$IPTABLES -F FORWARD

#Allow everything from and to Management Server
$IPTABLES -I INPUT -s $MGMT -j ACCEPT
$IPTABLES -I OUTPUT -d $MGMT -j ACCEPT

# Set default policy to drop
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP
```

> **Note:** The location of the **iptables** command is /usr/sbin/iptables on SuSE, and /sbin/iptables on the Red Hat system.

After we finished editing our ip-isolate script, we need to make it executable and run it. We can see the state of iptables with the **iptables -L** command:

*Example 3-31 iptables -L output*

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  p630sles             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             p630sles
```

As we see in Example 3-31, our default policy for all chains is drop, but we accept anything from machine p630sles and allow all traffic to it. If we try to ping lpar1 (which is another node in the same subnet), we will see following output:

```
ping lpar1
PING lpar1.residency.local (192.168.100.77) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
```

We also cannot ping lpar6 from any other node in the network, or log in to it with ssh. Only traffic to and from the management server is allowed.

We can improve our node isolation by regulating which kind of network traffic is allowed from and to the management server.

*Example 3-32 port-isolate script*

```
#!/bin/sh
### This is the location of the iptables command
IPTABLES="/sbin/iptables"
### Adress of our CSM Management Server
CSM=192.168.100.110
DNS=192.168.100.110
###
## Flush everything, start from scratch
## Incoming packets from the outside network
$IPTABLES -F INPUT
## Outgoing packets from the internal network
$IPTABLES -F OUTPUT
## Forwarding/masquerading
$IPTABLES -F FORWARD
########################## Allow ssh from and to CSM Management Server
$IPTABLES -I INPUT -s $CSM -p tcp --dport 22 -j ACCEPT
$IPTABLES -I OUTPUT -d $CSM -p tcp --dport 22 -j ACCEPT
########################## Allow usage of DNS
$IPTABLES -I OUTPUT -d $DNS -p udp --dport 53 -j ACCEPT
########################## CSM
$IPTABLES -I OUTPUT -d $CSM -p tcp --dport 657 -j ACCEPT
$IPTABLES -I OUTPUT -s $CSM -p udp --dport 657 -j ACCEPT
########################## Be stateful
$IPTABLES -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -I OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
######################### Default policy is DROP
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP
######################### We log for debugging only
#$IPTABLES -A OUTPUT -j LOG
#$IPTABLES -A INPUT -j LOG
```

Now we have forbidden everything except connections to the management server for ssh (port 22 tcp), name server (port 53 udp) and csm (port 657 tcp/udp). We also accept connections established and related to the connections allowed. We will be able to make an ssh connection to the management server, but not to ping it. We have forbidden any connections to other nodes or from them.

**Important:** ssh login may fail if the client machine is an LDAP Client and cannot reach the LDAP server, even if root is not an LDAP user.

In the last example, we show a small script to remove our firewall rules.

*Example 3-33 fwremove script*

```
# Remove any existing rules from all chains
iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

## 3.16 OpenLDAP implementation

The OpenLDAP-Server and client package is included in the standard SuSE SLES installation on pSeries. Here we describe a simple LDAP configuration for user authentication in a networked environment. The benefits of using LDAP are centralized and simplified user management from anywhere in the network, hierarchical structure, and replication for availability.

In this section, we explain how to set up an LDAP server on SuSE and how to use it for common user authentication tasks. Once you have LDAP running in your network, it can be used for many different tasks: addressbook, hostname repository, assessment management, and so on.

### 3.16.1 Setting up an LDAP server

Setting up the LDAP server consists of two steps: configuring the server, and defining the users.

#### LDAP server configuration

The server configuration is managed by one file, /etc/openldap/slapd.conf.

We verify the openldap server package is installed:

```
p630sles:~ # rpm -qa |grep openldap
openldap2-client-2.1.4-43
openldap2-2.1.4-48
```

Next, we edit /etc/openldap/slapd.conf.

Example 3-4 on page 108 contains a very basic but working template for slapd.conf.

*Example 3-34* /etc/openldap/slapd.conf

```
# First we describe schemata to be included
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/inetorgperson.schema


allow           bind_v2
pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args

# We define acls here: with default acs authentication will work,
# but users will not be able to change their passwords

access to attr=userPassword
        by dn="cn=admin,dc=residency,dc=local" write
        by self write
        by anonymous read
        by * auth

access to *
        by dn="cn=admin,dc=residency,dc=local" write
        by anonymous read
        by self read

database        bdb
suffix          "dc=residency,dc=local"
rootdn          "cn=admin,dc=residency,dc=local"
rootpw  abc123
directory       /var/lib/ldap

# Indices to maintain
index   objectClass     eq
```

### Starting the LDAP server

Change the lines with keywords suffix, rootdn, and rootpw to your needs. If you comment out all ACL settings, the authentication will still work, but users will not be able to change their own passwords. This example LDAP configuration is not a secure one. In a production, we strongly recommend using TSL.

W we start the server:

```
# /etc/init.d/ldap start
```

(or by using the SuSE shortcut: **`rcldap start`**)

Look for error messages:

```
# tail -f /var/log/messages
```

**Tip:** You may see messages like: `unable to open Berkeley db /etc/sasldb: No such file or directory`. Do not be concerned; you can avoid this by creating the /etc/sasldb database by issuing:

```
#saslpasswd dummyuser
```

Now you have a file, /etc/sasldb, and you need to make it readable by the LDAP user:

```
# chmod o+r /etc/sasldb
```

Check the server status after a few minutes:

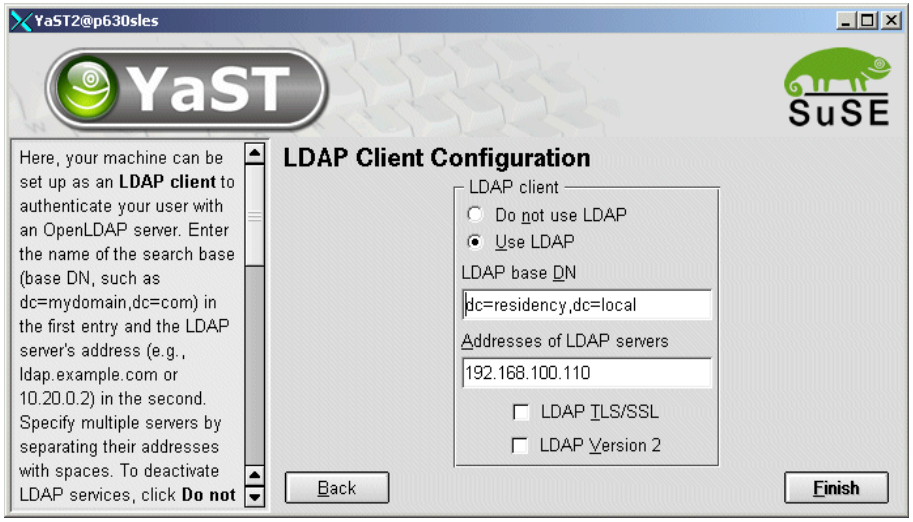
```
# /etc/init.d/ldap status
```

## 3.16.2 LDAP client configuration

Now we can set up the server to be an LDAP client of itself. We can use YaST, as follows:

Starting YaST2 with a shortpath:

```
# yast2 ldap
```



Here, your machine can be set up as an **LDAP client** to authenticate your user with an OpenLDAP server. Enter the name of the search base (base DN, such as dc=mydomain,dc=com) in the first entry and the LDAP server's address (e.g., ldap.example.com or 10.20.0.2) in the second. Specify multiple servers by separating their addresses with spaces. To deactivate LDAP services, click **Do not**

**LDAP Client Configuration**

LDAP client

○ Do not use LDAP

● Use LDAP

LDAP base DN: dc=residency,dc=local

Addresses of LDAP servers: 192.168.100.110

□ LDAP TLS/SSL

□ LDAP Version 2

*Figure 3-15 Using YaST to configure the LDAP client*

Since our users are in the LDAP container dc=residency, dc=local, make it a base DN. The server address this case is: 192.168.100.110.

After pressing Finish, our LDAP client is ready for use.

YaST made changes in three files:

- /etc/security/pam_unix2.conf

```
   ....
auth: use_ldap nullok
account: use_ldap
password: use_ldap nullok
...
```

- /etc/openldap/ldap.conf

```
...
base    ou=user,dc=residency,dc=local
host    192.168.100.81
ldap_version    3
...
```

- /etc/nssswitch.conf

```
...
passwd: files ldap
group: files ldap
shadow: files ldap
...
```

**Tip:** After changing these files, YaST calls the **`/sbin/SuSEconfig`** command. If you are going to distribute the files over manually, run this command afterwards.

You also need to run **`/sbin/SuSEconfig`** every time you change settings in the /etc/sysconfig directory.

### 3.16.3 Defining users in the LDAP directory

LDAP can import and export the contents of the database through ldif-files. In the next step, we create a user.ldif file and import the data into the LDAP directory.

This first ldif file contains the definition of the container (user) and one user; see Example 3-35 on page 156.

*Example 3-35 /root/user.ldif file*

```
dn: dc=residency,dc=local
objectClass: organization
o: residency
dn: ou=user, dc=residency,dc=local
ou: user
objectClass: organizationalUnit


dn: uid=lee,ou=user, dc=residency,dc=local
sn: lee
loginShell: /bin/bash
userPassword: abc123
gidNumber: 100
uidNumber: 1203
shadowMax: 99999
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
uid: lee
shadowLastChange: 11472
cn: lee
homeDirectory: /tmp
shadowWarning: 7
```

After we create the file, we import it to LDAP:

```
# ldapadd -c -x -D "cn=admin,dc=residency,dc=local" -W -x -f
/root/user.ldif
```

The following output should be received:

```
Enter LDAP Password:
adding new entry "dc=residency,dc=local"
adding new entry "ou=user, dc=residency,dc=local"
adding new entry "uid=lee,ou=user, dc=residency,dc=local"
```

Now we can search LDAP for entries we added; see Example 3-36:

*Example 3-36 searching LDAP entries*

```
# ldapsearch -x
# extended LDIF
#
# LDAPv3
# filter: (objectclass=*)
# requesting: ALL
#
```

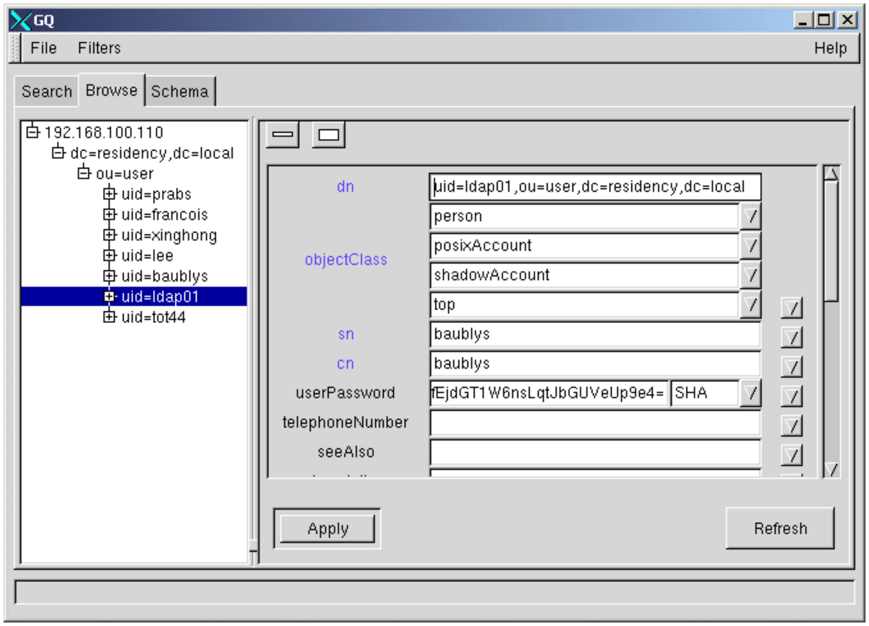
```
# user, residency.local
dn: ou=user,dc=residency,dc=local
ou: user
objectClass: organizationalUnit

# prabs, user, residency.local
dn: uid=prabs,ou=user,dc=residency,dc=local
uid:: cHJhYnMg
cn: prabs
sn:: cHJhYnMg
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
userPassword:: YWJjMTIz
shadowLastChange: 11472
shadowMax: 99999
shadowWarning: 7
uidNumber: 1200
gidNumber: 100
homeDirectory: /tmp
loginShell: /bin/bash
```

We can also try ssh with a new, added LDAP user:

```
# ssh lee@localhost
```

### 3.16.4 Managing users

There are various LDAP clients or browsers that can help you to understand the directory structure or fulfill administration tasks. One of them, which is available in the default SLES8 installation, is gq; see Figure 3-16 on page 159.



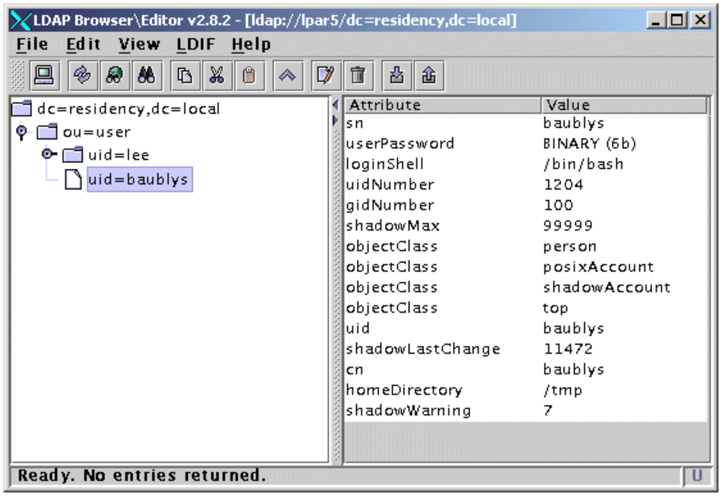
dn: uid=ldap01,ou=user,dc=residency,dc=local
objectClass: person
posixAccount
shadowAccount
top
sn: baublys
cn: baublys
userPassword: fEjdGT1W6nsLqtJbGUVeUp9e4= SHA
telephoneNumber:
seeAlso:

*Figure 3-16 gq LDAP client*

Another example is “LDAP Browser/Editor” written in Java by Jarek Gawor. You can download it from:

```
http://www.iit.edu/~gawojar/ldap/
```

This browser works on Windows and Linux systems, and can be used to manage LDAP users; see Figure 3-17 on page 160.


LDAP Browser\Editor v2.8.2 - [ldap://lpar5/dc=residency,dc=local]
File
Edit
View
LDIF
Help

dc=residency,dc=local
ou=user
uid=lee
uid=baublys

| Attribute | Value |
|---|---|
| sn | baublys |
| userPassword | BINARY (6b) |
| loginShell | /bin/bash |
| uidNumber | 1204 |
| gidNumber | 100 |
| shadowMax | 99999 |
| objectClass | person |
| objectClass | posixAccount |
| objectClass | shadowAccount |
| objectClass | top |
| uid | baublys |
| shadowLastChange | 11472 |
| cn | baublys |
| homeDirectory | /tmp |
| shadowWarning | 7 |

Ready. No entries returned.

*Figure 3-17 LDAP browser/editor in Java*

## 3.16.5 Using SSL/TLS with OpenLDAP

LDAP supports encryption and authentication using Transport Layer Security (TLS). LDAP passwords in our previously described setup will be sent in clear text over the network if the LDAP server is not a local machine. This is generally not a good idea. In order to secure LDAP traffic, we can use TLS as follows:

1. Generate a CA.

   If we do not have a certificate signed by an official certificate authority, we need to generate it. First we use a tool to generate a new certificate authority CA, so we can sign our own certificates:

   ```
   # /usr/share/ssl/misc/CA.sh –newca
   ```

   This tool generates the Certificate for our CA (demoCA/cacert.pem )and the private key (demoCA/private/cakey.pem).

2. Generate a server certificate.

   Now we generate the server certificate:

   ```
   # openssl req -newkey rsa:1024 -nodes -keyout newreq.pem -out newreq.pem
   ```

3. Sign the server certificate.

   In order to sign our server certificate, we issue:

   ```
   # /usr/share/ssl/misc/CA.sh -sign
   ```

4. Copy certificates to LDAP.

Now we need to copy certificates to /etc/openldap/cert directory and chown them to the ldap user.

```
lpar7:~ # mkdir /etc/openldap/cert
lpar7:~ # cp demoCA/cacert.pem /etc/openldap/cert/
lpar7:~ # cp demoCA/cacert.pem /etc/openldap/cert/cacert.pem
lpar7:~ # cp newcert.pem /etc/openldap/cert/servercrt.pem
lpar7:~ # cp newreq.pem /etc/openldap/cert/serverkey.pem

lpar7:~ # chown -R ldap:ldap /etc/openldap/cert
```

5. Add the keys to slapd.conf.

   We add these lines to /etc/openldap/slapd.conf in order to utilize the created certificates:

```
TLSCertificateFile    /etc/openldap/certificates/servercrt.pem
TLSCertificateKeyFile /etc/openldap/certificates/serverkey.pem
TLSCACertificateFile  /etc/openldap/certificates/cacert.pem
```

6. Start the server.

```
# /usr/lib/openldap/slapd -d9 -f /etc/openldap/slapd.conf -h
ldaps://192.168.100.83:636 -u ldap -g ldap
```

7. Configure the client.

   To configure the client, we need to change ldap.conf to use secure communication; see Example 3-37.

*Example 3-37 /etc/openldap/ldap.conf*

```
base    dc=residency,dc=local
uri     ldaps://192.168.100.83:636
host    192.168.100.83
SASL_SECPROPS noactive
#ssl    start_tls
TLS     hard
TLS_REQCERT     allow
```

4

# Linux for pSeries RAS and problem determination

Problem determination is definitely a key area when it comes to system administration. System administrators tend to spend hours debugging and trying to find out what is wrong with the system. In this chapter, we discuss how to do centralized system logging using the syslog and the evlog. We also discuss Linux rescue methods, and provide tips that allow you to fix boot partition errors, corrupt file systems, and configuration problems.

We also discuss the ppc64 Run Time Abstraction Service (RTAS). RTAS allows you to query and extract information from the pSeries firmware. We share with you the tools developed by the IBM Linux Technology Center to allow you to work more closely with AIX, commands which are used in AIX to extract hardware information, and patch system firmware.

Finally, we discuss common tuning methods in Linux, and explain which tools you can use to tune Linux to allow it to do more for you.

## 4.1 Linux on pSeries RAS

The IBM pSeries hardware has been known for its RAS capabilities due to IBM's knowledge and experience in developing mainframes and mission-critical servers. Much of the RAS design has been developed to analyze failures within the Central Electronic Complex (CEC) to either eliminate the errors or to contain and reduce them to avoid bringing the entire server down. Some of the RAS features that you see available for Linux on pSeries are:

- Persistent deallocation for memory and processor during boot-time
- Automatic First Failure Data Capture and diagnostics capability
- ECC and chipkill correction in the real memory
- Fault tolerance with N+1 redundancy of power and cooling
- Dual line cords
- Predictive failure analysis and diagnostics
- Robust journaled file system using JFS, reiserfs and others

Many of the design efforts put into the development of the pSeries server RAS have been designed to be operating system-independent. This basically means that you do not need the AIX operating systems to exploit most of the RAS capability inside the hardware.

In the booting up process, the Built-In Self Test (BIST) and Power-on Self Test (POST) are designed to check the processors, caches, memory prior to loading the operating systems. If a critical error is detected, the system tries to deallocate the component and continue the boot-up process. In this way, your system is not at risk of running with a faulty component. Detected errors are logged into the system non-volatile RAM (NVRAM). Refer to 4.1.2, "IBM diagnostics tools" on page 168 for more information on the nvram.

Surveillance of the system operation is provided by the service processor. The service processor basically records and automatically checks for heartbeats from the operating systems. It can be configured to automatically reboot the system if the service processor does not detect any heartbeat within a default time interval. If the system is unable to come up successfully, the service processor logs the error and leaves the system powered on. The service processor is also designed to report errors to the Service Focal Point. In environments where the system is attached to a Hardware Management Console (HMC), the errors are logged and reported to the Service Focal Point application running in the HMC.

In additional, the IBM diagnostic tools for Linux on pSeries records and analyzes pSeries-specific messages, and logs them into the Linux system log facility. Generic software and hardware errors are also recorded and analyzed by the Linux error log analysis (LELA).

Refer to 4.1.2, “IBM diagnostics tools” on page 168 for the description of the tools packages inside the IBM diagnostics tools.

### 4.1.1 RunTime abstraction service in PPC64

Specific to the PowerPC kernel, /proc/rtas/* gives you some interface to interact with the service processor directly. The RunTime Abstraction Service or RTAS in Linux is enabled by default by the SuSE Linux Enterprise Server (SLES) 8, or you can recompile the kernel by hand with the `CONFIG_PPC_RTAS` option. In Figure 4-1, you can see how the RTAS in Linux interacts with the pSeries firmware.

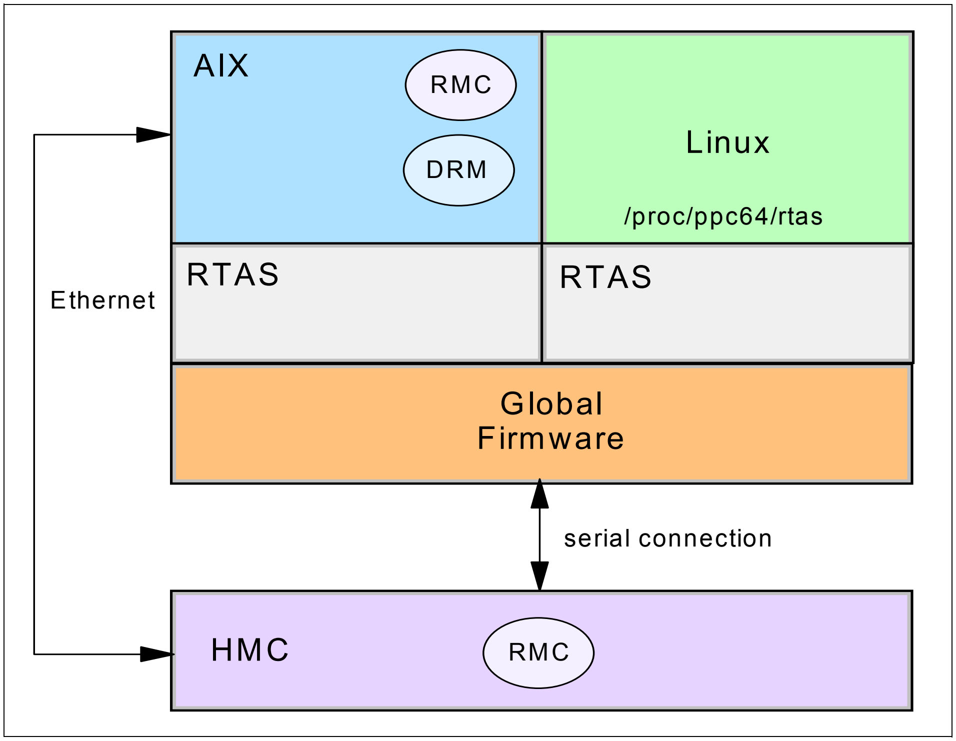


*Figure 4-1 ppc64 Linux RunTime abstraction service*

The open source community is continuously improving RTAS in the PowerPC kernel; here are some of the RTAS service in the /proc file systems that we can use today:

- /proc/ppc64/rtas/progress (read/write)

  The “progress” file allows you to write the LCD operator panel. For an HMC-attached device, this is shown in the Operator Panel Value.

This is very useful for displaying uptime or system performance of that LPAR or server:

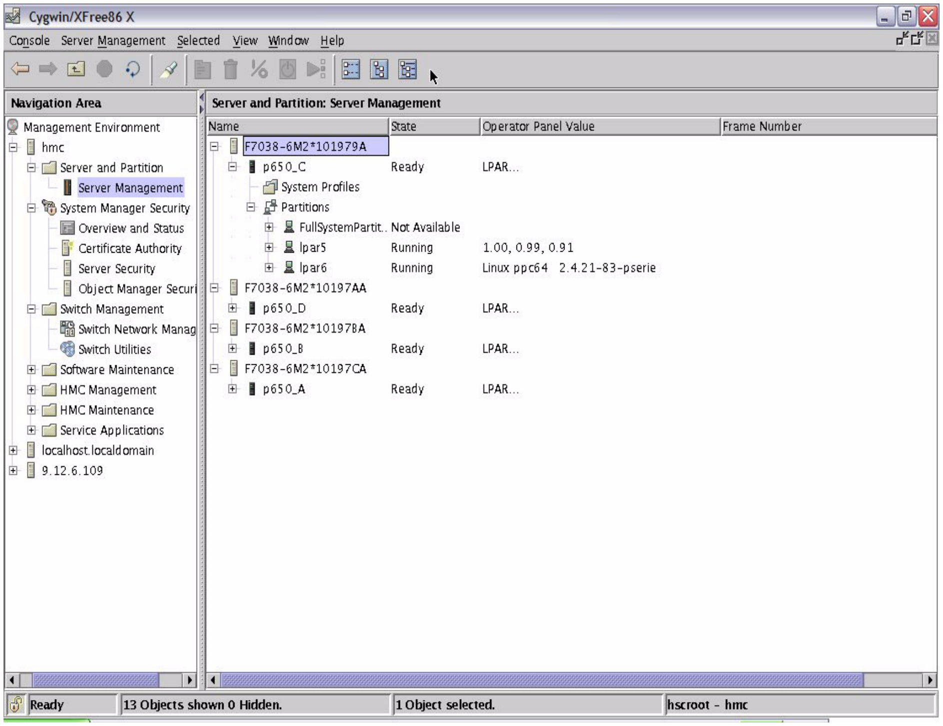
```
echo “this is a testing string” > /proc/ppc64/rtas/progress
```

One way to make use of the progress indicator is to display information about the health of the system. The script shown in Example 4-1 basically gets the output from the system load from the **uptime** command and displays it on the operator panel.

*Example 4-1 Shell script to echo system load to the pSeries operator panel*

```
#!/bin/sh
## Script to echo system load to Operator Panel
##

while true
do
UPTIME=‘uptime | sed ’s/^.*verage: //’‘
echo “                          “ > /proc/ppc64/rtas/progress
echo “$UPTIME” > /proc/ppc64/rtas/progress
sleep 1
done
```

After you run the script in the background shown in Example 4-1, you notice that the LED operator panel in the HMC displays the uptime as shown in Figure 4-2 on page 167.



Server and Partition: Server Management

| Name | State | Operator Panel Value | Frame Number |
|---|---|---|---|
| F7038-6M2*101979A | | | |
| p650_C | Ready | LPAR... | |
| System Profiles | | | |
| Partitions | | | |
| FullSystemPartit.. | Not Available | | |
| lpar5 | Running | 1.00, 0.99, 0.91 | |
| lpar6 | Running | Linux ppc64 2.4.21-83-pserie | |
| F7038-6M2*10197AA | | | |
| p650_D | Ready | LPAR... | |
| F7038-6M2*10197BA | | | |
| p650_B | Ready | LPAR... | |
| F7038-6M2*10197CA | | | |
| p650_A | Ready | LPAR... | |

Ready | 13 Objects shown 0 Hidden. | 1 Object selected. | hscroot - hmc

*Figure 4-2 Changing text in the LED panel*

Such a script can be run by the cron daemon based on a certain interval. In this way, the administrator knows about the system load without having to log on to the system to check.

- /proc/ppc64/rtas/clock (read/write)

  The `date` command in Linux for pSeries currently only changes the date/time for that particular session. If you need to change the time permanently, you need to update this file. The format of the file is the same as of the command **`date +%s`**.

  ```
  # echo "1068155156" > /proc/ppc64/rtas/clock
  ```

- /proc/ppc64/rtas/sensors (read)

  The sensors file allows you to be aware of the hardware operations of the server; the file presents you with a list of the sensors detected in the hardware, and gives you its environmental performance.

  Example 4-2 on page 168 shows the available sensors detected in a standalone p630 when you run the command **`cat /proc/ppc64/rtas/sensors`**.

*Example 4-2 Sensors for standalone servers*

```
RTAS (RunTime Abstraction Services) Sensor Information
Sensor           Value          Condition      Location
***************************************************************************
Key switch:      Normal         (read ok)      ---
Power source:    AC             (read ok)      ---
Interrupt queue: Enabled        (read ok)      ---
Surveillance:    0              (read ok)      ---
```

Example 4-3 shows sensors of an LPAR in a p650 server.

*Example 4-3 Sensors for an LPAR server*

```
RTAS (RunTime Abstraction Services) Sensor Information
Sensor           Value          Condition      Location
***************************************************************************
Key switch:      Normal         (read ok)      ---
Power source:    AC             (non existent) ---
Interrupt queue: Enabled        (read ok)      ---
```

The sensors that are available differ from those of servers, and will also be different if you are in LPAR mode. In LPAR mode, most of the hardware monitoring capabilities are done by the Hardware Management Console (HMC).

- /proc/ppc64/rtas/poweron (read/write)

  The file poweron allows you to set the date and the time to power on the system. This is very useful for a development server that needs to be shut down at night.

  ```
  date -d ‘tomorrow 9:00’ + %s > /proc/ppc64/rtas/poweron
  ```

- /proc/ppc64/rtas/frequency & /proc/rtas/volume (read/write)

  Frequency and volume allows you to manage the speaker in older RS/6000® hardware.

## 4.1.2 IBM diagnostics tools

IBM has recently released the Linux for pSeries Service aids for hardware diagnostics. The service aids allow system administrators to extract valuable information from the robust pSeries service processor for problem determination and servicing. Many of the commands packaged inside the service aids are very similar to the commands that you may find in AIX. The service aid can be downloaded from the Web site:

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

The IBM diagnostics tools require POWER4-based systems and a supported release of Linux on pSeries (SuSE SLES8 SP3 or Red Hat Advance Server 3).

To install the packages, run:

```
# rpm -ivh ppc64-utils-0.4-77.rpm
# rpm -ivh lsvpd-0.9.2-1.ppc.rpm
# rpm -ivh diagela-1.1.0.1-2.ppc.rpm
# rpm -ivh IBMinvscount-2.1-1.ppc.rpm
# rpm -ivh devices.chrp.base.ServiceRM-2.1.0.0-2.ppc.rpm
```

> **Important:** Be aware that the last RPM in this list (devices.chrp.base.ServiceRM) is dependent on the installation of the five RMC RPMs (src, rsct.core.utils, rsct.core, csm.core, and csm.client).
>
> These RPMs are also downloadable from:
>
> `http://techsupport.services.ibm.com/server/Linux_on_pSeries`

You need to initialize the lsvpd if you are running it for the first time:

```
# /etc/init.d/lsvpd start
```

Make sure that the lsvpd service is started. This basically creates a symbolic link inside /etc/rc.d/rc3.d and /etc/rc.d/rc5.d:

```
# chkconfig lsvpd 35 # (this will start it at runlevel 3 & 5)
```

After installing the packages, you find the following commands available to extract information from your pSeries server using Linux. These commands are installed into the /usr/sbin/ibmras or /usr/sbin/ directory. Some of these commands require root access.

- **`nvram`**

  This command is used to query and print the data stored in the nvram of the PowerPC-64 system. Example 4-4 shows the output of the **`nvram`** command.

  – `print-config`: prints out all the variables in the open firmware
  – `print-vpd`: prints out all the VPD
  – `print-all-vpd`: prints out all the VPD including vendor specific data
  – `print-event-scan`: displays the event scan log
  – `print-err-log`: displays error log information

*Example 4-4 nvram command option to extract event logs*

```
p630sles:/ # nvram --print-event-scan
Number of Logs: 7
Log Entry 0:  flags: 0x00  type: 0x52
```

```
        Severity: WARNING
        Disposition: FULLY RECOVERED
        Extended error type: 3
   c6 00 00 08 21 32 58 00 20 03 10 23 00 00 00 00 |....!2X....#....|
Log Entry 1:  flags: 0x00  type: 0x52
        Severity: WARNING
        Disposition: FULLY RECOVERED
        Extended error type: 3
   c6 00 00 08 22 05 02 00 20 03 10 16 00 00 00 00 |...."...........|
Log Entry 2:  flags: 0x00  type: 0x52
        Severity: WARNING
        Disposition: FULLY RECOVERED
        Extended error type: 3
   c6 00 00 08 23 42 52 00 20 03 10 08 00 00 00 00 |....#BR.........|
Log Entry 3:  flags: 0x00  type: 0x52
        Severity: NO ERROR
        Disposition: FULLY RECOVERED
        Extended error type: 0
Log Entry 4:  flags: 0x00  type: 0x52
        Severity: NO ERROR
        Disposition: FULLY RECOVERED
        Extended error type: 0
Log Entry 5:  flags: 0x00  type: 0x52
        Severity: NO ERROR
        Disposition: FULLY RECOVERED
        Extended error type: 0
Log Entry 6:  flags: 0x00  type: 0x52
        Severity: NO ERROR
        Disposition: FULLY RECOVERED
        Extended error type: 0number: 19
```

- **snap**

  The **snap** command in Linux for pSeries provides functionality that is similar to that of the snap command in AIX. For example, it captures your /var/log/messages, your device-tree inside the /proc file system, /proc tuning, /dev/nvram and your yaboot.conf. It gzips the file for IBM technical support to analyze.

- **update_flash**

  This command allows you to update the pSeries firmware directly from Linux. If you are updating the firmware for an LPAR, the respective LPAR requires “service authority” capability. There are cases where you need to download the latest firmware to give your service processor more intelligence and more capability. You can specify it with the option **-f** and the firmware file that you have just downloaded.

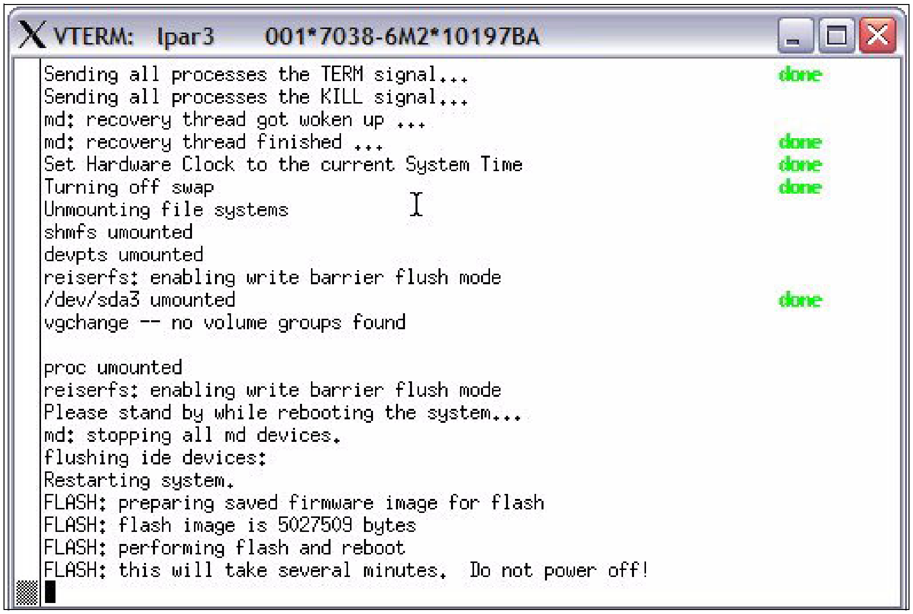
  Remember to run the checksum (using the command **sum**) on the firmware file prior to installation. This ensures that you can download the proper and complete file. Incomplete downloads can be disastrous and could corrupt your server. You can download the latest firmware from the Web site:

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

While you are updating your firmware, you will see that the operator panel of LPAR will indicate that it is in the process of flashing.

Figure 4-3 shows you what you will see after the **`update_flash`** command is run.

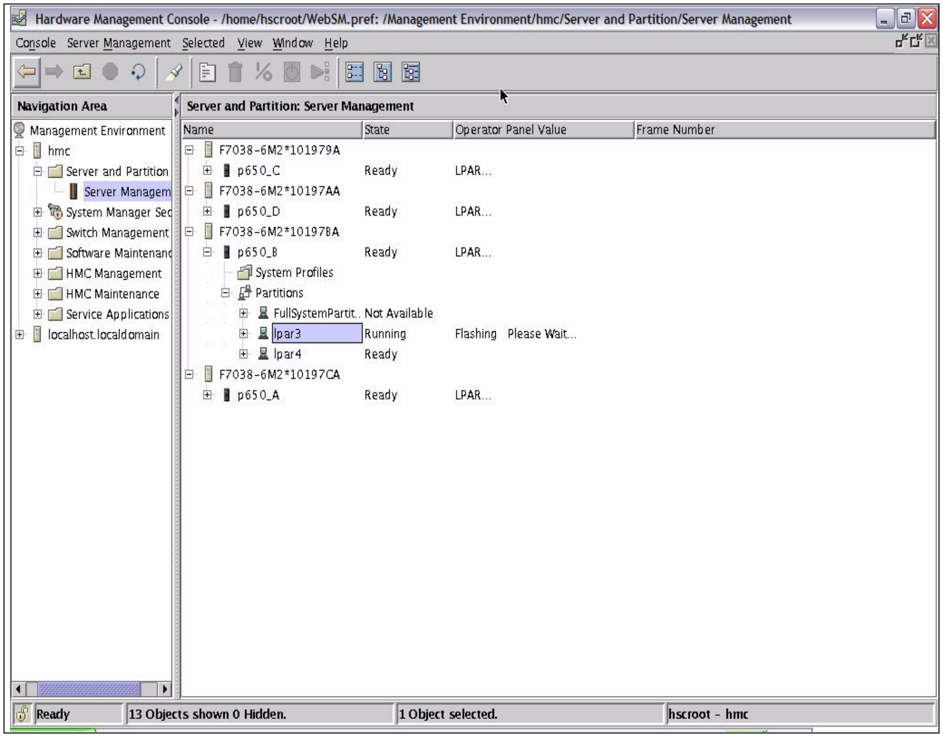
Refer to the IBM Redbook *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038, for information on how to set service authority.



```
Sending all processes the TERM signal...                             done
Sending all processes the KILL signal...
md: recovery thread got woken up ...
md: recovery thread finished ...                                     done
Set Hardware Clock to the current System Time                        done
Turning off swap                                                     done
Unmounting file systems
shmfs umounted
devpts umounted
reiserfs: enabling write barrier flush mode
/dev/sda3 umounted                                                   done
vgchange -- no volume groups found

proc umounted
reiserfs: enabling write barrier flush mode
Please stand by while rebooting the system...
md: stopping all md devices.
flushing ide devices:
Restarting system.
FLASH: preparing saved firmware image for flash
FLASH: flash image is 5027509 bytes
FLASH: performing flash and reboot
FLASH: this will take several minutes.  Do not power off!
```

Figure 4-3 Updating of firmware using the update_flash command

Corresponding to the above screen, you will notice that the Operator Panel of the LPAR will show as “Flashing”. Figure 4-4 on page 172 shows the Operator Panel.



| Name | State | Operator Panel Value | Frame Number |
|---|---|---|---|
| F7038-6M2*101979A | | | |
| p650_C | Ready | LPAR... | |
| F7038-6M2*10197AA | | | |
| p650_D | Ready | LPAR... | |
| F7038-6M2*10197BA | | | |
| p650_B | Ready | LPAR... | |
| System Profiles | | | |
| Partitions | | | |
| FullSystemPartit... | Not Available | | |
| lpar3 | Running | Flashing Please Wait... | |
| lpar4 | Ready | | |
| F7038-6M2*10197CA | | | |
| p650_A | Ready | LPAR... | |

*Figure 4-4 Operator in HMC showing Firmware Flashing in progress*

- **`lscfg`**

  This command lists all the hardware information that is available in the system.

  – `v`: prints out in verbose mode
  – `vp`: prints out in details including vendor specific information

  The following example shows how to query number of processors:

```
# lscfg -v | grep proc
p630sles:/usr/src/linux # lscfg -v | grep proc
  proc0          U0.1-P1-C1       Processor
  proc1          U0.1-P1-C2       Processor
```

- **`diagela`**

  This command is part of the error log analysis tool that provides automatic analysis and notification errors reported by the RunTime Abstraction Service on the pSeries hardware. When an error is detected and corrective actions are required, notification is automatically sent to the Service Focal Point on the Hardware Management Console, or to users specified in the

/etc/diagela/mail_list configuration file. At the same time, the logs of the analysis are inside /var/log/messages file.

Figure 4-5 shows what is happening at the background of the diagela application.

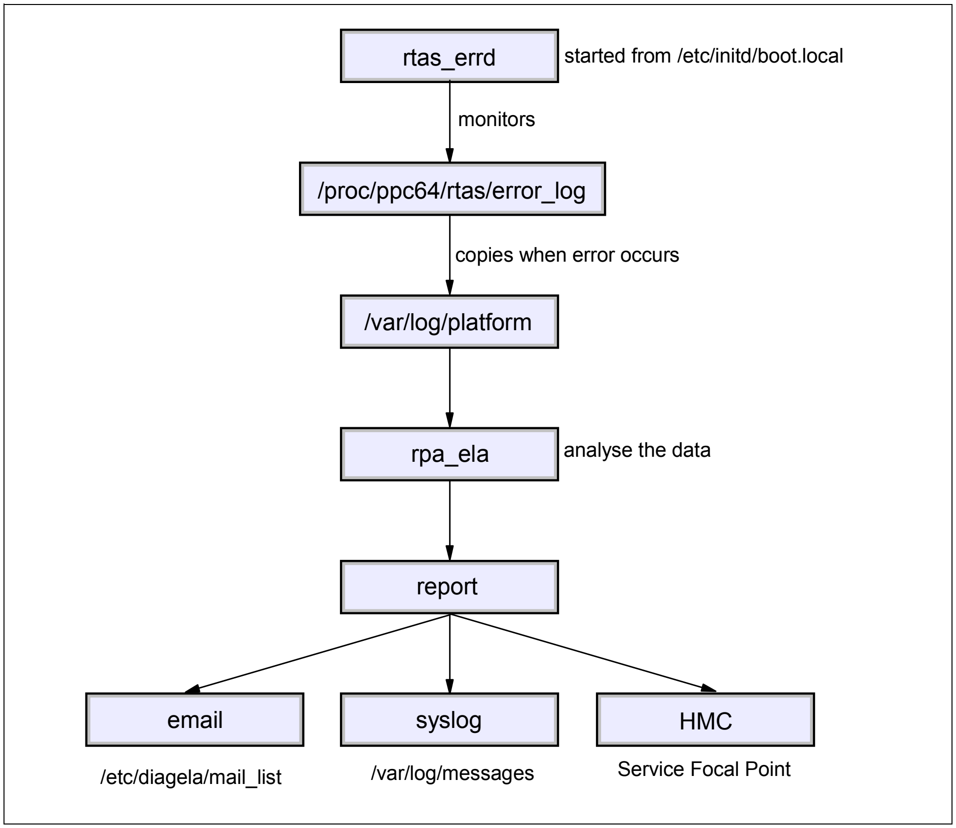


*Figure 4-5 Flowchart of the diagela analysis tool in Linux*

Whenever the rtas_errd background daemon scans and detects any error reported by the system firmware, it basically activates the analysis program to deduce what kind of problem it is facing.

After analysis, it reports it back to the system logs and the respective mechanism that you have configured for it to use. Example 4-5 on page 174 shows the analysis of power failure output from the diagela daemon.

*Example 4-5 /var/log/messages showing analysis of power failure of the system*

```
Diagela for Linux for pSeries
Oct 28 14:29:41 lpar1 diagela: 10/28/2003 14:29:40
Oct 28 14:29:41 lpar1 diagela: Automatic Error Log Analysis reports the 
following:
Oct 28 14:29:41 lpar1 diagela:
Oct 28 14:29:41 lpar1 diagela: 651204 ANALYZING SYSTEM ERROR LOG
Oct 28 14:29:41 lpar1 diagela: A loss of redundancy on input power was 
detected.
Oct 28 14:29:41 lpar1 diagela:
Oct 28 14:29:41 lpar1 diagela: Check for the following:
Oct 28 14:29:41 lpar1 diagela:   1. Loose or disconnected power source 
connections.
Oct 28 14:29:41 lpar1 diagela:   2. Loss of the power source.
Oct 28 14:29:41 lpar1 diagela:   3. For multiple enclosure systems, 
loose or
Oct 28 14:29:41 lpar1 diagela:      disconnected power and/or signal 
connections
Oct 28 14:29:41 lpar1 diagela:      between enclosures.
Oct 28 14:29:41 lpar1 diagela:
Oct 28 14:29:41 lpar1 diagela: Supporting data:
Oct 28 14:29:41 lpar1 diagela:       Ref. Code: 10111520
Oct 28 14:29:41 lpar1 diagela:
Oct 28 14:29:41 lpar1 diagela: Analysis of /var/log/platform sequence 
number: 3
```

## 4.2 System logs

System logs are critical components of an operating system that allow us to track and debug errors. They also provide more in-depth information about system performance and behavior.

### 4.2.1 Linux syslog

The system logging utility that we use in Linux is called “syslog”. The syslog bundled inside Linux is derived from BSD sources and is bundled natively into the operating system. It is turned on by default on Linux. This tool allows you to trap kernel messages and capture system messages, with capabilities to customize the level of criticality that you choose to capture.

Syslog provides two daemons as part of the package, syslogd and klogd. The syslogd daemon provides the system logging facilities, and klogd provides the kernel logging facility.

Daemons are processes that runs continuously in the operating system and are started during the runlevel processing at boot time (for example, /etc/rc.d/rc<N>.d/S06syslog).

The core configuration file in the syslog configuration is located in /etc/syslog.conf. It lists the types of errors, levels, and where in the file the errors be directed to.

The /etc/syslog.conf file contains logging parameters in the form of application.levelofseverity. The syslog.conf allows multiple entries of the same application error in different lines or separated by semicolons, as shown in the sample syslog configuration file in Example 4-6.

*Example 4-6 Sample of /etc/syslog configuration file*

```
# /etc/syslog.conf
kern.warn;*.err;authpriv.none /dev/tty10
kern.warn;*.err;authpriv.none|/dev/xconsole

*.emerg       *
*.*         /dev/tty12
mail.*      -/var/log/mail

news.crit   -/var/log/news/news.crit
news.err    -/var/log/news/news.err
news.notice -/var/log/news/news.notice

*.=warn;*.=err;*.crit-/var/log/warn
*.*;mail.none;news.none-/var/log/messages
```

The incoming logs are basically logged to files accordingly to their classification. Some of the common application classifications are listed in Table 4-1.

*Table 4-1 Application classification in syslog*

| | |
|---|---|
| auth | Authentication programs like login, telnet, ssh |
| authpriv | Authentication privileges |
| cron | cron daemons |
| daemon | Any other daemons which did not fall into the standard list |
| ftp | File transfer protocol service |
| kern | Linux kernel itself |
| lpr | Line printing service |

| | |
|---|---|
| auth | Authentication programs like login, telnet, ssh |
| mail | Mail service daemon |
| news | News service daemon |
| security | Miscellaneous security application |
| syslog | Syslog daemon |
| user | Generic User Level messages |
| uucp | UNIX-to-UNIX copy service application |
| local0-7‘ | Use by any application or daemons to write to local console |

Based on the application classification, you can filter the messages based on their severity levels. Table 4-2 lists some of the severity levels that you can define inside your /etc/syslog.conf.  They are listed in increasing severity.

*Table 4-2   Severity levels in syslog*

| | |
|---|---|
| none | Do not log message |
| debug | Debugging messages |
| info | Informational messages |
| notice | Notice which denote something is not amiss |
| warning | Warning condition |
| err | Error condition |
| crit | Critical errors that should be checked immediately |
| alert | Severe error |
| emerg | A unrecoverable error has occurred. If this occurred at the kernel space, it is often followed by kernel panic or your system could risk corruption. |

## System logs in a centralized environment

In an environment with more than two servers, syslog can be configured to allow servers to forward syslogs to a central server where all the system logs can be stored. Syslog allows this operation by forwarding the logs from the client to the server through the port 514/UDP.

First, you need to configure the syslog server to accept incoming logs. By default, this is turned off.

### Server side

In the server side:

- Update /etc/sysconfig/syslog file under the SYSLOGD_PARAMS option.

  SYSLOGD_PARAMS=”-r”

- Restart the syslog service by using the command **/etc/init.d/syslog restart** or you can use the command **kill -HUP ‘cat /var/run/syslogd.pid‘**

Now you need to configure the client to forward the logs to the server. You can still have logs stored in the client machines as if it is a standalone server.

### Client side

In the client side:

- Update /etc/syslog.conf with the respective configuration.

  In Example 4-7, all the logs will be forwarded to the p630sles /var/log/messages log file. You can customize it further to forward only necessary facility and criticality.

*Example 4-7 Client syslog.conf*

```
# /etc/syslog.conf
kern.warn;*.err;authpriv.none /dev/tty10
kern.warn;*.err;authpriv.none|/dev/xconsole

*.emerg        *
*.*           /dev/tty12
mail.*        -/var/log/mail

news.crit     -/var/log/news/news.crit
news.err      -/var/log/news/news.err
news.notice  -/var/log/news/news.notice

*.=warn;*.=err;*.crit-/var/log/warn
*.*;mail.none;news.none-/var/log/messages
*.*;mail.none;news.none-p630sles@/var/log/messages
```

- Restart syslog service using the command **/etc/init.d/syslog restart**
- Test logging using the **logger** command:

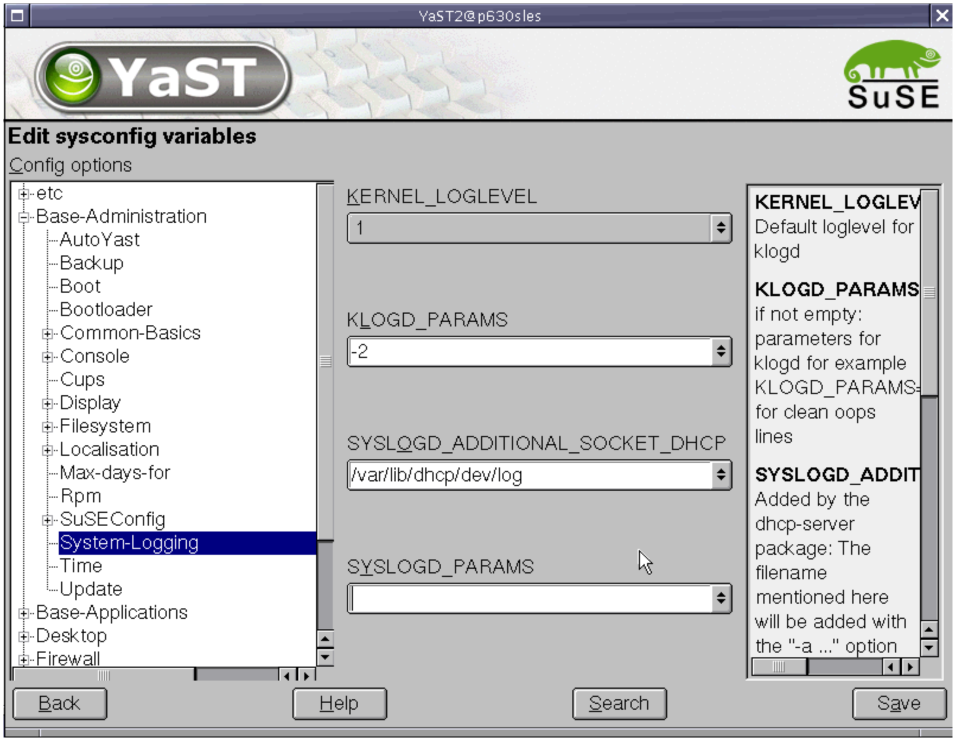
  **logger -p local0.crit -t TEST testing**

On the server side, check the /var/log/messages file. You should see that the “testing” message is logged as shown in Example 4-8 on page 178.

*Example 4-8 Example of logger test with output in the /var/log/messages*

```
   Oct 21 13:18:52 p630sles kernel: Kernel logging (proc) stopped.
   Oct 21 13:18:52 p630sles kernel: Kernel log daemon terminating.
   Oct 21 13:18:53 p630sles exiting on signal 15
   Oct 21 13:18:54 p630sles syslogd 1.4.1: restart (remote reception).
   Oct 21 13:18:59 p630sles kernel: klogd 1.4.1, log source = /proc/kmsg
   started.
   Oct 21 13:18:59 p630sles kernel: Inspecting
   /boot/System.map-2.4.21-83-pseries64
   Oct 21 13:18:59 p630sles kernel: Loaded 31486 symbols from
   /boot/System.map-2.4.21-83-pseries64.
   Oct 21 13:18:59 p630sles kernel: Symbols match kernel version 2.4.21.
   Oct 21 13:18:59 p630sles kernel: Loaded 110 symbols from 8 modules.
   Oct 21 13:19:44 lpar8 syslogd 1.4.1: restart.
   Oct 21 13:19:47 lpar8 TEST: testing
Oct 21 13:19:49 lpar8 kernel: klogd 1.4.1, log source = /proc/kmsg started.
```

Besides configuring syslog by hand, SLES8 bundles a graphical YaST2 tool to help you with the configuration. Figure 4-6 on page 179 shows the YaST2 utility for configuring system logging.



Edit sysconfig variables

Config options

KERNEL_LOGLEVEL: 1

KLOGD_PARAMS: -2

SYSLOGD_ADDITIONAL_SOCKET_DHCP: /var/lib/dhcp/dev/log

SYSLOGD_PARAMS:

**KERNEL_LOGLEV** Default loglevel for klogd

**KLOGD_PARAMS** if not empty: parameters for klogd for example KLOGD_PARAMS= for clean oops lines

**SYSLOGD_ADDIT** Added by the dhcp-server package: The filename mentioned here will be added with the "-a ..." option

*Figure 4-6 SuSE YaST2 syslog configuration*

**Tip:** If the error logs appear in IP addresses instead of hostname, add the respective server's hostname and IP address into the /etc/hosts of the syslog server and restart syslog.

## 4.3 Event logging - enterprise event log

Event Logging or evlog is a set of tools that implements an enterprise-level event logging facility, according the POSIX draft standard. evlog stores event records in binary and uses two daemons: evlogd and evlogrmtd. The evlogd daemon is used to log the events, and evlogrmtd is for clients to send remote logs for the evlog server.

The evlog installable RPM is located inside the first CD1 in the directory suse/RPMS/ppc/.

1. Install the evlog-1.5.3-36.rpm:

   ```
   # rpm -i evlog-1.5.3-36.rpm
   ```

2. Create an empty file inside /var/log/evlog/ called eventlog:

   ```
   # touch /var/log/evlog/eventlog
   ```

3. Start evlog services:

   ```
   # /etc/init.d/evlog start
   ```

   To enable forward kernel and system logs to the evlog:

   ```
   # /sbin/slog_fwd
   ```

The **evlog** command stores the files in the directory /var/log/evlog. Table 4-3 lists some of the commands you can use the query, modify or update evlog services and logs.

*Table 4-3 The evlog commands*

| | |
|---|---|
| evlview | Utility to view logs<br>example: **evlview -m** |
| evlconfig | Utility to configure logging daemon |
| evlfacility | Utility to manage facility registry |
| elvsend | Utility for Event Generation, useful for testing and application; for example: # elvsend -f user -t 1 -m “testing” |
| elvnotify | Utility for Event Notification |

> **Tip:** If Linux is behaving strangely after enabling the forwarding of system log using the command **/sbin/slog_fwd**, you can disable it by using **/sbin/slog_fwd -r** or you can remove the file libevlsyslog.so.1 from /etc/ld.so.preload and run **/sbin/ldconfig**. This will remove the system logging forwarding service to evlog.

### 4.3.1 evlog in a centralized environment

The remote event consolidator, or evlogrmtd, is the daemon that evlog runs to accept incoming events from other hosts in the network. All the incoming events will be consolidated and stored into a single file for easy retrieval and searching.

You optionally set filters to check for certain more alarming words like “crit”, “emerg”, “error” and others.

First, you need to configure the server to listen for incoming logs and to allow specific clients for access.

### Server side

On the server side, perform these steps:

1. Login as root and create a new evlogrmtd.conf file from the template available from /etc/evlog.d/evlogrmtd.conf.sample. Example 4-9 shows what should be inside the evlogrmt.conf file.

*Example 4-9 Sample of /etc/evlog.d/evlogrmt.conf file*

```
   # evlogrmtd.conf file syntax
   Password=<cleartextpassword>
   TCPPort=12000
  UDPPort=34000
```

2. Update /etc/evlog.d/evlhosts similar to what is shown in Example 4-10 for each client that you want evlog to receive events from. Each entry must have a resolvable hostname and also a unique identifier for each host. If you leave the identifier blank, it will be ignored by evlog.

*Example 4-10 Sample of /etc/evlog.d/evlhosts file*

```

   #evlhosts file syntax
   #
   # hostid hostname
   # ----------------
   # 1    mylinuxbox
   # 10.128 mylinuxbox2
100.79  lpar3
```

   Change the permissions to read and write only for root.

```
# chmod 700 /etc/evlog.d/evlhosts
```

3. Restart the evlog daemon and check that the evlogrmtd daemon is running.

```
# /etc/init.d/evlog restart
# ps -ef | grep evlogrmtd
```

In the client side, you will need to configure it to send logs in the corresponding port that the server is listening on.

## Client side

On the client side, perform these steps:

1. Login in as root and go into the /etc/evlog.d/ directory.
2. If you plan to use UDP, edit udp_rmtlog_be.conf to specify the IP address of the server and Port number; see Example 4-11.

*Example 4-11 udp_rmtlog_be.conf configuration file*

```
# udp_rmtlog_be.conf file syntax
Remote Host=192.168.100.84
Port=34000
Disable=no
```

If you plan to use TCP, edit tcp_rmtlog_be.conf to specify the IP address of the server, Port number, Password and BufferLenInKbytes, as shown in Example 4-12.

*Example 4-12 tcp_rmtlog_be.conf*

```
# tcp_rmtlog_be.conf file syntax
Remote Host=192.168.100.84
Password=<clear_textpassword_from_management_server>
Port=12000
BufferLenInKbytes=128
Disable=no
```

3. Finally, test whether your management host can receive the logging message from the newly added client.

   ```
   # evlsend -f user -t 1 -m testing from lpar3
   ```

   You should see the following at the evlog management server:

   ```
   # evlview -m


   Oct 22 15:41:14 lpar8 tcp_rmtlog_be: plugin is disabled.
   Oct 22 15:41:14 lpar8 tcp_rmtlog_be: plugin unloaded.
   Oct 22 15:41:14 lpar8 udp_rmtlog_be: plugin is disabled.
   Oct 22 15:41:14 lpar8 udp_rmtlog_be: plugin unloaded.
   Oct 22 17:12:59 lpar8 evlogrmtd: This 192.168.100.79 host is successfully
   authenticated.
   Oct 22 17:13:06 lpar3 testing from lpar3
   ```

## 4.4 Log rotation

Most of daemons like syslog append their events and logs into a file. As a result, the file size will get larger over time. To avoid this, Linux uses logrotate to manage logs. Logrotate is run daily by the cron; it can create and then zip up the files into their respective folders. The main configuration file of logrotate is located at /etc/logrotate.conf. Example 4-13 shows logrotate.conf configuration file in a standard SLES 8 server.

*Example 4-13 Sample of logrotate configuration file*

```
# /etc/logrotate.conf configuration file

# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# uncomment these to switch compression to bzip2
#compresscmd /usr/bin/bzip2
#uncompresscmd /usr/bin/bunzip2
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own lastlog or wtmp -- we'll rotate them here
#/var/log/wtmp {
#    monthly
#    create 0664 root utmp
#    rotate 1
#}
# system-specific logs may be also be configured here.
```

Any application that you would like logrotate to manage should be created as a file and placed into the directory /etc/logrotate.d/; it will be automatically run by the cron daemon on daily basis, /etc/cron.daily.

# 4.5 Linux rescue methods

You may often face problems with boot loaders, file systems, configuration files and so on. These problems may, in some cases, prevent you from booting up the system for fixing or debugging. In this section, we discuss common problems that you may encounter, and describe solutions. Figure 4-7 shows problem determination flow.

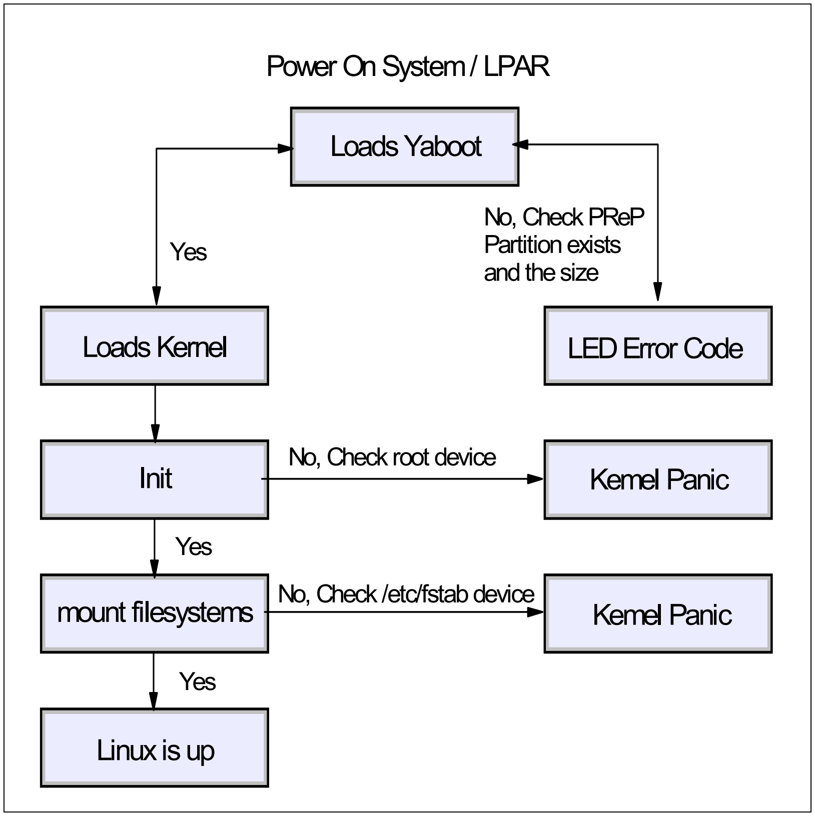


*Figure 4-7 Flow chart of debugging Linux for pSeries*

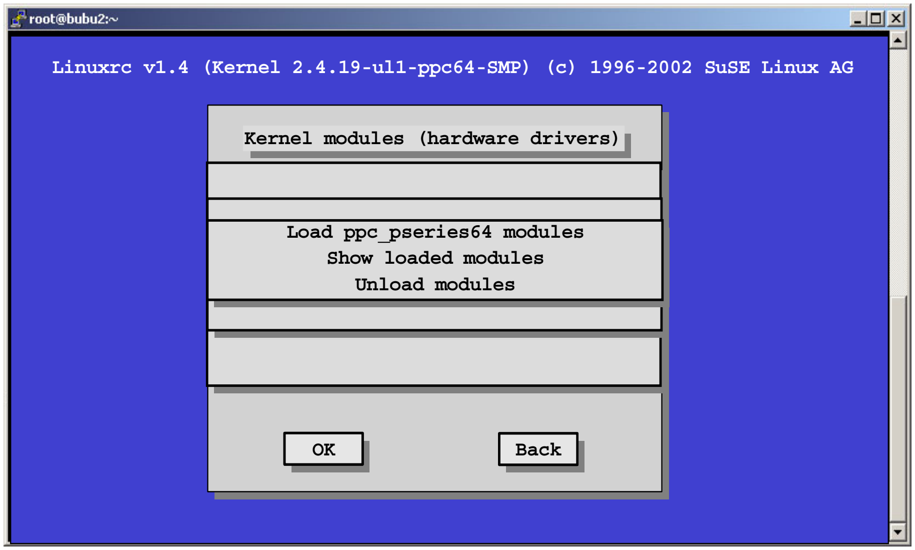
## 4.5.1 Boot loader corruption

The most common problem in Linux for pSeries that administrators face is that the system is unable to boot after installing. This could also happen if you have accidentally overwritten your boot loader. If the boot loader in the PReP partition called yaboot is corrupted, you will need to create another a new partition and reactivate it.

### PReP boot loader corrupted

Before diagnosing boot loader corruption, make sure the system or LPAR boots up to the E1F1 LED panel, and proceed to load the respective boot loader from the disk.

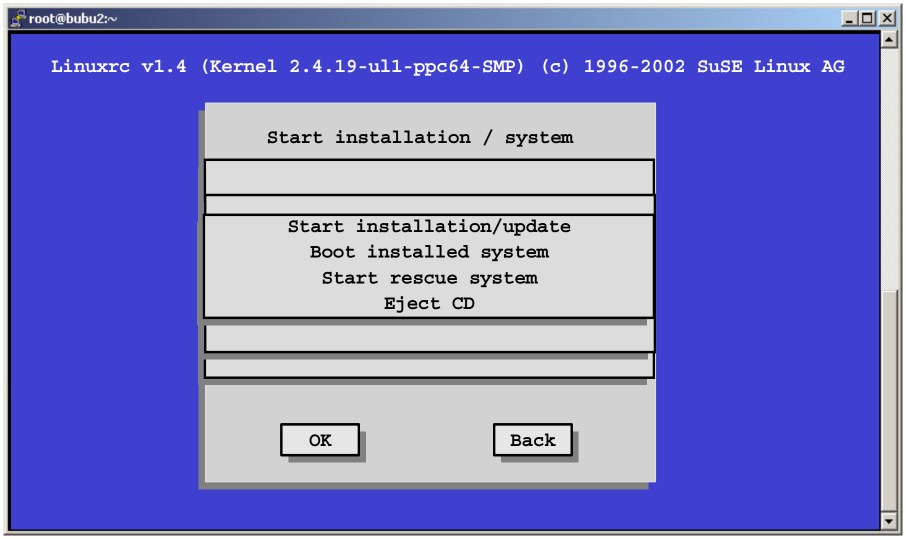
1. Boot from the network or from CD-ROM and load the appropriate driver that you may need for your system. Refer to 2.1.4, “Review your choices” on page 23 for the basic device drivers used by adapters and Linux on pSeries.
2. Load the all the modules that are needed for your system to load, as shown in Figure 4-8. For example, if you have an SCSI external CD-ROM that you plan to use to boot the installation, you will need to load the driver for the SCSI card.



```
Linuxrc v1.4 (Kernel 2.4.19-ul1-ppc64-SMP) (c) 1996-2002 SuSE Linux AG
```

*Figure 4-8 Loading modules from the SuSE Installer*

3. After loading the required module, you will then select the **Start Rescue System** shown in Figure 4-9 on page 186, and then select the location of your kernel. You have the choice of booting from CD, network, hard disk or floppy. When boot into the rescue mode, the SLES Installer will give you a small Linux operating system located in the ramdisk. With this, you will be given a “Rescue” prompt with full root access.



```
Linuxrc v1.4 (Kernel 2.4.19-ul1-ppc64-SMP) (c) 1996-2002 SuSE Linux AG

Start installation / system
```

*Figure 4-9 Booting into rescue mode for recovery*

4. Do a file system check on your disk:

   ```
   Rescue :/ # fsck.reiserfs /dev/<disk>
   ```

5. Create a new partition with the command **fdisk** and set the type of the partition to PReP Boot (ID 41) and active boot device. The size of the partition is recommended not to exceed 8 Mb in size, as it will only contain the image that will be used to boot the system.

   ```
   Rescue :/ # fdisk /dev/<disk>
   ```

   Select option “n” and add a primary partition. If you have an existing partition, use the option “d” to delete it first. Make sure the size created is less than 8 Mb. Then use option “t” to change the boot type to 41. 41 is the boot type for PPC PReP Boot.

6. Recreate the PReP Boot image using the **dd** command:

   ```
   Rescue :/ # dd if=/boot/yaboot.chrp of=/dev/<disk> bs=4k
   ```

7. Reboot the system. If everything goes right, you should now be able to boot your system without any problem.

**Tip:** If you have a DHCP and NFS server, you can place the zImage.initrd.ppc64-2.4.21 kernel file into the server. The file is available from the first SLES8 CD. Set the server or LPAR to boot from this kernel image. In this way, you can rescue or boot your system even if the PReP boot partition is corrupted.

## 4.5.2 File system corruption

Very often, when the server did not shut down properly, the file systems or file could risk corruption. Although a journaled file system can help in many cases, it is not foolproof. There are cases where you might need to rebuild the logs and database structure.

### File system corruption

1. If you have file system corruption or configuration file corruption, you can boot the system into single user mode. If it is your root partition that is corrupted, skip this step and proceed to step 3.

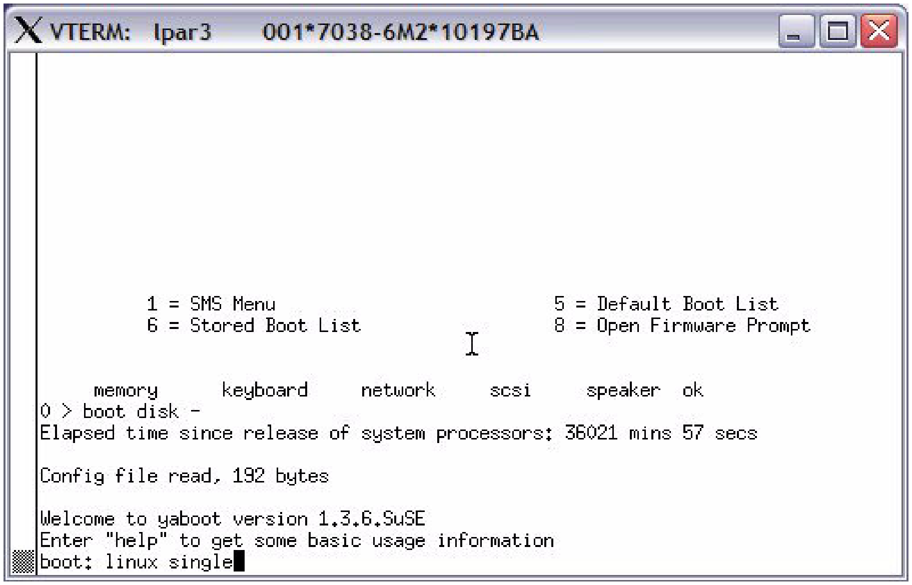
   If you choose not to use yaboot for booting automatically (for example, in dual-boot systems), you should still create the PReP boot loader, but it must be not active. You can boot up your system to the openfirmware prompt and the pass the respective parameter to the yaboot prompt:

   ```
   0> boot disk
   ```

   When it reaches the yaboot prompt, key in the following:-

   ```
   yaboot : linux single console=hvc0
   ```

   Figure 4-10 on page 188 shows the diagram of booting the disk from the openfirmware prompt to the yaboot prompt.



```
        1 = SMS Menu                             5 = Default Boot List
        6 = Stored Boot List                     8 = Open Firmware Prompt


     memory      keyboard     network     scsi     speaker  ok
0 > boot disk -
Elapsed time since release of system processors: 36021 mins 57 secs

Config file read, 192 bytes

Welcome to yaboot version 1.3.6.SuSE
Enter "help" to get some basic usage information
boot: linux single
```

*Figure 4-10 Boot up system into rescue from open firmware*

2. Running file system check on the file system:

   ```
   (none):~ # fsck.reiserfs /dev/<disk>
   ```

3. If this did not fix your problem, you will need to use the first CD1 from the SLES and boot into rescue mode. After boot into rescue mode, rerun the **fsck.reiserfs** command. If you are using other file systems, the command will differ.

   After that, create a new mount point in the rescue system and then mount your file systems over it.

4. Mount the root file system into a mount point:

   ```
   Rescue:/ # mount -t reiserfs /dev/sda2 /mnt/<mount_point>
   ```

5. If necessay, you can modify and update /etc/fstab accordingly so that it can boot up correctly.

Should you need to reset the root password, you can change the root directory and provide the root password accordingly.

```
Rescue:/ # chroot /mnt/root
Rescue:/ # passwd root
```

## 4.5.3 RHAS 3 rescue mode

For RHAS 3, you can use the installation disk in rescue mode to provide quick access to your disk partition to perform recovery and changes for your corrupted Linux system. To boot up into rescue mode, boot up the CD-ROM until the yaboot prompt:

```
yaboot : linux rescue
```

If you do not have a CD-ROM attached to the system, you can boot up the system into open firmware and run the following. You also press the key 8 when the LED shows E1F1; this will get you to the open firmware prompt as well.

```
0 > boot net rescue
```

Once the kernel is loaded, select the language and the location of the rescue image. Then the installation program will attempt to mount the disk partition on your system. It will presents you with a shell prompt, where you can perform the necessary rescue methods. To exit, type: `exit 0`; this will automatically reboot the system.

Refer to 2.3.3, “Unattended installation” on page 67 for information about how to set up the network boot.

## 4.5.4 Using /proc file systems

The /proc file system in Linux provides real-time information about the kernel and the hardware devices that are present in the server. Some of these are read-only and others are read-write which allows you to modify or tune the hardware for better performance. Refer to “File system tuning” on page 208 for information on how to tune your system using /proc.

Some commonly used commands on Linux are listed in Table 4-4 on page 190.

*Table 4-4 Commands used on Linux*

| | |
|---|---|
| # **procinfo** | Provides a brief overview of the system. |
| # **cat /proc/cpuinfo** | Display CPU information, clock speed. |
| # **cat /proc/ppc64/lparcfg** | Display a snapshot of the current LPAR configuration; this is useful for the server attached to HMC.<br># **cat /proc/ppc64/lparcfg**<br>`serial_number=IBM,xxxxxxxx`<br>`system_type=IBM,7038-6M2`<br>`partition_id=1`<br>`system_active_processors=2`<br>`system_potential_processors=2`<br>`partition_active_processors=2`<br>`partition_potential_processors=2`<br>`partition_entitled_capacity=200`<br>`partition_max_entitled_capacity=400`<br>`shared_processor_mode=0` |
| # **hwscan --list** | Display the list system devices and adapters with classification of the type of device. |
| # **hwinfo** | Display detailed output of the complete hardware that is currently in the server. This is very useful for administrators trying to analyze and debug hardware problems. |

In addition to displaying details about the devices in /proc, you can also use /proc to remove and add SCSI devices on the fly. To list the SCSI devices you have on your system, run the command **cat /proc/scsi/scsi**. Example 4-14 shows the output of the command.

*Example 4-14 Display SCSI devices from /proc*

```
lpar7:/proc/scsi # cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: IBM      Model: CDRM00203     !K Rev: 1_06
  Type:   CD-ROM                           ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 08 Lun: 00
  Vendor: IBM      Model: IC35L146UCDY10-0 Rev: S25F
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi0 Channel: 00 Id: 09 Lun: 00
  Vendor: IBM      Model: IC35L146UCDY10-0 Rev: S25F
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi0 Channel: 00 Id: 14 Lun: 00
```

```
  Vendor: IBM      Model: HSBPM2   PU2SCSI Rev: 0015
  Type:   Enclosure                        ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 15 Lun: 00
  Vendor: IBM      Model: HSBPD4M  PU3SCSI Rev: 0015
  Type:   Enclosure                        ANSI SCSI revision: 02
```

You have a number of attached devices after the output shown in Example 4-14. The first line describes the how the hardware are being connected, followed by the vendor and the type of device. Existing devices can be removed using the command **`echo “scsi remove-single-device <h> <b> <t> <l>” > /proc/scsi/scsi`** where <h> is the host adapter, <b> for channel id, <t> for scsi target id and <l> for lun. After this, run the command **`cat /proc/scsi/scsi`** to see if the remove was successful. Example 4-15 shows removing a single scsi disk ( /dev/sdb ).

*Example 4-15   Removing a single device in Linux*

```
lpar7:/proc/scsi # fdisk -l

Disk /dev/sda: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1         1      8001   41  PPC PReP Boot
/dev/sda3            15       537   4200997+  83  Linux
/dev/sda4           538     17848 139050607+   5  Extended
/dev/sda5           538       799   2104483+  82  Linux swap
/dev/sda6           800     17848 136946061   fd  Linux raid autodetect

Disk /dev/sdb: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
lpar7:/proc/scsi # echo "scsi remove-single-device 0 0 9 0" > /proc/scsi/scsi
lpar7:/proc/scsi # fdisk -l

Disk /dev/sda: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1         1      8001   41  PPC PReP Boot
/dev/sda3            15       537   4200997+  83  Linux
/dev/sda4           538     17848 139050607+   5  Extended
/dev/sda5           538       799   2104483+  82  Linux swap
/dev/sda6           800     17848 136946061   fd  Linux raid autodetect
```

You can also add a new SCSI device by using the command echo “scsi add-single-device <h> <b> <t> <l>” > /proc/scsi. In Example 4-16, we add the SCSI disk we removed in Example 4-15 back to the system.

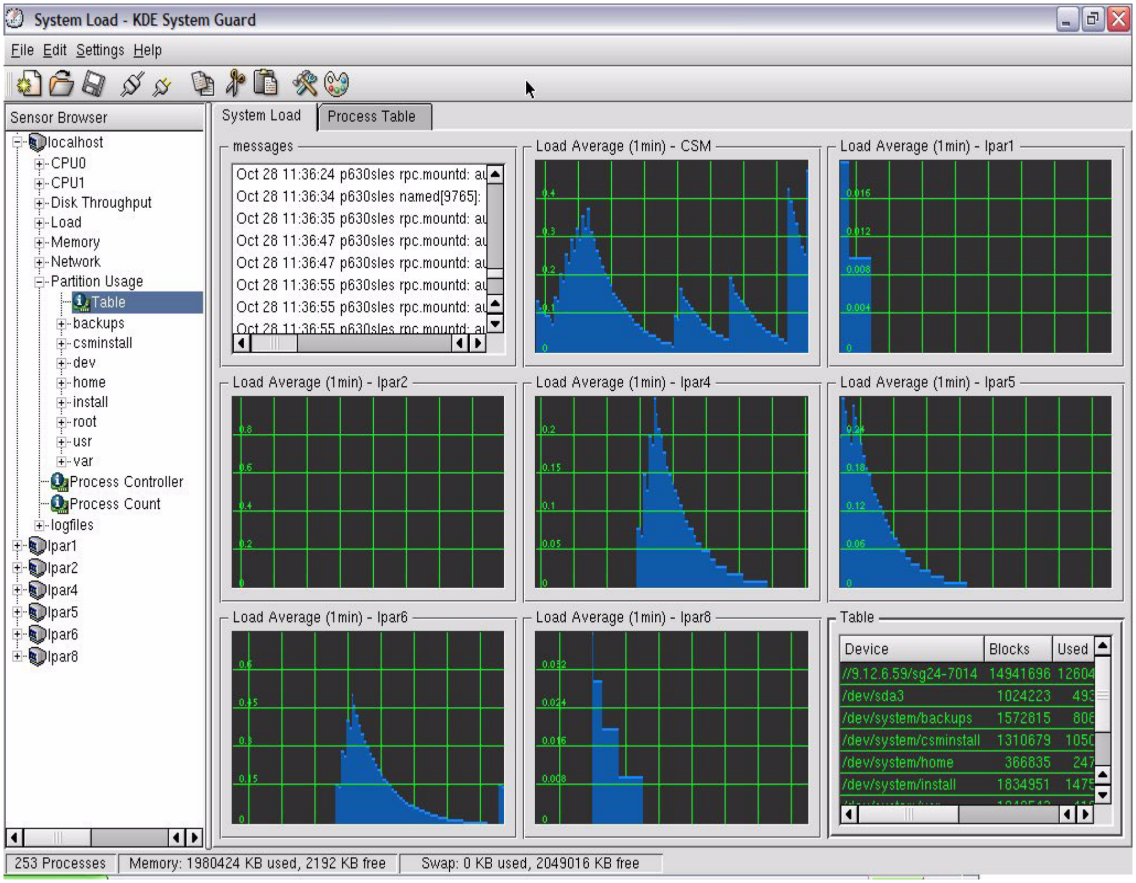
*Example 4-16 Adding a SCSI disk*

```
lpar7:/proc/scsi # echo "scsi add-single-device 0 0 9 0" > /proc/scsi/scsi
lpar7:/proc/scsi # fdisk -l

Disk /dev/sda: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1         1      8001   41  PPC PReP Boot
/dev/sda3            15       537   4200997+  83  Linux
/dev/sda4           538     17848 139050607+   5  Extended
/dev/sda5           538       799   2104483+  82  Linux swap
/dev/sda6           800     17848 136946061   fd  Linux raid autodetect

Disk /dev/sdb: 255 heads, 63 sectors, 17849 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
```

# 4.6 Performance monitoring

Monitoring allows you to get more out of your operating system by helping you determine the types of applications and their behavior in the system. TLinux bundles a good monitoring tool natively within the operating system. A good starting point is to use standard tools such as ps, top, vmstat, sar and iostat. These applications are easy to use and provide administrators with a quick glance at system performance.

In addition, you can also use bundled tools like ksysguard, which can consolidate multiple systems and display it in a single window. Figure 4-11 on page 193 shows multiple LPARs that were added to the ksysguard window. It shows the average load of all the servers.



```
Oct 28 11:36:24 p630sles rpc.mountd: au
Oct 28 11:36:34 p630sles named[9765]:
Oct 28 11:36:35 p630sles rpc.mountd: au
Oct 28 11:36:47 p630sles rpc.mountd: au
Oct 28 11:36:47 p630sles rpc.mountd: au
Oct 28 11:36:55 p630sles rpc.mountd: au
Oct 28 11:36:55 p630sles rpc.mountd: au
Oct 28 11:36:55 p630sles rpc.mountd: au
```

| Device | Blocks | Used |
|---|---|---|
| //9.12.6.59/sg24-7014 | 14941696 | 12604 |
| /dev/sda3 | 1024223 | 493 |
| /dev/system/backups | 1572815 | 806 |
| /dev/system/csminstall | 1310679 | 1050 |
| /dev/system/home | 366835 | 247 |
| /dev/system/install | 1834951 | 1475 |

253 Processes | Memory: 1980424 KB used, 2192 KB free | Swap: 0 KB used, 2049016 KB free

*Figure 4-11 KDE's ksysguard showing multiple servers being monitored*

## 4.6.1 Performance monitoring with ganglia

Ganglia[1] is very popular in the Linux community for monitoring the performance and usage of large clusters, up to 2000 nodes. You can see ganglia working live from Berkeley University:

```
http://monitor.millennium.berkeley.edu/
```

The performance data is extracted locally on each node by gmond, which is a monitoring daemon. One of the nodes, or a separate system, is used to collect the data with gmetad, which is the collector daemon that is then stored on a Web server. The data is stored in XML format with RRDtool[2], which is a round-robin

[1] http://ganglia.sourceforge.net/
[2] Tobi Oetiker, http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/

database tool that is used to accumulate the data for a given period of time, generally a few hours. To visualize the data, all you need to do is point a browser to the collector machine. All the scripting in this server is done in PHP.

A single collector machine can visualize data from nodes running various operating systems and of various architectures. You can use ganglia to monitor a pSeries cluster and an xSeries cluster, both running Linux, for example.

In the following sections, we describe how to get the source code, make RPMs, and install the gmond and gmetad part. Then we describe the Web server setup and how to get going.

## Downloading ganglia

The current stable version of ganglia at the time of writing is 2.5.5. Go to the following Web site and follow the instructions for downloading the ganglia-monitor-core-2.5.X.tar.gz file, as well as the Web front-end part ganglia-webfrontend-2.5.X-1.noarch.rpm:

`http://ganglia.sourceforge.net/downloads.php`

You can download a pre-build ganglia RPM for Linux on pSeries from the below site.

`ftp://www.redbooks.ibm.com/redbooks/SG247014/`

## Compiling ganglia

The gmond and gmetad need to be compiled; the Web front-end is using PHP scripts, which we do not need to recompile as SLES8 bundles PHP. In this section, we show how to compile and build an RPM to ease the deployment of ganglia in your cluster. There are many references on the Web regarding RPM[3]. You need the gcc compiler installed.

Building ganglia on SLES 8 is much easier than on RHAS 3, as the RRDtool is provided by SuSE. On RHAS 3, you must download and install it before compiling ganglia. Nevertheless, we describe both distributions. It is also perfectly valid to build on SLES 8 for deployment on RHAS 3 and vice versa. However, for the sake of simplicity, we assume you are building and deploying on the same distribution. Once you have built and installed your RPM, whichever the distribution, you can monitor a cluster with nodes from any distribution. You can also monitor nodes of different architecture.

To create RPMs, we strongly recommend using a regular account. If your user account is not already set to build RPMs, issue the commands shown in Example 4-17 on page 195.

[3] Edward C. Bailey, Maximum RPM http://www.rpm.org/max-rpm/

*Example 4-17 Be ready for building RPM s*

```
[leecy@p630]> mkdir ~/build
[leecy@p630]> for i in RPMS SRPMS SPECS SOURCES BUILD
do
mkdir ~/build/$i
done
[leecy@p630]> echo “%_topdir /home/leecy/build” > ~/.rpmmacros
```

The ganglia tarball contains a RPM spec file that works well, so we will extract it and use it. RPM spec files describe all the commands needed to:

- Set up the source code before compiling
- Compile the various binaries and libraries
- Pack the necessary files in the RPM file
- Install and configure the RM
- Unconfigure and uninstall the RPM

We extract the spec file and save it to the right place after copying the tarball in the build/SOURCES/ directory, as shown in Example 4-18.

*Example 4-18 Extract the spec file*

```
[leecy@p630]> cp ganglia-monitor-core-2.5.5.tar.gz ~/build/SOURCES/.
[leecy@p630]> cd ~/build/SPECS
[leecy@p630]> tar zxf ../SOURCES/ganglia-monitor-core-2.5.5.tar.gz
[leecy@p630]> mv ganglia-monitor-core-2.5.5/ganglia.spec ./
[leecy@p630]> rm -rf ganglia-monitor-core-2.5.5
```

If you run SLES 8, you can skip the next section.

### Building RPM for RHAS 3

For RHAS 3, you need to install the rpm-build package. Unlike SLES 8, the command to install and query packages (rpm) is different from the command used to create packages (rpmbuild). It is on RHAS 3 CD3.

You need to download and compile the RRDtool before compiling ganglia. This one is not provided by Red Hat. The URL for downloading RRDtool is:

```
http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/rrdtool.tar.gz
```

Building the rrdtool RPM is shown in Example 4-19 on page 196.

At the time of writing, the tcl RPM package in RHAS 3 lacks the header files to compile Tcl applications. We therefore have to edit the spec file and build without tcl support. This is not a problem for the operation of ganglia later on. There are four places to fix in the rrdtool.spec file:

- Comment out `BuildRequires: tcl` around line 16
- Change the configure command near line 44, replacing `--with-tcllib=%{_libdir}` with `--without-tcllib`
- Add the following 4 lines around line 69 after the `# remove .in/.am files` comment line
  - `rm -rf ${RPM_BUILD_ROOT}/usr/contrib`
  - `rm -rf ${RPM_BUILD_ROOT}/usr/doc`
  - `rm -rf ${RPM_BUILD_ROOT}/usr/html`
  - `rm -rf ${RPM_BUILD_ROOT}/usr/examples`
- Add a line with `%{_libdir}/librrd.la*` after line 84 reading `%{_libdir}/librrd.so*`

We are now ready to build and install the RRDtool packages as shown in Example 4-19.

*Example 4-19 Build the RRDtool RPM*

```
[leecy@p630]> cp rrdtool.tar.gz ~/build/SOURCES/rrdtool-1.0.45.tar.gz
[leecy@p630]> cd ~/build/SPECS
[leecy@p630]> tar zxf ../SOURCES/rrdtool-1.0.45.tar.gz
[leecy@p630]> mv rrdtool-1.0.45/rrdtool.spec ./
[leecy@p630]> rm -rf rrdtool-1.0.45/
[leecy@p630]> touch ../SOURCES/MRTG-HOWTO # fix a missing file ...
[leecy@p630]> vi rrdtool.spec # hack the spec file to build without tcl as
desribed above
[leecy@p630]> rpmbuild -ba --target ppc rrdtool.spec
[leecy@p630]> su - # switch to root to install the RPMs
[root@p630]> rpm -i --nodeps ~leecy/build/RPMS/ppc/rrdtool-*.rpm
[root@p630]> ^D
[leecy@p630]
```

Now we move back to the RPMization of ganglia. Fortunately, once the RRDtool is installed, no changes are required.

*Example 4-20 Building ganglia for RHAS 3*

```
[leecy@p630]> rpmbuild -ba --target ppc ganglia.spec
[leecy@p630]> su - # switch to root to install the RPMs
[root@p630]> rpm -i --nodeps ~leecy/build/RPMS/ppc/ganglia-monitor-g*.rpm
Starting GANGLIA gmond: [ OK ]
Starting GANGLIA gmetad: [ OK ]
[root@p630]> ^D
[leecy@p630]
```

If you run RHAS 3, you can skip the next section.

### *Building ganglia for SLES 8*

You must install the rrdtool RPM from the SLES 8 distribution. Some changes need to be applied to the ganglia spec file and the two startup scripts for gmond and gmetad to make them fit SLES 8. They are currently very Red Hat-specific.

A valid ganglia.spec file is listed in Example 4-21. It replaces the Red Hat init scripts directory with the LSB-compliant one and allows for a patch (Patch0) to modify the startup scripts so that they fit SLES 8. Another solution would be to use the skeleton given in SLES 8 under /etc/init.d skeleton to build the two startup scripts for gmond and gmetad.

*Example 4-21 ganglia.spec file for use with SLES 8*

```
#
# $Id: ganglia.spec.in,v 1.12 2003/10/29 20:40:03 sacerdoti Exp $
#
# ganglia.spec.  Generated from ganglia.spec.in by configure.
#
Summary: Ganglia Cluster Toolkit http://ganglia.sourceforge.net/
Name: ganglia-monitor-core
Version: 2.5.5
Release: 1
Copyright: BSD
Vendor: Ganglia Development Team <ganglia-developers@lists.sourceforge.net>
Group: System Environment/Base
Source: %{name}-%{version}.tar.gz
Patch0: %{name}-%{version}-sles.patch
Buildroot: /tmp/%{name}-%{version}-buildroot
Prefix: /usr

%description
Ganglia is a scalable, real-time monitoring and execution environment
with all execution requests and statistics expressed in an open
well-defined XML format.


%package gmetad
Summary: Ganglia Meta daemon http://ganglia.sourceforge.net/
Group: System Environment/Base
Obsoletes: ganglia-monitor-core

%description gmetad
Ganglia is a scalable, real-time monitoring and execution environment
with all execution requests and statistics expressed in an open
well-defined XML format.
```

```
This gmetad daemon can aggregate monitoring data from several clusters
to form a monitoring grid. It also keeps metric history using the RRD tool.


%package gmond
Summary: Ganglia Monitor daemon http://ganglia.sourceforge.net/
Group: System Environment/Base
Obsoletes: ganglia-monitor-core

%description gmond
Ganglia is a scalable, real-time monitoring and execution environment
with all execution requests and statistics expressed in an open
well-defined XML format.

This gmond daemon provides the ganglia service within a single cluster or
Multicast domain.


%package lib
Summary: Ganglia Toolkit Library http://ganglia.sourceforge.net/
Group: System Environment/Base

%description lib
The Ganglia Monitoring Core library provides a set of functions that
programmers
can use to build scalable cluster or grid applications

##
## PREP
##

%prep
%setup
%patch0 -p 1

##
## BUILD
##
%build
./configure --with-gmetad --prefix=/usr
make

##
## PRE
##
%pre

##
## POST GMETA
```

```
##
%post gmetad
/sbin/chkconfig --add gmetad

if [ "$1" == "1" ]; then
   # Installing new package - start gmond
   /etc/init.d/gmetad start
elif [ "$1" > "1" ]; then
   # Upgrading ganglia package - restart gmond
   /etc/init.d/gmetad restart
fi


##
## POST GMON
##
%post gmond
/sbin/chkconfig --add gmond

if [ "$1" == "1" ]; then
   # Installing new package - start gmond
   /etc/init.d/gmond start
elif [ "$1" > "1" ]; then
   # Upgrading ganglia package - restart gmond
   /etc/init.d/gmond restart
fi


##
## PREUN GMETA
##
%preun gmetad
if [ "$1" = 0 ]
then
   /etc/init.d/gmetad stop
   /sbin/chkconfig --del gmetad
fi

##
## PREUN GMON
##
%preun gmond
if [ "$1" = 0 ]
then
   /etc/init.d/gmond stop
   /sbin/chkconfig --del gmond
fi

##
```

```
## INSTALL
##
%install
%__mkdir -p $RPM_BUILD_ROOT/etc/init.d
%__mkdir -p $RPM_BUILD_ROOT/usr/include/ganglia
%__mkdir -p $RPM_BUILD_ROOT/var/lib/ganglia/rrds
%__cp -f %{_builddir}/%{name}-%{version}/gmond/gmond.init
$RPM_BUILD_ROOT/etc/init.d/gmond
%__cp -f %{_builddir}/%{name}-%{version}/gmetad/gmetad.init
$RPM_BUILD_ROOT/etc/init.d/gmetad
%__cp -f %{_builddir}/%{name}-%{version}/gmond/gmond.conf
$RPM_BUILD_ROOT/etc/gmond.conf
%__cp -f %{_builddir}/%{name}-%{version}/gmetad/gmetad.conf
$RPM_BUILD_ROOT/etc/gmetad.conf
%__make install prefix=$RPM_BUILD_ROOT/usr

##
## FILES GMETA
##
%files gmetad
%defattr(-,root,root)
%attr(0755,nobody,nobody)/var/lib/ganglia/rrds
/usr/sbin/gmetad
/etc/init.d/gmetad
%config(noreplace) /etc/gmetad.conf

##
## FILES GMON
##
%files gmond
%defattr(-,root,root)
%attr(0500,root,root)/usr/bin/gmetric
%attr(0555,root,root)/usr/bin/gstat
/usr/sbin/gmond
/etc/init.d/gmond
%config(noreplace) /etc/gmond.conf

%files lib
/usr/include/ganglia.h
/usr/include/ganglia
/usr/lib/libganglia*

##
## CLEAN
##
%clean
%__rm -rf $RPM_BUILD_ROOT

##
```

```
## CHANGELOG
##
%changelog
* Mon Oct 14 2002 Federico Sacerdoti <fds@sdsc.edu>
- Split package into -gmetad and -gmond subpackages for clarity,
  and separation of purpose/functionality.
* Thu Sep 19 2002 Federico Sacerdoti <fds@sdsc.edu>
- Added config files, made /var/lib/ganglia for RRD storage.
* Mon Mar 11 2002 Matt Massie <massie@cs.berkeley.edu>
- Added support for libganglia, added Prefix: for RPM relocation
* Wed Feb 27 2002 Matt Massie <massie@cs.berkeley.edu>
- Merge gmetric and gmond together into one RPM.  Fix some small bugs.
* Fri Nov  2 2001 Matt Massie <massie@cs.berkeley.edu>
- initial release
```

A patch named ganglia-monitor-core-2.5.5-sles.patch must reside under the build/SOURCES directory. Its contents are listed in Example 4-22.

*Example 4-22   Patch for the startup scripts*

```
*** ganglia-monitor-core-2.5.5/gmond/gmond.init 2003-04-09 20:37:38.000000000 -0400
--- ganglia-monitor-core-2.5.5_new/gmond/gmond.init     2003-10-15 18:28:54.000000000
-0400
***************
*** 6,12 ****
  #
  GMOND=/usr/sbin/gmond

! . /etc/rc.d/init.d/functions

  RETVAL=0

--- 6,12 ----
  #
  GMOND=/usr/sbin/gmond

! . /etc/rc.status

  RETVAL=0

***************
*** 15,22 ****
        echo -n "Starting GANGLIA gmond: "
        [ -f $GMOND ] || exit 1

!       daemon $GMOND
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/gmond
        ;;
--- 15,23 ----
        echo -n "Starting GANGLIA gmond: "
        [ -f $GMOND ] || exit 1

!       startproc $GMOND
        RETVAL=$?
+       rc_status -v
echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/gmond
        ;;
***************
*** 25,30 ****
--- 26,32 ----
        echo -n "Shutting down GANGLIA gmond: "
        killproc gmond
        RETVAL=$?
+       rc_status -v
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/gmond
        ;;
***************
*** 35,42 ****
        RETVAL=$?
        ;;
    status)
!       status gmond
!       RETVAL=$?
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
--- 37,47 ----
        RETVAL=$?
        ;;
    status)
!         echo -n "Checking for gmond: "
!         checkproc $GMOND; RETVAL=$?
!         if test $RETVAL = 0; then echo "OK"
!         else echo "No process"
!         fi
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
*** ganglia-monitor-core-2.5.5/gmetad/gmetad.init       2003-04-09 20:37:37.000000000
-0400
--- ganglia-monitor-core-2.5.5_new/gmetad/gmetad.init   2003-10-15 18:27:26.000000000
-0400
***************
*** 1,12 ****
  #!/bin/sh
! # $Id: gmetad.init,v 1.2 2002/10/18 21:57:58 sacerdoti Exp $
  #
! # chkconfig: 2345 20 80
  # description: gmetad startup script
  #
  GMETAD=/usr/sbin/gmetad

! . /etc/rc.d/init.d/functions

  RETVAL=0

--- 1,12 ----
  #!/bin/sh
! # $Id: gmetad.init,v 1.2 2003/03/07 20:38:30 sacerdoti Exp $
  #
! # chkconfig: 2345 70 40
  # description: gmetad startup script
  #
  GMETAD=/usr/sbin/gmetad

! . /etc/rc.status

  RETVAL=0

***************
*** 15,46 ****
        echo -n "Starting GANGLIA gmetad: "
        [ -f $GMETAD ] || exit 1

!       daemon $GMETAD
        RETVAL=$?
        echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/gmetad
!       ;;

    stop)
        echo -n "Shutting down GANGLIA gmetad: "
        killproc gmetad
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/gmetad
!       ;;

    restart|reload)
!       $0 stop
!       $0 start
!       RETVAL=$?
!       ;;
    status)
!       status gmetad
!       RETVAL=$?
!       ;;
    *)
!       echo "Usage: $0 {start|stop|restart|status}"
!       exit 1
  esac

  exit $RETVAL
```

```
--- 15,51 ----
        echo -n "Starting GANGLIA gmetad: "
        [ -f $GMETAD ] || exit 1

!       startproc $GMETAD
        RETVAL=$?
+       rc_status -v
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/gmetad
!       ;;

    stop)
        echo -n "Shutting down GANGLIA gmetad: "
        killproc gmetad
        RETVAL=$?
+       rc_status -v
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/gmetad
!       ;;

    restart|reload)
!       $0 stop
!       $0 start
!       RETVAL=$?
!       ;;
    status)
!         echo -n "Checking for gmetad: "
!         checkproc $GMETAD; RETVAL=$?
!         if test $RETVAL = 0; then echo "OK"
!         else echo "No process"
!         fi
!       ;;
    *)
!       echo "Usage: $0 {start|stop|restart|status}"
!       exit 1
  esac

  exit $RETVAL
```

With these changes, building and installing proceeds as shown in Example 4-23.

*Example 4-23 Building rpm using the spec file*

```
[leecy@p630]> rpm -ba --target ppc ganglia.spec
[leecy@p630]> su - # switch to root to install the RPMs
[root@p630]> rpm -i --nodeps ~leecy/build/RPMS/ppc/ganglia-monitor-g*.rpm
[root@p630]> ^D
[leecy@p630]
```

Under SLES 8, the two services are not started automatically.

### Installing the Web front-end

You need to set up a PHP-enabled Web server somewhere on your network. On this system, you have to install the front-end RPM downloaded in “Downloading ganglia” on page 194.

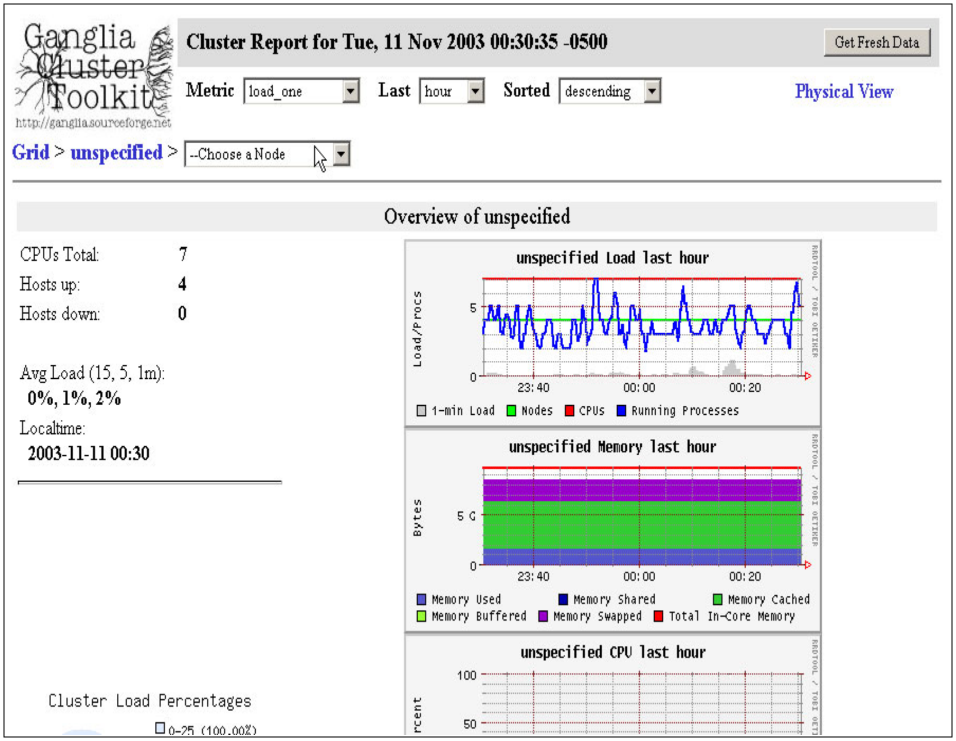
This file installs under /var/www, which will soon become the “old” location for the server’s data. SLES 8 implements today the proposed LSB standard location, /srv/var/www. After installing under SLES8, you will have to move the /var/www/html/ganglia directory under /srv.

The next step is to create the space in /var to hold the monitoring data. Make sure you create the /var/lib/ganglia/rrds directory.

On this collector system, you can install the gmond RPM and you must install the gmetad RPM that will receive the monitoring data from all the nodes.

### Starting ganglia

Start the Web browser and point to the ganglia directory of your Web site. The ganglia Web page on your Web site might look similar to Figure 4-12 on page 206.



Cluster Report for Tue, 11 Nov 2003 00:30:35 -0500

Metric load_one Last hour Sorted descending

Grid > unspecified > --Choose a Node

Overview of unspecified

| | |
|---|---|
| CPUs Total: | 7 |
| Hosts up: | 4 |
| Hosts down: | 0 |

Avg Load (15, 5, 1m):
0%, 1%, 2%

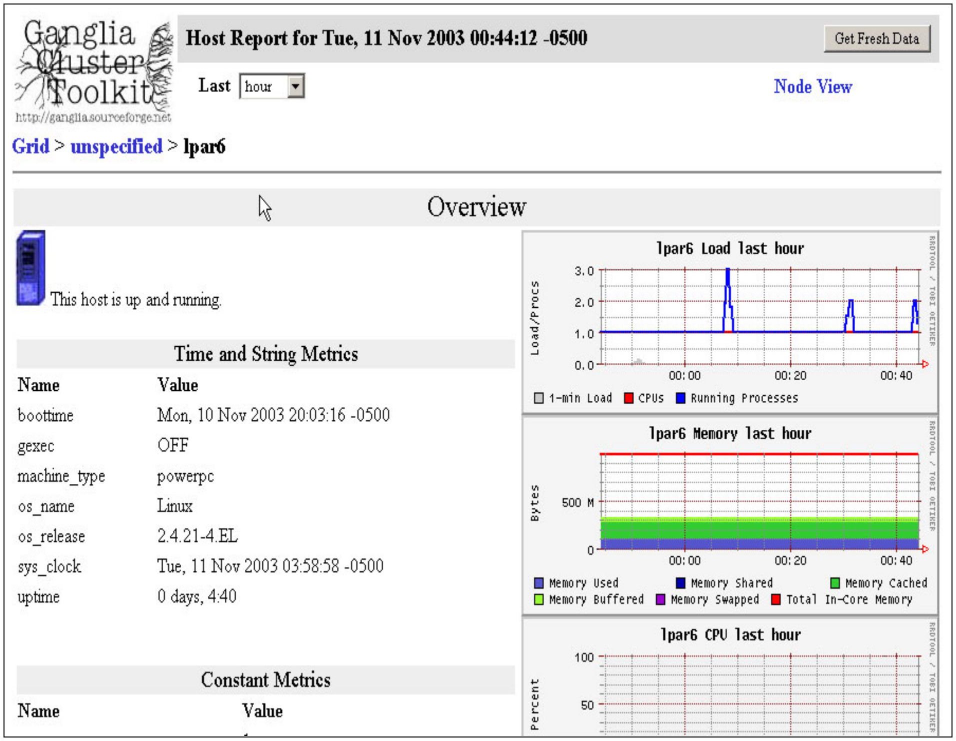
Localtime:
2003-11-11 00:30

*Figure 4-12 Ganglia summary page*

## Using ganglia

The summary screen shown in Figure 4-12 condenses the cluster information on one page. You can zoom in on one specific node by following the **Grid** -> **unspecified** -> **Choose a node path**.

In our example, shown in Figure 4-13 on page 207, lpar6 is a RHAS 3 node.



Host Report for Tue, 11 Nov 2003 00:44:12 -0500

Grid > unspecified > lpar6

Overview

This host is up and running.

Time and String Metrics

| Name | Value |
|---|---|
| boottime | Mon, 10 Nov 2003 20:03:16 -0500 |
| gexec | OFF |
| machine_type | powerpc |
| os_name | Linux |
| os_release | 2.4.21-4.EL |
| sys_clock | Tue, 11 Nov 2003 03:58:58 -0500 |
| uptime | 0 days, 4:40 |

Constant Metrics

| Name | Value |
|---|---|

*Figure 4-13 Zooming into a node*

Check the documentation for more configuration options. The two files that you may wish to change are the `/etc/gmond.conf` on all the nodes and the `/etc/gmetad.conf` on the collector node.

## 4.6.2 Basic tuning tips for Linux

Linux sports a dynamic kernel that allows you to modify and change many of the parameters without rebooting the system. Most of these parameters are located n the /proc file systems. There are two ways to change it, by using **cat** and **view**, and by using the command **sysctl**. We describe these methods in more detail here.

You can use **cat** to view, and use **echo** to change the variable.

To disable IP forwarding:

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

The command **sysctl** provides a simple interface to all the tunable parameters in the /proc file systems.

```
# sysctl -w net.ipv4.ip_forward="1"
```

Because most of the default kernel parameters for system performance are geared toward workstation workload rather than file server or large computation workload, there are tuning parameters that you can use to tune Linux for better performance. Following are some of the sample tuning parameters that can be applied into Linux. For a complete listing, use **sysctl** or the YaST2 utility.

### File system tuning

In Linux, the bdflush file in the /proc directory governs when the bdflush should be activated to clear cache dirty pages.

```
# echo "100 5000 640 2560 150 30000 5000 1884 2" > /proc/sys/vm/bdflush
```

For heavy I/O in a file server and Web server environment, you can also disable the atime. atime basically stores the information of when is the last time the file has been accessed. You can update your /etc/fstab to reflect this.

```
   # cat /etc/fstab
/dev/sda3        /       reiserfs        defaults 1 1
/dev/system/home         /home   reiserfs        defaults 1 2
/dev/system/usr /usr    reiserfs        defaults 1 2
/dev/system/var /var    reiserfs        defaults 1 2
/dev/system/web /web    reiserfs        noatime,defaults 1 2
/dev/sda2       swap    swap    pri=42 0 0
devpts          /dev/pts devpts  mode=0620,gid=5 0 0
proc            /proc   proc    defaults 0 0
/dev/cdrom      /media/cdrom    auto    ro,noauto,user,exec 0 0
```

Rather than changing the whole file system to reflect this change, use the command **chattr** to tag or mark individual files that do not need to store the access time.

```
# chattr -R +A /usr/local/apache/logs/httpd.logs
```

### Network tuning

To reduce the amount of work done at the TCP stack to check on every packet, you can basically disable by echo "0" to the file:

```
# echo "0" > /proc/sys/net/ipv4/tcp_sack
# echo "0" > /proc/sys/net/ipv4/tcp_timestamps
```

Or you can use the command **"sysctl -w <kernel_parameter="choice">"**.

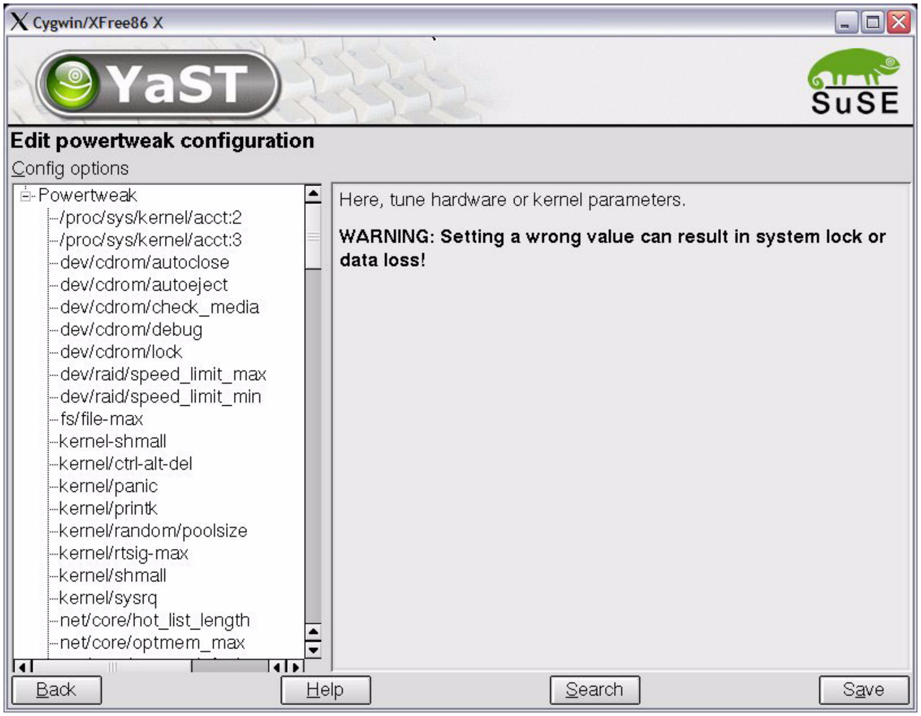
Often we see clients that do not properly close the TCP connections and so the server keeps the connections open, and may wind up with a large number of open connections. The Linux TCP stack will probe the TCP connections after a given amount of time of inactivity (by default, it is two hours). You can change this wait time as follows:

```
# echo “1800” > /proc/sys/net/ipv4/tcp_keepalive_time
```

If needed, you can also change the tcp_keepalive_intvl and tcp_keepalive_probes to determine the length of time to wait before the next probe occurs.

### Powertweak

The Powertweak tool from SuSE, shown in Figure 4-14, allows you to tweak many of the system parameters, including those mentioned. After you click the option you wish to tweak, the right-hand panel will explain what the option is for. Most tweaking takes effect immediately.



Here, tune hardware or kernel parameters.

**WARNING: Setting a wrong value can result in system lock or data loss!**

*Figure 4-14 YaST2 powertweak*

**Tip:** Any tuning done using the command **sysctl** is only good for the session, so if you want the changes to be permanent, create a file called /etc/sysctl.conf and put in the tuning parameters. The file will be read each time Linux boots up.

# 5

# Cluster Systems Management (CSM)

In this chapter, we explain the concepts of Cluster Systems Management (CSM) used for setting up and managing a cluster of nodes running the Linux operating system on IBM pSeries hardware. More specifically, we discuss how to set up the CSM management environment to manage a cluster of Linux nodes.

The following topics are discussed:

- 5.1, “CSM concepts and architecture” on page 212
- 5.2, “CSM planning, installation and configuration” on page 222
- 5.3, “CSM administration” on page 251
- 5.5, “CSM interoperability” on page 272
- 5.6, “Future of CSM” on page 277

This chapter is based on the functionality of the Cluster System Manager software running with SuSE Linux Enterprise Server-8 (SLES 8) on IBM pSeries hardware. At the time of writing, SLES 8 is the only supported and recommended version of operating system for pSeries Linux.

## 5.1 CSM concepts and architecture

To understand what CSM is, it is useful to understand the details of what clustering is, what types of clusters there are, and what are the features of each cluster. We cover the fundamentals in this section, and delve into details in later sections of this chapter.

We specifically discuss:

- Clustering basics
- Type of clusters
- CSM components such as Reliable Scalable Clustering Technology (RSCT), Resource Monitoring and Control (RMC), Resource Managers (RM), CSM hardware control, Cluster File Manager (CFM), CSM software maintenance, CSM diagnostic probes, CSM security and distributed shell.

### 5.1.1 Clustering basics

In simple terms, *clustering* is defined as two or more computers (generally referred to as *nodes*) working together and joined by a common network, sharing resources and services.

It differs from a workgroup or local area network (LAN) in that the nodes in the cluster use the resources (such as computing power and network adapters) of the other nodes. Moreover, nodes or member servers in the cluster communicate with each other or with one centralized server to maintain the state of the cluster. To keep the centralized server and/or member servers informed of up-to-date changes, the cluster design can get quite complex with respect to network connectivity and arrangements of shared resources, depending on cluster type and business requirements.

### 5.1.2 Types of clusters

Based on functionality, clusters are grouped into four major types:

- High availability clusters
- High performance clusters
- Virtual load balancing clusters
- Management clusters

These clustering groups are discussed in the following sections.

## High availability (HA) clusters

As the name suggests, high availability clusters provide high availability of resources in a cluster. These resources can be processors, shared disk, network adapters and nodes.

The main goal of a high availability cluster is to eliminate single points of failure (SPOFs) in the design. High availability is often confused with fault tolerant design. A fault tolerant design provides 100% availability by using spare hardware, whereas high availability provides 90+% availability by eliminating SPOFs and better usage of shared resources. RAID levels 1, 5 are examples of a fault tolerant design.

HA clusters are commonly designed based on the functionality of applications and resources that are part of the cluster. Figure 5-1 shows a simple high availability cluster.

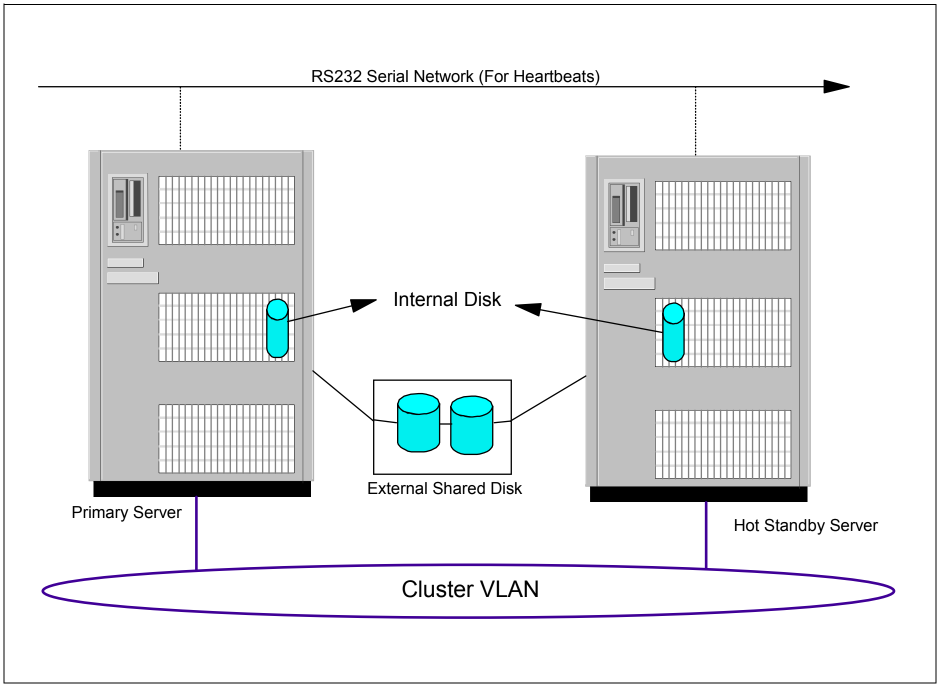


*Figure 5-1 High availability cluster*

A simple HA cluster consists of two servers sharing a common disk and network. One server is the primary server running active applications, and the second server is the hot standby server, which takes over resources (shared disk, network IP and applications) in case of a problem with the primary server. This type of configuration is typically referred as an active-passive type of high availability.

There is another type of configuration in which both servers in the cluster function as primaries, running active applications at the same time. In case of a failure or problem, the surviving or active node will take over the resources of the failed server. This type of configuration is referred as an active-active type of high availability.

Depending on the failover mechanism, a high availability cluster can be designed to meet complex needs of a business scenario. Failover behavior is customized in several ways to keep the resources running on takeover node; some examples of how this can be customized include fail back immediately when the failed node becomes active, or fail back later at a scheduled time so that no outage occurs.

For more complex applications, such as parallel databases or applications, concurrent access configurations are designed with a distributed lock type of control and availability.

### High Performance Computing (HPC) clusters

HPC clusters are used in high computing environments, with two or more nodes accessing the processors in parallel combining the power of multiple nodes at the same time.

HPC clusters are normally used in scientific computing and benchmarking centers that require a lot of processing power to analyze complex mathematical and scientific solutions. More information on HPC clusters can be found in Chapter 7, “High Performance Computing case studies” on page 303.

Figure 5-2 on page 215 shows a typical high performance cluster.

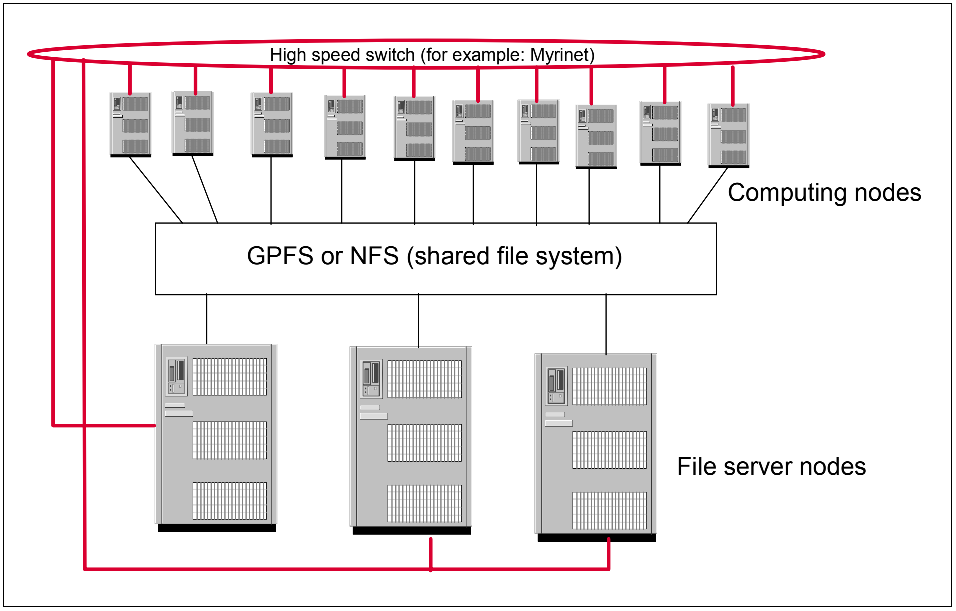

High speed switch (for example: Myrinet)
Computing nodes
GPFS or NFS (shared file system)
File server nodes

*Figure 5-2 High performance cluster*

### Virtual load balancing clusters

These clusters are frequently used in high volume Web server and e-business environments where a front-end load balancing server routes TCP traffic to back-end Web servers on common TCP ports such as port 80 (http), port 443 (ssl), 119 (nntp) and 25 (smtp). Figure 5-3 on page 216 shows a sample virtual load balancing cluster.

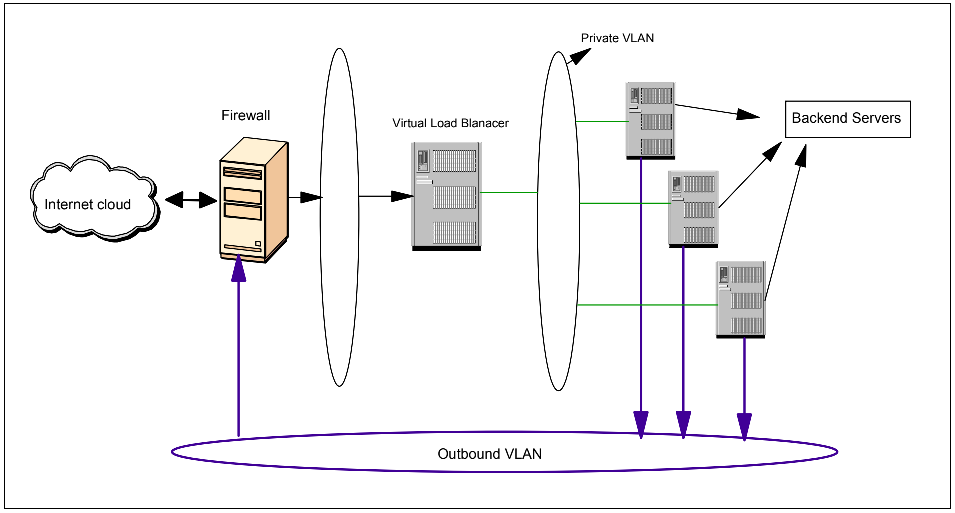


*Figure 5-3 Virtual load balancing cluster*

The inbound traffic is received on the load balancer and is dispatched immediately to an available back-end server. The availability of back-end servers, and the server to be routed next, is determined using an algorithm that assigns a particular weight based on number of TCP connections and custom polling agents.

The main advantage with these clusters is that the outside world sees only one virtual IP which in turn is mapped to an intranet network of computers forming a virtual cluster. Example 5-1 illustrates virtual load balancing.

*Example 5-1 Example of virtual IP load balancing*

```
VirtualIP = 192.168.1.10 (URL = www.testsite.com)
IP of load balancer ent0 interface= 192.168.1.1
IP of load balancer ent1 interface = 10.10.10.4
IP of web server1 = 10.10.10.1
IP of web server2 = 10.10.10.2
IP of web server3 = 10.10.10.3

When a client connects to www.testsite.com, it resolves to IP 192.168.1.10 and
client browser loads the page directly from one of available web servers
without knowing the real IP address of that particular web server.
```

Depending on the requirements (such as virtual IPs, ports, and applications), load balancing can become quite complex with rules, cookie-based affinities, sticky connections, and so on. The IBM WebSphere® Edge server (IBM eNetwork dispatcher) is an example of a virtual load balancing server; refer to

*IBM WebSphere Edge Server User Guide*, GC09-4567, for more information on this subject.

## Management clusters

A management cluster consists of a group of computers networked together and functioning either independently or together by sharing resources, but all being managed and controlled from one centralized management server. IBM Cluster Systems Management (CSM) is an example of a management cluster.

CSM uses the management server as a single point of control in the management domain managing a set of servers referred as managed nodes. Using the management server, a system administrator centralizes the cluster environment. Some of the shared tasks that can be performed from a management server are:

- Monitor all nodes in the cluster
- Install and update software
- Distribute files across the cluster, and centralize user ID management shared functions
- Remotely control node hardware such as reboot, power on/off
- Manage node groups
- Diagnose problems on nodes
- Run commands on multiple nodes at the same time with a distributed shell

Figure 5-4 on page 218 shows a CSM cluster environment.

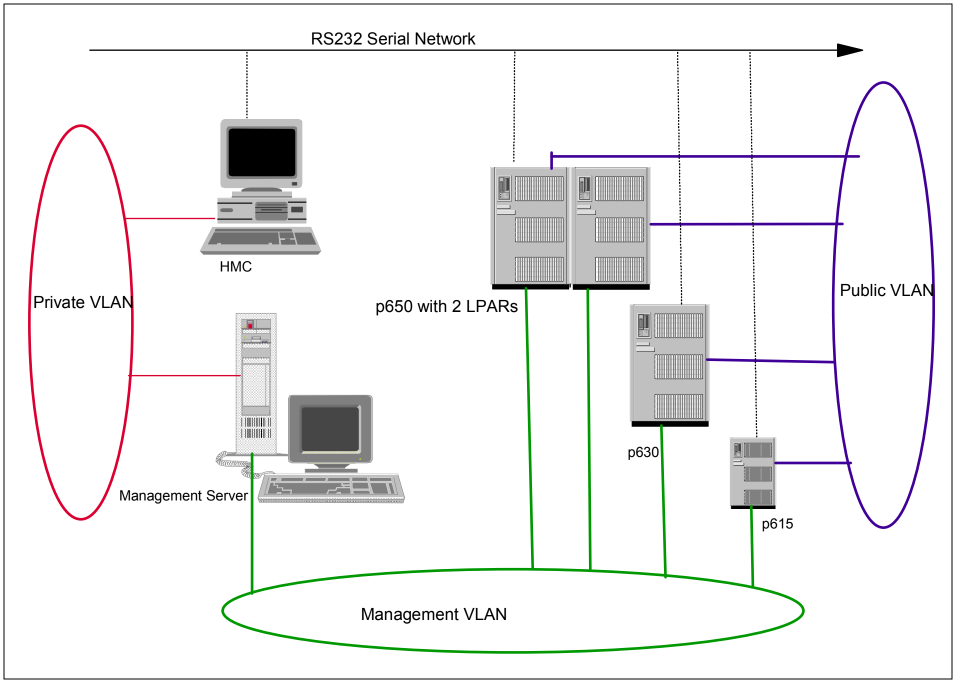


*Figure 5-4 CSM cluster*

A management server consists of a single server with management tools communicating to a group of nodes across a shared LAN. The CSM architecture includes a set of components and functions that work together to form a management cluster.

The following sections briefly define and describe the major components of CSM and how they interact with each other.

## 5.1.3 RSCT, RMC, and RM

Reliable Scalable Clustering Technology (RSCT) is the backbone of the Cluster Systems Management domain. Other CSM components such as event monitoring and CSM security are based on RSCT Infrastructure. RSCT and its related components are explained in this section.

### Reliable Scalable Cluster Technology (RSCT)

RSCT provides a monitoring environment for CSM. RSCT was primarily developed for high availability applications such as GPFS and PSSP for AIX, and it has ported to CSM.

A detailed description of RSCT is available in *RSCT for Linux: Guide and Reference,* SA22-7892.

### Resource Monitoring and Control (RMC)

RMC is part of RSCT. RMC is a framework for providing availability monitoring to CSM. The RMC subsystem monitors system resources such as file systems, CPU, and disks, and performs an action based on a condition and response behavior.

For further information on RMC, refer to *RSCT for Linux: Guide and Reference,* SA22-7892.

### Resource Manager (RM)

RM is a daemon that maps resources and class attributes to commands for several resources. Some of the standard resource managers available are IBM.AuditRm for system-wide audit logging, IBM.HWCtrlRM for hardware control for managed nodes, IBM.ERRM for providing actions in response to conditions, and IBM.DMSRM for managing a set of nodes and node groups.

RSCT, RMC and RM are discussed with examples and in more detail in 5.3, “CSM administration” on page 251.

## 5.1.4 CSM hardware control

As the name indicates, CSM hardware control allows control of the Hardware Management Console (HMC)-connected pSeries hardware from a single point. This feature lets system administrators remotely power on/off and open a serial console from the management server to the pSeries LPARs. An HMC is a must for using this feature as the LPARs are connected over an RS232/RS422 serial connection to the HMC.

For further information on hardware control, including diagnostic messages, refer to *CSM for Linux: Hardware Control Guide,* SA22-7856*.*

## 5.1.5 Cluster File Manager (CFM)

CSM cluster file manager allows centralized repository and file management across the cluster. This is installed along with CSM server packages. On the management server, the default directory structure under /cfmroot is created at install time with sample config files.

Using CFM, file management is simplified for all common files across a management cluster. CFM uses a push mechanism to push defined files to managed nodes thorough a cron job once every 24 hours. Customizations can be made to control which file needs to be pushed, and to what nodes.

Configuration of CFM and details are discussed in 5.3.4, “Configuration File Manager (CFM)” on page 256.

## 5.1.6 CSM software maintenance

The software maintenance component provides an interface to manage remote software install and updates on to the pSeries Linux nodes. RPM packages can be queried, installed, updated and removed with this tool.

SMS uses NFS to remotely mount RPM directories, dsh to distributed commands, and Autoupdate to automatically update installed software. Auto update software is not packaged with CSM and has to be installed separately. Detailed prerequisites and configuration are discussed further in “Software maintenance” on page 259.

## 5.1.7 CSM diagnostic probes

CSM diagnostic probes consists of a probe manager and a set of probes. It can be used to diagnose a system problem, which is helpful to identify root cause of a problem.

Detailed descriptions of the probes, and usage examples, can be found in “Diagnostic probes” on page 265.

## 5.1.8 CSM security

CSM security uses underlying RMC to provide a secure environment. CSM security provides authentication, secure shell, and authorization for all managed nodes in the cluster.

By default, shell security is provided using openSSH. Secure shell is used for opening a remote shell from the management server to all managed nodes using SSH-v2 protocol. The SSHD daemon is installed and started at install time on managed nodes.

Authentication is a host-based authentication (HBA) model which uses public-private key pairs. Public keys are exchanged from all managed nodes with the management server. CSM performs the key exchange at node install time; no manual activity needs to be performed by the system administrator.

**Note:** Public keys are exchanged only between a managed node and a management server. No keys are exchanged between the managed nodes.

A public and private key pair is generated using **ssh-keygen** at node install time. By default, CSM security uses “dsa”-based security for generating keys. These keys *cannot* be used for any other remote command applications, and they are stored at the following locations:

- /var/ct/cfg/ct_has.qkf
- /var/ct/cfg/ct_has.pkf
- /var/ct/cfg/ct_has.thl

CSM authorization is based on an access control list (ACL) file. RMC uses this control list to verify and control any command execution by a user. The ACL file is stored at /var/ct/cfg/ctrmc.acl. This file can be modified as needed to grant access control. By default, the root account has read and write access to all resource classes, and all other uses are allowed read access only.

### 5.1.9 Distributed shell (dsh)

CSM packages a distributed shell in the csm.dsh package and it is installed while running installms. dsh is used for most cluster distributed management functions such as simultaneous node updates, commands, queries and probes. By default, **dsh** uses ssh for Linux nodes; it can be modified to use any other remote shells such as rsh using **csmconfig -r**. Example 5-2 shows the **csmconfig** command output.

*Example 5-2 csmconfig output*

```
#csmconfig
   AddUnrecognizedNodes = 0 (no)
    ClusterSNum =
    ClusterTM = 9078-160
    ExpDate = Mon Dec 15 18:59:59 2003
    HeartbeatFrequency = 12
    HeartbeatSensitivity = 8
    MaxNumNodesInDomain = -1 (unlimited)
    RegSyncDelay = 1
    RemoteShell = /usr/bin/ssh
    SetupRemoteShell = 1 (yes)
```

The **dsh** command runs concurrently on each node specified with the **-w** flag, or on all nodes specified in the WCOLL environment variable. Nodegroups can also be specified with the **-N** flag.

A sample **dsh** output is shown in Example 5-3 for a single node.

*Example 5-3 dsh command output for a single node*

```
#dsh -w lpar date
   lpar1: Wed Oct 22 17:06:51 EDT 2003
```

Example 5-4 shows a group of nodes part of a node group called group1.

*Example 5-4 dsh command output for a node group*

```
#dsh -N group1 date
   lpar3: Wed Oct 22 17:08:28 EDT 2003
   lpar1: Wed Oct 22 17:08:27 EDT 2003
```

The **dshbak** command is also part of the distributed shell and is used to format dsh output. Example 5-5 shows how to use **dshbak** for formatting dsh output.

*Example 5-5 dsh and dshbak output*

```
#dsh -N group1 date | dshbak
   HOST: lpar1
   -----------
   Wed Oct 22 17:09:03 EDT 2003

   HOST: lpar3
   -----------
   Wed Oct 22 17:09:03 EDT 2003
```

## 5.2 CSM planning, installation and configuration

In this section, we discuss planning a CSM cluster, installing and configuring CSM on management server, and configuring CSM to add and install managed nodes.

More specifically, we discuss how and what to plan for a CSM cluster, describe supporting hardware, software, and network, give tips on installing and configuring the management server, and how to add and install managed nodes in the cluster.

At a high level, planning for setting up a CSM cluster involves:

- Planning hardware, software, network and security
- Installing and configuring the management server
- Adding/defining and installing nodes

Each of these topics are discussed in detail in the next few sections.

## 5.2.1 Planning hardware

The following hardware requirements must be met to support CSM on the management server, nodes in the cluster and hardware control.

### Management server requirements

- Must be a pSeries machine p615, p630 or p650 running Linux. p655 (Cluster 1600) is not supported to be a management server.
- A logical partition (LPAR) can be configured as a management server, but we do not recommend using an LPAR as a management server. For further information on why an LPAR is not recommended, refer to *CSM Guide for the PSSP Administrator*, SG24-6953.
- At a minimum, 128 MB of memory and 120 MB of disk space is required to install CSM.
- Each copy of the operating system image (such as SuSE Linux 8) requires at least 2 GB of disk space.
- Each version of the operating system service packs or RPM updates and prerequisites requires an additional 2 GB of disk space.

### Managed nodes requirements

- Any pSeries managed node, including p655, can be part of a CSM cluster. pSeries logical partitions (LPARs) are supported to be managed nodes.

**Restriction:** Nodes must be pSeries hardware only and cannot be mixed with any other xSeries or iSeries nodes in the pSeries Linux cluster at this time.

- Nodes can either be pre-installed with Linux or installed with the CSM management server. In the former case, the management server will install CSM packages and make it a managed node.
- A minimum of 128 MB of memory and 20 MB of disk space is required for CSM packages.
- A minimum of 2 GB of disk space is required for each operating system install.

**Important:** For p615, p630, p650 and p655, the required open firmware version is RG030407_GA4_ALL or greater.

### Hardware control requirements

As explained in the previous section, CSM hardware control provides remote hardware functions such as powering on/off, querying power status, and remote console from the management server to managed nodes. Following are some guidelines and requirements to utilize this feature:

- The IBM Hardware Management Console (HMC) for pSeries hardware such as p650 and p630 is a must for hardware control.
- The management server must communicate with a HMC on a management LAN.
- HMC is connected to pSeries hardware using async serial adapters.

> **Important:** To enable hardware control, HMC open firmware/driver version must be version 3.4 or greater and HMC build level 20030410.

More information about hardware control requirements can be found in *CSM for Linux: Hardware Control Guide*.

## 5.2.2 Planning software

CSM has requirements for both IBM and non-IBM-developed software. On a pSeries Linux cluster, including the management server and all managed nodes, SuSE Linux Enterprise Server (SLES) 8 (8.1) is mandatory. The management server must have Linux installed prior to installing CSM.

Other software requirements are summarized as follows for IBM and non-IBM vendors

### IBM software

CSM for pSeries Linux version 1.3.2 is shipped on CSM for pSeries Linux CD-ROMs. It can also be downloaded from the following site:

```
http://techsupport.services.ibm.com/server/cluster/fixes/csmfixhome.html
```

Following is a list of the IBM software packaged with the CSM CD-ROM:

- csm.core 1.3.2
- csm.client 1.3.2
- csm.server 1.3.2
- csm.diagnostics 1.3.2
- csm.dsh 1.3.2
- csm.license
- rsct.basic 2.3.1
- rsct.core 2.3.1

- rsct.core.utils 2.3.1
- src 1.2.0.2
- IBM Java2-JRE 1.3.1

### Non-IBM software

SuSE Linux Enterprise Server 8 (8.1).

The following non-IBM software prerequisites for CSM are shipped on the base SLES CD-ROM:

- dhcp-base-*
- dhcp-server-*
- expat-*
- expect-5.34-63*
- freetype2*
- gppshare-*
- nfs-utils-*
- pdksh-*
- perl-5*
- perl-XML-Parser-*
- perl-XML-RegExp*
- perl-libwww-perl*
- rdist-6.1.5-524*
- rsh-server-*
- rsync-*
- telnet-server*
- tcl-8*
- tk-8*
- xf86-*

**Important:** On the management server, kernel-ppc64-2.4.19-228.ppc.rpm or greater is required.

You can copy the kernel rpm from:

```
</csminstall/Linux/SLES/8.1/ppc64/updates/kernel/kernel***.rpm>
```

For updates, visit:

http://support.suse.de/psdb

### Other required non-IBM software

The following lists other non-IBM software required to install CSM:

- conserver-7.2*, shipped with the CSM for pLinux CD-ROM
- tftp-hpa-0.34-1.ppc.rpm, shipped with the CSM for pLinux CD-ROM

- OpenCIMOM, available from the following site:

  http://www.ibm.com/servers/aix/products/aixos/linux/download.html

- Autoupdate 4.3.4 or higher, available from the following site:

  http://freshmeat.net/projects/autoupdate

### 5.2.3 Planning network

CSM requires a management server connected to HMC, and managed nodes on network virtual local area networks (VLANs), either on the same subnet, or on a separate subnet. But to have a secure cluster, we recommend that you have a management server connected to the HMC and managed nodes on two different VLANs. That way, the management server can maintain a secure connection with HMC on a private VLAN.

We also recommend that you have all managed nodes use a separate public VLAN for internode communications, so that the Management VLAN is used exclusively for management server-to-managed node traffic only, for functions such as node updates, monitoring, and so on. Figure 5-5 shows a recommended secure CSM cluster network.

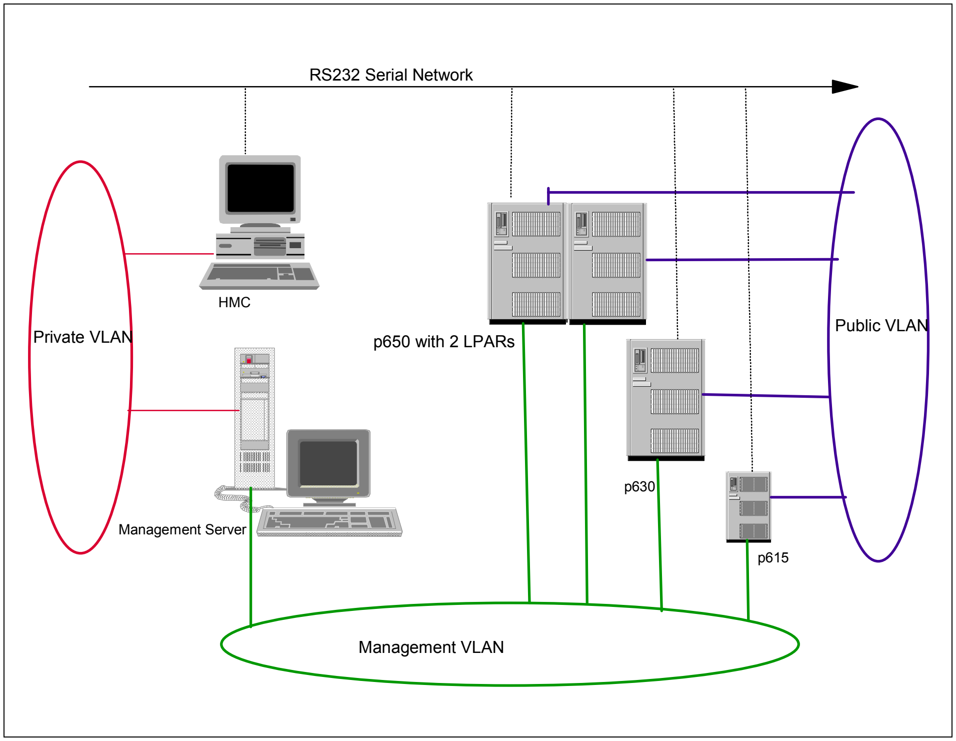


*Figure 5-5 Secure CSM cluster network*

As shown in the figure, all pSeries managed nodes are connected to HMC on an RS232 serial network.

## 5.2.4 Planning security

As discussed in “CSM security” on page 220, this includes shell security, architecture, and authentication. Shell security and authentication are performed by default using ssh and host-based authentication.

The system adiministrator can switch to rsh and enable rhosts-based security, but ssh is more secure than rsh. Network security should be planned to isolate networks and protect clusters. Authorizations can be planned to determine which user can have which access to resource classes.

## 5.2.5 Installing CSM on the management server

This section discusses installing CSM on the management server. Linux must be installed on the management server prior to performing CSM install. The CSM install involves the following major steps:

- Installing SuSE Linux and pre-requisites
- Installing csm.core fileset
- Creating /csminstall partition
- Running installms to install CSM and other prerequisites

### Installing SuSE Linux and Linux pre-requisites

Install the designated management server with SuSE Linux 8.1. The Enterprise server install option installs most of the CSM requisite packages. Update the server with the latest service pack updates from SuSE. This upgrades the kernel to latest version (which was v2.4-21-83, at the time of writing).

### Installing csm.core

Now that the management server is installed and upgraded to the latest kernel and packages, CSM install can be started. csm.core is the first package to install, as this contains the executable files required for all cluster nodes, and it installs the scripts required for further installation and setting up the environment.

This can be done either by using the CSM CD-ROM directly, or by installing it from packages copied to local disk. The CSM packages can be copied from CD-ROM if you ordered and received the media, or you can download the software from the Internet. The license file must be ordered from IBM separately and is shipped in a CD-ROM.

The CSM package can be downloaded from the following Internet site:

http://techsupport.service.ibm.com/servers/cluster/fixes/clusterfixhome.html

Copy and unpack the contents to a temporary directory(/tmp/csm):

```
# cd /tmp/csm; gzip -dc csm-linux-1.3.2.0.ppc64.tar.gz | tar -xvf -
```

The `ls` command on /tmp/csm is shown in Example 5-6.

*Example 5-6 ls -l command output in /tmp/csm*

```
total 49047
drwxr-xr-x    3 root     root          584 Oct 23 15:39 .
drwxr-xr-x    3 root     root          200 Oct 23 15:39 ..
-rw-r--r--    1 root     root         7360 Sep 23 10:19 README
-rwxr--r--    1 root     root     34140929 Oct 23 15:32 csm-linux-1.3.2.0.ppc64.tar.gz
-rw-r--r--    1 root     root       496702 Sep 23 10:19 csm.client-1.3.2.0-26.ppc.rpm
```

```
-rw-r--r--    1 root     root       352444 Sep 23 10:19 csm.core-1.3.2.0-26.ppc.rpm
-rw-r--r--    1 root     root        67982 Sep 23 10:19
csm.diagnostics-1.3.2.0-26.ppc.rpm
-rw-r--r--    1 root     root        67605 Sep 23 10:19 csm.dsh-1.3.2.0-26.ppc.rpm
-rw-r--r--    1 root     root      4261877 Sep 23 10:19 csm.server-1.3.2.0-26.ppc.rpm
drwxr-xr-x    2 root     root          272 Oct 23 15:39 reqs
-rw-r--r--    1 root     root      3786767 Sep 23 10:19 rsct.basic-2.3.1.3-0.ppc.rpm
-rw-r--r--    1 root     root      5550757 Sep 23 10:19 rsct.core-2.3.1.3-0.ppc.rpm
-rw-r--r--    1 root     root       950877 Sep 23 10:19
rsct.core.utils-2.3.1.3-0.ppc.rpm
-rw-r--r--    1 root     root       465261 Sep 23 10:19 src-1.2.0.3-0.ppc.rpm
```

Copy the two following requisite packages and place them in the /tmp/csm/reqs subdirectory:

- OpenCIMOM (available from the following website).
  http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html
- Autoupdate 4.3.4 or higher (available from the following website).
  http://freshmeat.net/projects/autoupdate

The `ls` command on /tmp/csm/reqs subdirectory is shown in Example 5-7.

*Example 5-7 ls -l command output in /tmp/csm/reqs directory*

```
-rw-r--r--    1 root     root       332141 Sep 23 10:19 tftp-hpa-0.34.tgz
-rw-r--r--    1 root     root       106386 Sep 23 10:19 tftp-hpa-0.34-1.ppc64.rpm
-rw-r--r--    1 root     root       168690 Sep 23 10:19 conserver-7.2.4.tar.gz
-rw-r--r--    1 root     root        94586 Sep 23 10:19 conserver-7.2.4-3.ppc64.rpm
-rw-r--r--    1 root     root     17750534 Sep 23 10:19 IBMJava2-JRE-1.3.1-3.0.ppc.rpm
drwxr-xr-x    3 root     root          536 Oct  9 14:46 ..
-rwxr--r--    1 root     root        92381 Oct 13 10:30 autoupdate-5.2.6-1.noarch.rpm
-rwxr--r--    1 root     root       403858 Oct 17 14:02 openCIMOM-0.7-4.noarch.rpm
drwxr-xr-x    2 root     root          368 Oct 17 15:29 .
```

Install the csm.core package as follows:

```
# rpm -i /tmp/csm/csm.core-*
```

Or, if installing directly from CD-ROM:

```
# mount /dev/cdrom /mnt/cdrom; rpm -i /mnt/cdrom/csm.core-*
```

csm.core install creates and installs binaries and files in the /opt/csm directory

### Create /csminstall partition

We recommend that you create a separate /csminstall partition either on LVM or on your local disk. This partition holds all necessary CSM directories and files required to manage the cluster. We created the partition on an LVM and the method is explained below.

- Create and initialize a partition for LVM use on two SCSI disks:

  # **pvcreate** /dev/sda4 /dev/sda2

- Create a volumegroup on your LVM partition:

  # **vgcreate -v** system /dev/sda4

- Create a local volume in the system volume group:

  # **lvcreate** --size 3G --name csmlv /dev/system

- Create a resiserfs file system on the logical volume:

  # **mkresiserfs** /dev/system/csmlv

- Create /csminstall and mount csmlv:

  # mkdir /csminstall; mount /dev/system/csmlv /csminstall

- The logical volume size can be increased or decreased on the fly (online) using **lvextend** and **resize_resiserfs** etc.

- The logical volumes and volume groups can be viewed/verified using **lvdisplay**, **vgdisplay** commands.

### Running installms to install the CSM server and prerequisites

Now that the csm.core package is installed, the remaining CSM packages are installed using **installms** script. The following steps are required before running installms:

- Update PATH and MANPATH.
- Copy SuSE Linux Operating System CDs to local disk or have the CD-ROMs ready.
- Run installms.

### Update PATH and MANPATH

The csm.core package installs executable files in /opt/csm/bin and csm man pages in /opt/csm/man. In order to have CSM binaries executed from anywhere, update root's .profile to include these two directories for PATH and MANPATH variables, respectively. Update the bash_profile file in root's home directory and edit or add the following two lines as shown in Example 5-8:

*Example 5-8 PATH and MANPATH variables*

```
export PATH=$PATH:/opt/csm/bin
export MANPATH=$MANPATH:/opt/csm/man
```

Log out and relogin or re-execute root's .profile to force the two new variables. Verify that the variables are updated by using both of these commands:

```
# echo $PATH
# echo $MANPATH
```

## Copy the contents of the SuSE CD-ROM to local disk

installms requires you to access SuSE base images to build /csminstall build. This is required while installing nodes using CSM and autoyast. The CD images can either be copied or specified while running installms. The images are copied to local disk first and the directory is given as THE input path while running installms.

To copy THE SuSE CD image onto a local disk:

```
# mount /dev/cdrom/mnt/cdrom;cd cdrom; tar cvf- |(cd /install/sles;tar xvpf -)
```

**Note:** Make sure you have sufficient disk space in the root ( / ) file system before copying the SuSE CD images.

## Running installms

The remaining CSM installation is carried out by the **installms** script. The installms usage is shown in Example 5-9.

*Example 5-9 installms usage*

```
Usage: installms {-p <pkg_path> | -x} [-c] [-f] [-v | -V] [-h]
                 [Attr=value [Attr=value...]]
        -p <pkg_path>      List of directories containing packages (p1:p2:p3)
        -c                 Force copy from the path
        -f                 Force install
        -x                 Do not copy packages.
        -v | -V            Verbose mode
        -h                 Display usage information
        Valid attributes (see the installms man page for details):
                RemoteShell SetupRemoteShell
```

Start the installation:

```
# /opt/csm/bin/installms -v -p /tmp/csm:/install/sles
```

If the SuSE CD-ROM is not copied to local disk and CSM is copied to /tmp/csm, run installms as follow:

```
# /opt/csm/bin/installms -p /tmp/csm:/media/cdrom
```

installms prompts you to swap CD1 and 2 or the Supplementary CD1 as necessary during the install.

Example 5-10 shows the output of the **installms** command run with both CSM and all requisites copied to local disk.

*Example 5-10 Example installms console output*

```
Output log for installms is being written to /var/log/csm/installms.log.
Logging started
./installms -v -p /tmp/csm:/install/sles
Detected Linux distribution SLES 8.1.
Detected CSM distribution "CSM1.3.2".
Creating the required directories for CSM.
Searching for CSM, RSCT and  Director rpms.
Searching for Open Source rpms.
Installing LINUX OS PRE REQUISITIES RPMs.
dhcp-base is already installed.
rsh-server is already installed.
perl is already installed.
dhcp-server is already installed.
pdksh is already installed.
rsync is already installed.
nfs-utils is already installed.
rdist is already installed.
tcl is already installed.
tk is already installed.
Installing expect-5.34-154.ppc.rpm RPMs.
Installing termcap-2.0.8-513.ppc.rpm RPMs.
Installing perl-XML-RegExp-0.03-295.ppc.rpm RPMs.
Installing OPEN SOURCE PRE REQUISITIES RPMs.
Installing conserver
*******************************************************************************
*
CSM has detected that the following tftp rpm is installed:
        tftp-0.29-44
CSM ships with tftp-hpa-0.34-1 by default.
The above tftp rpms may conflict with
tftp-hpa-0.34-1.
You have several options at this point:
1. Remove the above tftp rpms and install tftp-hpa-0.34-1
2. Install tftp-hpa-0.34-1 without removing already installed rpms
        (note: There may be conflicting files with this option, and
       tftp-hpa-0.34-1 may not install correctly.)
3. Keep the above tftp rpms. Continue without installing tftp-hpa-0.34-1
4. Quit
Please choose 1, 2, 3, or 4:
1
Uninstalling tftp-0.29-44.
Installing tftp-hpa-0.34-1.ppc64.rpm RPMs.
IBMJava2-JRE is already installed.
Installing openCIMOM-0.7-4.aix5.1.noarch.rpm RPMs.
Installing RSCT RPMs.
Installing src-1.2.0.3-0.ppc.rpm RPMs.
Installing rsct.core.utils-2.3.1.3-0.ppc.rpm RPMs.
Installing rsct.core-2.3.1.3-0.ppc.rpm RPMs.
Installing CSM RPMs.
csm.core is already installed.
Installing csm.dsh-1.3.2.0-26.ppc.rpm RPMs.
Installing csm.server-1.3.2.0-26.ppc.rpm RPMs.
Installing csm.diagnostics-1.3.2.0-26.ppc.rpm RPMs.
Creating default tftpd user for the tftp server.
Creating default tftpd group for the tftp server.
About to copy CSM command binaries.
Securing file permissions in the /tftpboot directory and all subdirectories...
Installation of CSM has successfully completed!
Logging stopped
```

The complete installms log is written to the /var/log/csm/installms.log file.

### 5.2.6 Installing the CSM license

The CSM license is available in two options: a 60-day try and buy license available with the CSM products package, and a full production license and key shipped in a CD-ROM when purchased.

Once installms installation is complete, the try and buy license is accepted by running:

```
# csmconfig -L
```

The status and Expiration date can be checked with the **csmconfig** command. The Expiration date attribute (ExpDate) is populated only if it is a try and buy license.

Example 5-11 on page 234 shows **csmconfig** command output.

*Example 5-11 csmconfig command output*

```
AddUnrecognizedNodes = 0 (no)
 ClusterSNum =
 ClusterTM = 9078-160
 ExpDate = Mon Dec 15 18:59:59 2003
 HeartbeatFrequency = 12
 HeartbeatSensitivity = 8
 MaxNumNodesInDomain = -1 (unlimited)
 RegSyncDelay = 1
 RemoteShell = /usr/bin/ssh
 SetupRemoteShell = 1 (yes)
```

If a full production license is purchased, it is accepted by running:

```
# csmconfig -L /mnt/cdrom/csmlum.full
```

At the prompt, follow the directions to accept the license. Again, the **csmconfig** command shows the status of the license and the ExpDate Attribute shows blank.

### 5.2.7 Install verification

Ensure that the management server is installed correctly and is ready for use before starting any configuration.

Check the following to verify installation:

- Check the /var/log/csm/installms.log for errors, and fix any that you find.
- Check the installed software:
  ```
  #rpm -qa|grep csm
  ```

Example 5-12 shows CSM packages output.

*Example 5-12 CSM packages*

```
csm.dsh-1.3.2.0-26
csm.core-1.3.2.0-26
csm.diagnostics-1.3.2.0-26
csm.server-1.3.2.0-26
```

```
#rpm -qa | grep rsct
```

Example 5-13 shows RSCT packages output.

*Example 5-13 RSCT packages*

```
rsct.core.utils-2.3.1.3-0
rsct.core-2.3.1.3-0
```

- Verify that the RSCT subsystems are started and running:

  ```
  # lssrc -a | grep rsct
  ```

Example 5-14 shows the RSCT subsystems output.

*Example 5-14 RSCT subsystems*

```
ctrmc            rsct             1139    active
 IBM.DMSRM        rsct_rm          1268    active
 IBM.ERRM         rsct_rm          1269    active
 IBM.HWCTRLRM     rsct_rm          1270    active
 IBM.AuditRM      rsct_rm          1309    active
 ctcas            rsct             11490   active
 IBM.HostRM       rsct_rm          11497   active
 IBM.SensorRM     rsct_rm          11507   active
 IBM.CSMAgentRM   rsct_rm                  inoperative
```

- Issue the **csmconfig** command to ensure that CSM is running. The output should look similar to Example 5-11 on page 234.
- If any CSM software updates are not installed during installms, those can be installed now.
- The contents of /tmp/csm can now be deleted.

## 5.2.8 CSM management server configuration

The management server requires very minimal configuration after the installation and is basically ready for use once installms and the install verification steps outlined in 5.2.7, “Install verification” on page 234 are successfully completed.

To display current configuration and modify any attributes, enter the **csmconfig** command; the output is shown in Example 5-15.

*Example 5-15 csmconfig command output*

```
AddUnrecognizedNodes = 0 (no)
 ClusterSNum =
 ClusterTM = 9078-160
 ExpDate = Mon Dec 15 18:59:59 2003
 HeartbeatFrequency = 12
 HeartbeatSensitivity = 8
 MaxNumNodesInDomain = -1 (unlimited)
 RegSyncDelay = 1
 RemoteShell = /usr/bin/ssh
 SetupRemoteShell = 1 (yes)
```

The attributes are briefly explained in this section:

***AddUnrecognizedNodes***

A default value of -0- (no) indicates that the management server should not accept requests from unrecognized nodes to manage them and add them to cluster.

***ClusterSNum***

The Cluster Serial number attribute is used for service.

***ClusterTM***

The Cluster type and model is used for service and is in the form of: ####-###.

***ExpDate***

This is the Expiration date for the try and buy license. This is blank when a Full license is accepted with the -L option.

***HeartbeatFrequency***

This refers to the number of seconds between heartbeat messages sent to a node from the management server. The valid value range is 1 to 900, and the default is 12. RMC is responsible for monitoring heartbeats using the node hostname.

***HeartbeatSensitivity***

This is the number of missed heartbeat messages sent to a node to declare the node unreachable. The valid value range is 2 to 100, and the default is 8. The Node status attribute is changed from 1 to 0 when the node is unreachable.

***MaxNumNodesInDomain***

This is the maximum number of nodes allowed in the CSM cluster. This is determined by license key.

***RegSyncDelay***

This refers to the delay in seconds from when cluster data is updated in memory to when it is written on disk.

***RemoteShell***

This is the path of the remote shell that CSM uses to communicate with nodes. The default is ssh.

***SetupRemoteShell***

This indicates whether CSM can set remote shell security automatically on remote node at installtime. The default is 1 (yes).

### *Adding Cluster SerialNumber*

Modify the cluster serial number by entering:

```
# csmconfig -s abc1000
```

## 5.2.9 Defining/adding nodes to the CSM cluster

In this section, we describe how to define and add cluster-managed nodes to the CSM database, and what attributes are stored for each node.

Managed nodes can be either installed with CSM (which installs the operating system (Linux) and all required CSM packages), or the operating system can be pre-installed and CSM used exclusively to install CSM packages only. We will discuss both these options.

Defining and adding nodes includes these tasks:

- Defining a Hardware Control Point for nodes connected by HMC
- Updating /etc/hosts or Name resolution
- Generating node information and creating a node definition file
- Populating CSM database with node definition file
- Modifying node attributes

### Defining a Hardware Control Point

For pSeries standalone servers and LPARs managed by the Hardware Management Console (HMC), CSM communicates directly with the HMC by using an attribute called the Hardware Control Point.

Prior to using any hardware control, the IP address of the HMC, userid and valid password are defined using command **systemid**. The command basic syntax and result are displayed in Example 5-16.

*Example 5-16 Example of systemid*

```
# systemid hmc_hostname hscroot
Password: ******
Verifying, please re-enter password:******
       systemid: Entry updated.
```

Wherein **hmc_hostname** is the resolvable hostname of HMC and **hscroot** is the userid to connect to the HMC. The userid and encrypted password are stored in the /etc/opt/system_config directory on the management server. For further information, refer to the man page of systemid.

### Updating /etc/hosts or name resolution

A standard hostname resolution must be implemented before attempting to define nodes. This can be either /etc/hosts for local, or /etc/resolv.conf for Domain Name Server (DNS) resolution. All the managed nodes, hardware control point and management server are resolvable for an internal cluster and managed VLANs as applicable.

If planning a DNS named implementation, refer to Chapter 3, “System administration” on page 95, for detailed information.

### Generating node information - creating a node definitions file

Node information can be generated for HMC-attached nodes automatically. This data can be used to create a node definition file for multiple nodes very easily.

**Note:** For non-HMC-attached nodes, the file has to be created manually. Here, we discuss HMC-attached nodes.

Node information is gathered using the command **lshwinfo**. Sample command and output are shown in Example 5-17.

*Example 5-17 lshwinfo output*

```
# lshwinfo -p hmc -c hmc_hostname -o /tmp/nodes.dat
#Hostname::PowerMethod::HWControlPoint::NodeId::LParId::HWType::HWModel::HWSeri
alNum
no_hostname::hmc::hmc_hostname::lpar1::002::7038::6M2::10197CA
no_hostname::hmc::hmc_hostname::lpar2::002::7038::6M2::10197BA
```

The first column indicates no_hostname, as these nodes are not being defined in the CSM database yet. Edit and replace no_hostname with the proper hostname of each LPAR. The updated /tmp/nodes.dat file resembles Example 5-18.

*Example 5-18 Updated nodes.dat*

```
#
Hostname::PowerMethod::HWControlPoint::NodeId::LParId::HWType::HWModel::HWSeria
lNum
node1::hmc::hmc_hostname::lpar1::002::7038::6M2::10197CA
node2::hmc::hmc_hostname::lpar2::002::7038::6M2::10197BA
```

All the node hostnames in the file, such as node1 and node2, must be resolvable either in local hosts file or on DNS. For more information about the **lshwinfo** command, refer to its man page.

**Important:** Make sure the HMC is properly connected to the LPAR frame using an RS232 serial adapter, and ensure that the management server communicates with the HMC before running **lshwinfo**. This command fails if proper network connectivity does not exist.

Once the datafile is generated, it is used to create a node definitions file by using the **definenode** command. Important node attributes are added using definenode and a definition file is created in the right format to populate the CSM database.

Example 5-19 shows the creation of a node definition file.

*Example 5-19 Example definenode*

```
# definenode -s -M /tmp/nodes.dat >/tmp/nodes.def
# cat /tmp/nodes.def
node1 :
        ConsoleMethod=hmc
        ConsoleSerialDevice=ttyS0
        ConsoleServerName=hmc_hostname
        HWControlNodeId=lpar1
        HWControlPoint=ms_hostname
        HWModel=6M2
        HWSerialNum=10197AA
        HWType=7038
        InstallCSMVersion=1.3.2
        InstallDistributionName=SLES
        InstallDistributionVersion=8.1
        InstallOSName=Linux
        InstallPkgArchitecture=ppc64
        LParID=001
        ManagementServer=p630sles
        PowerMethod=hmc
node2 :
        ConsoleMethod=hmc
        ConsoleSerialDevice=ttyS0
        ConsoleServerName=hmc_hostname
        HWControlNodeId=lpar2
        HWControlPoint=ms_hostname
        HWModel=6M2
        HWSerialNum=10197AA
        HWType=7038
        InstallCSMVersion=1.3.2
        InstallDistributionName=SLES
        InstallDistributionVersion=8.1
        InstallOSName=Linux
        InstallPkgArchitecture=ppc64
```

```
LParID=002
ManagementServer=p630sles
PowerMethod=hmc
```

The **-s** option in definenode redirects the output to standard out, and the **-M** option reads from the mapping data file. For non-HMC-managed nodes, the same definition file can be manually created with similar attribute values. Refer to *IBM Cluster System Management Planning and Installation Guide,* SA22-7853 for detailed explanations of which attributes are required for manual node definition creation.

Once the definition file is created, the CSM database is populated with the node definitions. This is done using the following command:

```
# definenode -f /tmp/nodes.def
```

Verify the CSM database using **lsnode** and **lsnode -l** commands.

> **Note:** definenode simply appends the definitions to the CSM database. If you have a typo and/or if you ran definenode -f multiple times, the same node definitions are added multiple times and this can fail the node install.
>
> To avoid this, verify by using the **lsnode** command that node definitions are accurate before progressing to the next step. If duplicate entries are found, clean up the CSM database by using the **rmnode** command to delete duplicate node entries.

### Modifying node attributes

Node attributes added using definenode can be modified by using the **chnode** command. This command changes node attributes defined in the CSM database.

## 5.2.10 Preparing for node installation

The target install nodes defined and added in the CSM database are prepared using the **csmsetupyast** command. This command collects all configuration information from sources such as the SuSE CD-ROM and the default yast config file, and sets up the AutoYast configuration file to install the Linux nodes.

However, you should first update the kernel for node installation before you run the **csmsetupyast** command (because some models will experience problems with the original kernel in SLES). Then you need to copy the new kernel rpm to /csminstall/Linux/SLES/8.1/ppc64/updates/kernel/. **csmsetupyast** will get the kernel image from the latest RPM by the number of the RPM name.

The **csmsetupyast** command does the following:

- It sets up and populates the /csminstall directory path by copying Linux packages from a CD-ROM or directory.
- It resolves the node hostnames to be installed, and runs the **getadapters** command to retrieve the MAC or hardware addresses of the node's network adapters.
- It creates the dhcpd.conf file and starts dhcpd.
- It sets up the kernel+initrd image.
- It sets up the AutoYast xml config file.
- It modifies the InstallMethod node attributes in the CSM database.

The usage of the **csmsetupyast** command is illustrated in Example 5-20.

*Example 5-20 csmsetupyast usage*

```
Usage:  csmsetupyast [-h]
        csmsetupyast [-v | -V] [-p pkg_path | -x] [-k yastcfg_file]
                   [-P | -a | [-N node_groups] [-n node_list]]
                   [Attr=value [Attr=value...]]
        -a       Setup AutoYaST for all nodes.  This cannot be used with
                 the -n, -N or -P flags.
        -h       Writes usage information to standard output
        -n  node_list
                 Provide a comma-separated list of nodes and node ranges to set
                 up AutoYaST for.  (See the noderange man page for information
                 on node ranges.)  This option cannot be used with the -a or -P
                 options.
        -N  node_groups
                 Provide a comma-separated list of node groups to set up
                 AutoYaST for.  This cannot be used with the -a or -P flags.
        -p  pkg_path
                 Specifies one or more directories, separated by colons, that
                 contain copies of the SuSE/SLES CD-ROMs.  The default is
                 /media/cdrom.  This cannot be used with the -x flag.
        -P       Setup AutoYaST for all nodes whose Mode attribute is
                 "PreManaged".  This cannot be used with the -a, -n or -N
flags.
        -v | -V  Writes the verbose messages of the command to standard output.
        -x       Specifies to not copy SuSE/SLES disks.  This cannot be used
                 with the -p flag.
        -k  yastcfg_file
                 Specifies a AutoYaST control file. The
                 default is /opt/csm/install/yastcfg.SLES<version>.xml.
        Valid attributes (see the csmsetupyast man page for details):
                Netmask
                Gateway
```

```
              Nameserver
              Nameservers
```

Detailed information on each of the options can be found in the **csmsetupyast** man page

## Populating /csminstall

The /csminstall directory contains all the necessary information to install nodes such as configuration information, updated RPM packages, and the prerequisites required. Figure 5-6 shows the contents of /csminstall mainly populated during CSM installation on the management server and while running the **csmsetupyast** command.

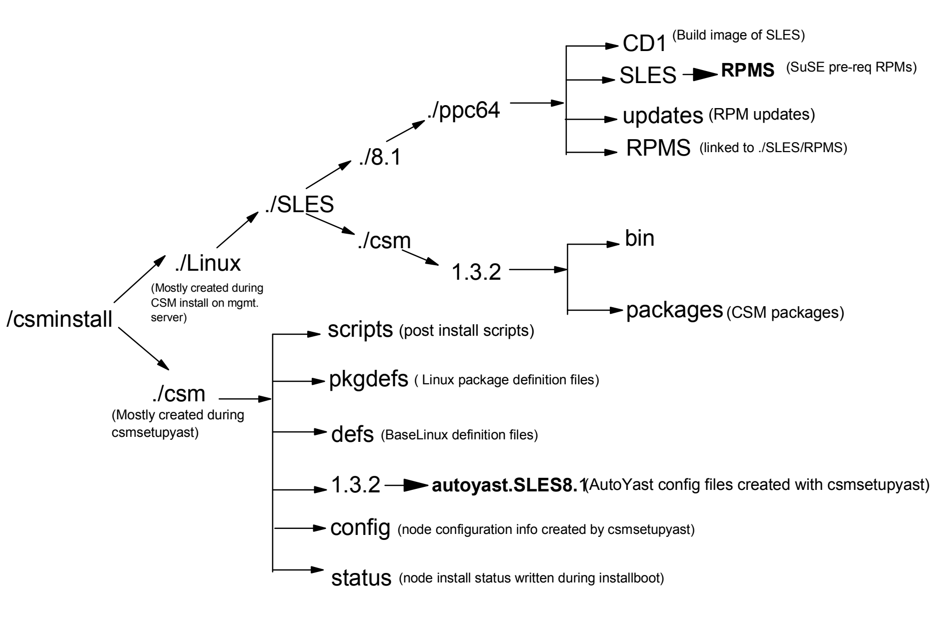

/csminstall
./Linux
(Mostly created during CSM install on mgmt. server)
./SLES
./8.1
./ppc64
CD1 (Build image of SLES)
SLES
RPMS (SuSE pre-req RPMs)
updates (RPM updates)
RPMS (linked to ./SLES/RPMS)
./csm
1.3.2
bin
packages (CSM packages)
./csm
(Mostly created during csmsetupyast)
scripts (post install scripts)
pkgdefs ( Linux package definition files)
defs (BaseLinux definition files)
1.3.2
autoyast.SLES8.1(AutoYast config files created with csmsetupyast)
config (node configuration info created by csmsetupyast)
status (node install status written during installboot)

*Figure 5-6 /csminstall directory tree*

## Host names and getadapters

Host names defined in the CSM database are mapped to IP addresses from local hosts file or DNS, and some network adapter information such as MAC addresses, adapter speed, duplex setting are obtained from target install nodes.

The CSM command **getadapters** collects network MAC addresses of the first adapter installed on target nodes, and it can be run separately before running **csmsetupyast**. The **getadapters** command will issue **dsh** to the node to get the

MAC address—if the node is on and has a running operating system. Otherwise, it will power-on the node and get the MAC address from the openFirmware by performing a passing ping test of the adapter.

Network information such as MAC address, adapter type, speed, and duplex setting can also be specified in a file and passed to getadapters instead of powering off/on the nodes. The same information is written to the CSM database by using the **-w** option.

Network attributes obtained in any other way are also be used to modify the CSM database by using the **chnode** command. **csmsetupyast** parses the node attributes database and if any network information is found, it skips running getadapters and proceeds to next step.

Example 5-21 shows common network attributes acquired and added to the CSM node attributes database.

*Example 5-21 CSM node network attributes*

```
InstallAdapterDuplex = half
 InstallAdapterGateway =
 InstallAdapterMacaddr = 00:02:55:3A:06:8C
 InstallAdapterNetmask =
 InstallAdapterSpeed = 100
 InstallAdapterType = ent
```

## DHCP configuration

The management server is configured as a DHCP server to assign IPs to target nodes during netboot. All the information required is obtained during **csmsetupyast** and a default dhcpd.conf file is written and the dhcpd daemon is started. Refer to Chapter 3, “System administration” on page 95 for more information about DHCPD.

Example 5-22 shows a sample dhcpd.conf file written for node lpar1.

*Example 5-22 /etc/dhcpd.conf file*

```
# This file has been automatically generated by IBM CSM
# Please do not modify or erase lines that contain "# CSM"
# such as the following line:
# CSM VERSION 1.3.2.0 (Please do not remove this line)


ddns-update-style none;
shared-network CSM {
option routers 192.168.100.60;
subnet 192.168.100.0 netmask 255.255.255.0 {
```

```
                # CSM RANGE 192.168.100.0 (Please do not remove this line)
                default-lease-time -1;
                filename "/pxelinux.0";
                next-server 192.168.100.110;
}
}
group { ### CSM STATIC ENTRIES (Please do not remove this line)
                host lpar1 {
                        hardware ethernet 00:02:55:3A:06:8C;
                        fixed-address 192.168.100.77;
                        filename "/csm/yast8.1-ppc64-zImage.initrd";
                        option root-path
"/csminstall/Linux/SLES/8.1/ppc64/CD1";
                }
```

## /tftpboot updates

CSM updates the /tftpboot directory during csmsetupyast processing with the kernel and boot loader information collected from **initrd** and boot images. This image is used to boot the target nodes during the netboot node condition.

## AutoYast XML config file

The default Yastconfig file is located at /opt/csm/install/yastcfg.SLES<version>.xml.

It contains the following information:

- Default inetd services
- Default bootloader information
- Default user and encrypted password
- RPM packages to install
- Partition information
- LVM information

This information can be modified and passed as an argument by using the **-k** option to **csmsetupyast**. If the **-k** option is not selected, a default XML file is used as input. This information is used to generate an AutoYast config file and is written to the /csminstall/csm/csm_version/autoyast.version/ directory for each target node with a node IP or hostname prefix.

**Tips:**

- The Yast config file is XML-formatted. Use an XML editor such as "konqueror" or "emacs" to ensure that the modified XML file is syntax-free before passing it to **csmsetupyast**.

Use the following tips if you are planning to modify default partition information:

- Do not create a PrepBoot partition that is more than 8 mb in size.
- Create a / (root) partition outside of Logical Volume Manager (LVM).

### Other node attributes

Other node attributes, such as InstallMethod, are modified during the **csmsetupyast** run. These attributes are referred during installing target nodes.

**Tip:** For error-free updates, check the following before running **csmsetupyast**:

- Run an **rpower -a** query to make sure all target nodes respond. If errors or <unknown> are reported, fix open CIMOM or HMC errors before proceeding. getadapters powers on/off target nodes and exits if failed.
- Run **rconsole -r -t -n lpar1** to see if a serial console can be opened. Fix errors such as refreshing conserver or corrupted conserver.cf before proceeding.
- Open the HMC webSM GUI and "close all virtual terminals open" to force closure of all console windows. Sometimes **csmsetupyast** may exit with the error message: `virtual term already open`.

Issue the following command to activate **csmsetupyast** on nodes lpar1 and lpar3:

```
# /opt/csm/bin/csmsetupyast -v -k working.xml -p /install/sles -n lpar1,lpar3
```

Verify the /var/log/csm/csmsetupyast.log and **lsnode -l** node_name to make sure no errors are reported and that the node information is defined accurately.

## 5.2.11 Installing managed nodes

Once **csmsetupyast** is successfully completed and all the node information is defined in the CSM database, the target nodes are ready to have the operating system and CSM installed, using the **installnode** command.

**Installnode** performs the following actions:

- It reboots the node and each node broadcasts its MAC address while rebooting.
- The management server’s dhcpd accepts dhcp requests and adds an arp entry to have the nodes rebooted in firmware mode.
- AutoYast initiates and completes the node install.
- AutoYast runs post-installation custom scripts and modifies /etc/inittab with csmfirstboot and reboots the node from the local hard drive.
- The csm boot script runs **makenode** on the node to install CSM and configure the node as a managed node by exchanging ssh public keys.
- Once makenode completes, the Mode attribute is changed from Installing to Managed.

When a node is defined in the database but not yet installed, the Mode attribute is set to “PreManaged”. Once a node is successfully installed using CSM, this attribute is changed to “Managed”, at which time CSM considers this as a managed node. If the mode is set to “MinManaged”, then it remains as MinManaged even after the installation.

If the installation fails for any reason, the attribute either remains as “PreManaged”or is changed to “Installing”, depending at which stage the installation failed. The Mode attribute must be either “PreManaged” or “MinManaged” in order to have installnode started.

**Tip:** If installation fails in the middle for any reason and another run of installnode fails immediately, check the Mode attribute for the node and run this command if it is set to “Installing”:

```
#chnode Mode=PreManaged -n node_name
```

The usage of **installnode** is shown in Example 5-23:

*Example 5-23 installnode usage*

```
Usage: installnode [-h]
        installnode [-v | -V] [ -P | -a | [-N node_groups] [[-n]
node_list][--file filename] ]
        -a          install all nodes whose InstallMethod attribute is
                    kickstart.  This flag cannot be used with the -P or -N
                    flags or the node_list.
        -h          Display this usage information.
        [-n] node_list
                    Space or comma separated list of nodes and node ranges.
                    (See the noderange man page for information on node
                    ranges.)  This option cannot be used with the -a or -P
                    options.  Node names may be specified as positional
```

```
            arguments or preceded by the -n flag.
-N node_groups
            Comma-separated list of node groups to install.  This
            flag cannot be used with the -a or -P flags.
--file filename
           specifies a file that contains a list of nodes names.
           If the file name "-", then the list is read from stdin.
           The file  can contain multiple lines and each line can have
           one or node names, separated by spaces.
-P          install all nodes whose Mode attribute is PreManaged and
            whose InstallMethod attribute is kickstart.  This flag
            cannot be used with the -a or -N flags or the node_list.
-v | -V     Verbose mode
```

**Note:** Before running installnode, make sure to have dhcpd, nfs and tftp daemons started. If not, they can be started as:

- service dhcp start
- service nfs start
- service xinetd start

Issue the **installnode** command to start the install on all PreManaged nodes:

```
# installnode -v -P
```

**Note:** The **installnode** command also runs **smsupdatenode** and **cfmupdatenode** at the end of the install to push software maintenance packages and distribute shared common CFM files.

If you do not want this to happen during **installnode** processing, do not populate the /cfmroot and /csminstall/Linux/OS/version/architecture/updates directory.

Example 5-24 shows the standard output of **installnode**.

*Example 5-24   installnode output*

```
#installnode -n lpar3
Resolving names in the specified node list...
Running cmd: /usr/bin/lsrsrc-api -D ':|:' -i -s IBM.ManagedNode::"Hostname IN
('lpar3')"::Name::Hostname::ManagementServer::Install1
Output log for installnode is being written to /var/log/csm/installnode.log.
Loading /opt/csm/install/pkgdefs/Linux-SLES8.1.pm.
Status of nodes before the install:
```

```
Running cmd: /opt/csm/bin/monitorinstall 2>&1
 Node                   Mode                  Status
--------------------------------------------------------------------------
lpar1                  Managed               Installed
lpar3                  PreManaged            AutoYaST Post-Install Complete

Nodes to install:
        lpar3
Running cmd: /usr/bin/lsrsrc-api -i -s IBM.DmsCtrl::::SetupRemoteShell 2>&1
Running cmd: /usr/bin/lsrsrc-api -i -s IBM.DmsCtrl::::RemoteShell 2>&1
Running cmd: /opt/csm/csmbin/remoteshell.expect -k 2>&1
Copying OpenSSH public keys to target nodes.
Running cmd: /bin/cp /root/.ssh/identity.pub /csminstall/csm/con-
fig/.ssh/authorized_keys
Running cmd: /bin/cp /root/.ssh/id_rsa.pub /csminstall/csm/con-
fig/.ssh/authorized_keys2
Running cmd: /bin/cat /root/.ssh/id_dsa.pub >> /csminstall/csm/con-
fig/.ssh/authorized_keys2

CSM_FANOUT = 16, CSM_FANOUT_DELAY = 1200 .
Rebooting Node for full install: lpar3
spawn /opt/csm/bin/rconsole -t -f -n lpar3
[Enter `^Ec?' for help]
..
          1 = SMS Menu                            5 = Default Boot List
          6 = Stored Boot List                    8 = Open Firmware Prompt


     memory      keyboard     network     scsi     speaker  ok
0 > dev /  ok
0 > ls
000000d9adf0:     /ethernet@1
 ok
0 > " local-mac-address" 000000d8f788 get-package-property  ok
3 > . 0  ok
2 > dump
000000d9aac0: 00 02 55 3a 06 19 :..U:..: ok
0 > boot /pci@400000000111/pci@2,2/ether-
net@1:speed=100,duplex=half,bootp,192.168.100.110,,192.168.100.79,0.0.0.0 auto-
yast=nfs://19
BOOTP: chosen-network-type = ethernet,Status of nodes after the install:
```

```
Running cmd: /opt/csm/bin/monitorinstall 2>&1

Node                    Mode                  Status
--------------------------------------------------------------------------
lpar1                   Managed               Installed
lpar3                   Installing            Rebooting and Installing Node.
#
```

**Note:** **`installnode`** runs /opt/csm/csmbin/hmc_nodecond, an expect script that passes SMS menu arguments while netbooting in firmware mode. Add the following two lines to the script to run in debug mode to capture more information.

```
log_user 1
exp_interval 1
```

Node installation is monitored in several ways:

- The **`monitorinstall`** command is used to monitor the status of the node install. Install Status and Mode status attributes are flagged from the CSM database and shown as standard out. Issue the following command to run monitor install continuously.

  ```
  # watch monitorinstall
  ```

  Sample monitorinstall is shown in Example 5-25.

*Example 5-25 monitorinstall output*

```
Node                    Mode                  Status
--------------------------------------------------------------------------
lpar1                   Managed               Installed
lpar3                   Installing            AutoYaST Post-Install Complete
```

- Also monitor the installation by opening a read only console window. (Do not open a read write window at the beginning, as this can interrupt the install.) Open a read only console by typing:

  ```
  # rconsole -r -t -n lpar1
  ```

Check the man page of **`rconsole`** for a list of detailed options. This command is used only for HMC-attached pSeries servers.

### Making pre-installed nodes into managed nodes

For nodes that have Linux pre-installed, and you only want to install CSM and make them managed nodes after you complete defining/adding nodes, the CSM **updatenode** command is used instead of installnode.

This command installs CSM only on PreManaged nodes and exchanges ssh public keys with the node. It runs **makenode** to mount CSM package directories NFSmounted to the target node, and installs CSM client packages. Once complete, the public keys are exchanged and the status of the node is changed from PreManaged to Managed.

**Tips:**

- If **updatenode** fails to exchange keys after installing CSM client filesets, the problem could be hostname resolution for that node. Check your /etc/hosts or DNS node definitions on the management server to resolve the node's hostname.
- Run **updatenode** with the **-k** option only to have RSCT public keys exchanged between the node and the management server. This is useful if the node's host name is changed and the management server no longer communicates with the node's old hostname.

## 5.2.12 Node install verification

Verify successful node installation as follows:

- Check the /var/log/csm/installnode.log.
- Run monitorinstall. As shown in Example 5-25,the Status column of monitorinstall shows the node as **Installed**.
- Run **dsh** -as date to display the date on all installed nodes.
- Run **lsnode** -H -l node-name to see if RMC is responding for the respective managed node. Sample output is shown in Example 5-26.

*Example 5-26 lsnode to find out RMC response*

```
#lsnode -H -l lpar1
Name = lpar1
 ActiveMgtScopes = 1
 Architecture = ppc64
 DistributionName = SLES
 DistributionVersion = 8.1
 KernelVersion = 2.4.21-83-pseries64
 LoadAverage = {0,0,0}
 NodeNameList = {lpar1}
 NumProcessors = 2
```

```
NumUsers = 1
OSName = Linux
PctRealMemFree = 95
PctTotalPgSpFree = 100
PctTotalPgSpUsed = 0
PctTotalTimeIdle = 99.8824
PctTotalTimeKernel = 0.071471
PctTotalTimeUser = 0.0461264
PctTotalTimeWait = 0
RealMemSize = 4190330880
TotalPgSpFree = 524284
TotalPgSpSize = 524284
UpTime = 1067381449
VMPgInRate = 16
VMPgOutRate = 74
VMPgSpInRate = 0
VMPgSpOutRate = 0
```

## 5.3 CSM administration

In 5.1, “CSM concepts and architecture” on page 212, we touch on the topics of CSM management and administration as a basic introduction to the main features of CSM and how they function.

In this section, we examine these administration topics in detail by using examples and sample scenarios, and discuss the following areas:

- Log file management
- Managing Node groups
- Hardware control
- Cluster File Manager
- Software Management System
- CSM monitoring
- Diagnostic probes
- CSM backup
- Querying CSM database
- CSM problem determination and diagnostics
- CSM hostname changes

### 5.3.1 Log file management

CSM logs to several different log files during installation and cluster management. These log files are available on the management server and managed nodes, and they help to determine the status of a command, or in troubleshooting a CSM issue.

Most of the CSM log files on the management server are located in the /var/log/csm directory. Table 5-1 lists the log files on the management server and their purpose.

*Table 5-1 Log files on management server*

| Log File | Purpose |
|---|---|
| /var/log/csm/install.log | csm.core install log |
| /var/log/csm/installms.log | Output of installms command |
| /var/log/csm/installnode.log | Verbose output of installnode command |
| /var/log/csm/installnode.node.log.* | hmc_nodecond out of each node's installation |
| /var/log/csm/csmsetupyast.log | Output of csmsetupyast command |
| /var/log/csm/updatenode.log | Output of updatenode command |
| /var/log/csm/smsupdatenode.log | Output of smsupdatenode command |
| /var/log/csm/cfmerror.log | CFM error log |
| /var/log/csm/cfmchange.log | Output of CFM file updates |
| /var/log/csm/hw_logfile | HW control daemon status log |
| /var/log/csm/hmc[IP_address].log.* | HMC communication error messages |
| /var/log/csm/hmc[IP_address].java_trace | Tracing for openCIMOM calls to HMC |
| /var/log/csm/hmc_logfile.314 | Tracing for libhmc_power.so |
| /var/log/csm/getadapters/getadapters.node.log.* | Output of getadapters command for each node |
| /var/ct/RMstart.log | Resource Manager status log |
| /var/ct/*.stderr | RSCT daemons standard errors |
| Linux log files in /var/logs | Refer to problem determination section 3. |
| Other Linux log files | Refer to problem determination section 3 |

Table 5-2 on page 253 lists log files on managed nodes and their purpose.

*Table 5-2 Log files on managed nodes*

| Log file | Purpose |
|---|---|
| /var/log/csm/install.log | csm.core install log |
| /var/log/csm/updatekernel.log | Kernel update log running smsupdatenode |

### 5.3.2 Node groups

Managed nodes can be grouped together by using the **nodegrp** command. Distributed commands can be issued against groups for common tasks, instead of performing them on each node. Default node groups created at install time are shown in Example 5-27.

*Example 5-27 nodegrp command*

```
# nodegrp
ManagedNodes
AutoyastNodes
ppcSLES81Nodes
AllNodes
SuSE82Nodes
SLES72Nodes
pSeriesNodes
SLES81Nodes
LinuxNodes
PreManagedNodes
xSeriesNodes
EmptyGroup
APCNodes
RedHat9Nodes
MinManagedNodes
```

Node groups are created with the **nodegrp** command:

```
#nodegrp -a lpar1,lpar2 testgroup
```

This creates a group called test group which includes nodes lpar1 and lpar2. For more information, refer to the nodegrp man page.

Distributed commands such as dsh can be run on nodegroups:

```
# dsh -w testgroup date
```

### 5.3.3 Hardware control

The CSM hardware control feature is used to remotely control HMC-attached pSeries servers. Remote nodes can be powered on, off, the power status can be queried, and you can open a remote console from the management server.

It is mandatory to have all pSeries servers connected to HMC, and to have the HMC communicate with the management server for the hardware control function. Figure 5-7 shows hardware control feature design for a simple CSM cluster.

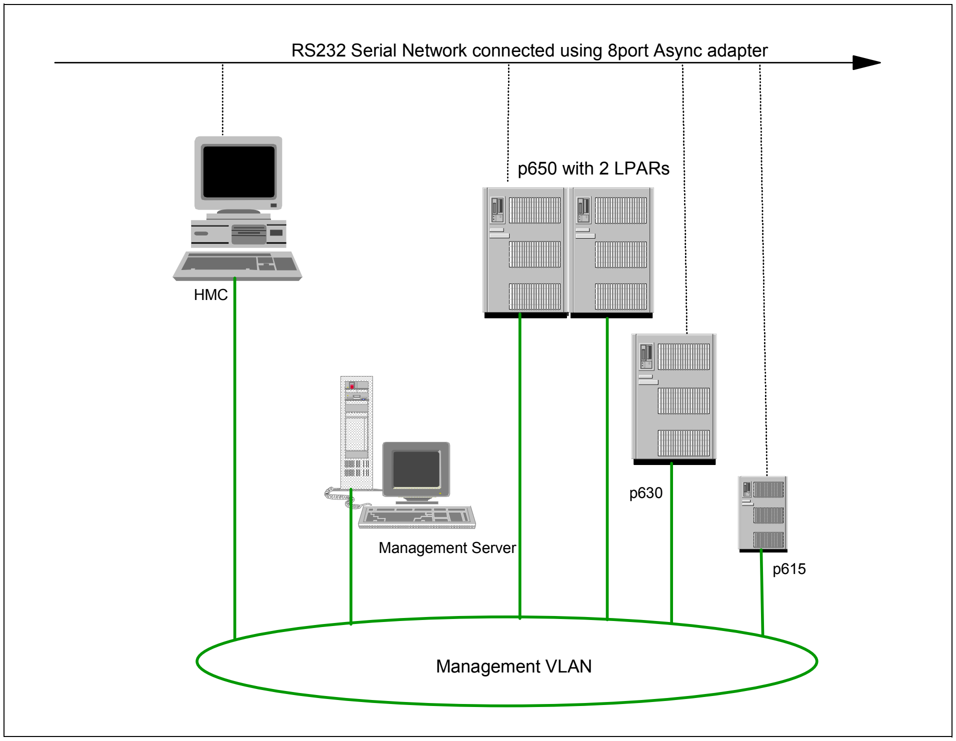


*Figure 5-7 pSeries CSM cluster with hardware control using HMC*

Hardware control uses openCIMOM (public software) and conserver software to communicate to HMC to issue remote commands. The IBM.HWCTRLRM daemon subscribes and maintains state to HMC openCIMOM events during startup. Conserver is started at boot time on the management server and reads from a defined config file located at /etc/opt/conserver/conserver.cf.

The following hardware control commands are available on the management server:

r**power** Powers nodes on and off and queries power status

**rconsole** Opens a remote serial console for nodes

**chrconsolecfg** Removes, adds and re-writes conserver config file entries

**rconsolerefresh** Refreshes conserver on the management server

**getadapters** Obtains MAC addresses of remote nodes

**lshwinfo** Collects node information from Hardware Control points

**systemid** Stores userid and encrypted password required to access remote hardware

The `rpower` and `rconsole` commands are frequently used hardware control commands and we discuss them in detail here:

## Remote power

Remote power commands access the CSM database for node attribute information.

**PowerMethod** Node attribute must be set to hmc to access pSeries nodes.

**HardwareControlPoint** is the hostname or IP address of the Hardware Management Console (HMC).

**HardwareControlNodeId** is the hostname or IP address of the managed node which is attached to the HMC over a serial link.

Other Node attributes such as HWModel, HWSerialNum, HWType are obtained automatically using `lshwinfo`.

Remote power configuration is outlined in 5.2.5, “Installing CSM on the management server” on page 228.

## Remote console

The Remote console command communicates with the console server to open remote console to nodes using management VLAN and Serial connections. The HMC works as the remote console server listening for requests from the management server.

Only one read write console, but multiple read only consoles, can be opened to each node by using the `rconsole` command.

## 5.3.4 Configuration File Manager (CFM)

Configuration File Manager (CFM) is a CSM component to centralize and distribute files across management nodes in a management cluster. This is similar to file collections on IBM PSSP. Common files such as /etc/hosts across the cluster are distributed from the management server using a push mechanism through root's crontab and/or event monitoring. CFM uses **rdist** to distribute files. Refer to 5.1.7, "CSM diagnostic probes" on page 220 for more information on hostname changes.

CFM uses /cfmroot as its main root directory, but copies all files to /etc/opt/csm/cfmroot with a symlink on the management server. File permissions are preserved while copying. Make sure that you have enough space in your root directory or create /cfmroot on a separate partition and symlink it from /etc/opt/csm/cfmroot.

Example 5-28 shows cfmupdatenode usage.

*Example 5-28 cfmupdatenode usage*

```
Usage: cfmupdatenode [-h] [-v | -V]
                [-a | -N node_group[,node_group] | --file file ] [-b]
                [[-y] | [-c]] [-q [-s] ] [-r remote shell path]
                [-t timeout] [-M number of max children]
                [-d location for distfile] [-f filename] [[-n] node_list]
      -a        Files are distributed to all nodes. This option cannot be
                used with the -N or host positional arguments.
      -b        Backup. Preserve existing configuration file (on nodes) as
                "filename".OLD
      -c        Perform binary comparison on files and transfer them
                if they differ.
      -d distfile location
                cfmupdatenode will generate a distfile in the given (absolute)
                path and exit (without transferring files). This way the user
                can execute Rdist with the given distfile and any options
                desired.
      -f filename
                Only update the given filename. The filename must be the
                absolute path name of the file and the file must reside in
                the cfmroot directory
      --file filename
                specifies a file that contains a list of nodes names. If the
                file name "-", then the list is read from stdin. The file
                can contain multiple lines and each line can have one or node
                names, separated by spaces.
      -h        Writes the usage statement to standard out.
      [-n] node_list
                Specifies  a list of node hostnames, IP addresses, or node
                ranges on which to run the command. (See the noderange man
                page for information on node ranges.)
      -M number of maximum children
                Set the number of nodes to update concurrently.
                (The default is 32.)
      -N Node_group[,Node_group...]
                Specifies one or more node groups on which to run the command.
      -q        Queries for out of date CFM files across the cluster.
      -s        Reports which nodes are up to date by comparing last CFM
                update times. Must be called with the -q option.
      -r remote shell path.
                Path to remote shell. (The default is the DSH_REMOTE_CMD
                environment variable, or /usr/bin/rsh).
      -t timeout
                Set the timeout period (in seconds) for waiting for response
                from a remote process. (The default is 900).
      -v | V    Verbose mode.
      -y        Younger mode. Does not update files younger than master copy.
```

> **Note:** CFM can be set up prior to running the **installnode** command, and common files are distributed at install time while installing nodes.

At CSM install time, root's crontab is updated with an entry to run **cfmupdatenode** every day at midnight.This can changed to suit your requirements.

```
#crontab -l |grep cfmupdate
0 0 * * * /opt/csm/bin/cfmupdatenode -a 1>>/var/log/csm/cfmerror.log
2>>/var/log/csm/cfmerror.log
```

Some common features of CFM, along with usage examples, are described here.

- In general, it is important to have a single /etc/hosts file across the management cluster. The CSM database and other commands do hostname resolution either using /etc/hosts or DNS. To keep a single copy of /etc/hosts, symlink /etc/hoststo /cfmroot/etc/hosts:

  ```
  # ln -s /etc/hosts /cfmroot/etc/hosts
  ```

- Run the **cfmupdatenode** command to copy the hosts file to all managed nodes defined in the CSM database:

  ```
  # cfmupdatenode -a
  ```

- If you want to have a file that is different on the management server and all managed nodes, copy or create the file to /cfmroot instead of symlinking it and then distributing it across to nodes. Files in /cfmroot are not distributed to the management server.

  ```
  # copy /etc/file.1 /cfmroot/etc/file.1
  ```

```
#touch /cfmroot/etc/file.1
#cfmupdatenode -a
```

- Files can be distributed to selected nodegroups only, instead of to all managed nodes, by creating the files with a ._groupname extension. To distribute ntp.conf file to, for example, nodegroup group1, create the file with ._group1 and when cfmupdatenode is run next time, it copies the ntp.conf file only to group1 nodes.

  ```
  # cp /etc/ntp.conf /cfmroot/etc/ntp.conf.group1
  # cfmupdatenode -a
  ```

- CFM customizations can be done after distributing files using pre-install and post-install scripts. Create script files with pre- and post- extensions in cfmroot and when cfmupdatenode runs, pre- and post- scripts are run accordingly. Example 5-29 shows running a pre-install script to save the file and distribute the file, and then running the post script to reset permissions.

*Example 5-29 Example of cfmupdatenode*

```
Create pre and post install scripts
root@ms#cat >/cfmroot/etc/ntp.conf.pre
#!/bin/sh
cp /etc/ntp.conf /etc/ntp.conf.‘date‘
^D
root@ms# cat >/cfmroot/etc/ntp.conf.post
#!/bin/sh
/sbin/service ntprestart
^D
root@ms#chmod 755 /cfmroot/etc/ntp.conf.pre ntp.conf.post
root@ms#cp /etc/ntp.conf /cfmroot/etc/ntp.conf._group1
root@ms#cfmupdatenode -a
```

- To have event monitoring monitor CFM file modifications and push the files whenever files are modified, start the condition and responses as below:

  ```
  # startcondresp CFMRootModTimeChanged CFMModResp
  ```

Whenever a file in /cfmroot is modified, the changes are propagated to all managed nodes in the cluster.

> **Note:** Use caution while enabling CFM event monitoring, as it can impact system performance.

### User id management with CFM

CFM can be used to implement centralized user id management in your management domain. User ids and passwords are generated on the management server, stored under /cfmroot, and distributed to nodes as scheduled.

Copy the following files to /cfmroot to set up effective user id management:

- /etc/passwd ----> /cfmroot/etc/password_useridmgmt.group
- /etc/shadow------> /cfmroot/etc/shadow_useridmgmt.group
- /etc/group---------> /cfmroot/etc/group_useridmgmt.group
- /etc/hosts------> /cfmroot/etc/hosts_useridmgmt.group

Be aware that any id and password changes made on the nodes will be lost once centralized user id management is implemented. However, you can force users to change their passwords on the management server instead of on nodes. Set up scripts or tools to centralize user id creation and password change by group on the management server, and disable password command privileges on managed nodes.

CFM distributes files to managed nodes, but never deletes them. If a file needs to be deleted, delete it manually or with a dsh command from the management server. All CFM updates and errors are logged to files /var/log/csm/cfmchange.log and /var/log/csm/cfmerror.log.

For more information, refer to *IBM Cluster Systems Management for Linux: Administration Guide,* SA22-7873.

## 5.3.5 Software maintenance

The CSM Software Maintenance System (SMS) is used to install, query, update and delete Linux RPM packages on the management server and managed nodes. It is performed using the **smsupdatenode** command. Autoupdate open source software is a prerequisite for using SMS.

SMS uses either install mode to install new RPM packages, or update mode to update existing RPM packages on cluster nodes. Preview or test mode only tests the update without actually installing the packages.

The SMS directory structure includes /csminstall/Linux/InstallOSName/InstallOSVersion/InstallOSArchitecture/RPMS ../updates and ../install subdirectories to maintain all SMS RPMs, updates and install packages, respectively. Sample SMS directory structure on SuSE8.1 looks like the following:

- /csminstall/Linux/SLES/8.1/ppc64/RPMS - contains all dependent RPM packages

- /csminstall/Linux/SLES/8.1/ppc64/updates - -contains all RPM package updates
- /csminstall/Linux/SLES/8.1/ppc64/install - contains all new RPM packages that are not installed with OS and need to be installed. All third party vendor software can also be placed in this subdirectory.

Copy the requisite RPM packages in the respective subdirectories from Install or Update CDs.

> **Note:** SMS is only for maintaining RPM packages. OS patch CDs cannot be used for updating OS packages.
>
> Follow these steps to copy the RPM packages from patch CDs to respective subdirectories, and then issue **`smsupdatenode`**:
>
> 1. Mount the Patch CD on /mnt/cdrom.
> 2. `#cd /mnt/cdrom;cp ‘find . -name *.rpm \`
>    `/csminstall/Linux/SLES/8.1/ppc64/updates ‘`
> 3. `# smsupdatenode -v`

Example 5-30 shows usage of smsupdatenode.

*Example 5-30 smsupdatenode usage*

```
Usage:
smsupdatenode   [-h] [-a | -N node_group[,node_group] | --file file ]
                [-v | -V] [-t | --test] [-q | --query [-c | --common]]
                [--noinsdeps] [-r "remote shell path"]
                [-i | --install packagename[,packagename]]
                [-e | --erase {--deps | --nodeps} packagename[,packagename]]
                [-p | --packages packagename[,packagename]] [[-n] node_list]
smsupdatenode   [--path pkg_path] --copy {attr=value... | hostname}
       -a       Run Software Maintenance on all nodes.
       --copy {attr=value... | hostname}
                Copy the distrobution CD-Roms corresponding to the given
                attributes or hostname to the correct /csminstall directory.
                If you give attr=value pairs they must come at the end of the
                command line. The valid attributes are:
                        InstallDistributionName
                        InstallDistributionVersion
                        InstallPkgArchitecture
                If a hostname is given, the distribution CD-ROMs, and
                destination directory, are determined by the nodes
                attributes.
       -e | --erase {--deps | --nodeps} packagename[,packagename]
                Removes the RPM packages specified after either the --deps
                or --nodeps option.
```

```
            --deps
                Removes all packages dependent on the package targeted for
                removal.
            --nodeps
                Only removes this package and leaves the dependent packages
                installed.
       --file filename
                specifies a file that contains a list of nodes names. If the
                file name "-", then the list is read from stdin. The file
                can contain multiple lines and each line can have one or node
                names, separated by spaces.
       -h       Writes the usage statement to standard out.
       [-n] node_list
                Specifies  a list of node hostnames, IP addresses, or node
                ranges on which to run the command. (See the noderange man
                page for information on node ranges.)
       -i | --install packagename[,packagename]
                Installs the given RPM packages.
       -N Node_group[,Node_group...]
                Specifies one or more node groups on which to run the
                command.
       --noinsdeps
                Do not install RPM dependencies.
       -p | --packages packagename[,packagename]
                Only update the given packages. The user does not have to
                give the absolute path. It will be determined by looking under
                directory structure corresponding to the node.
       --path pkg_path
                Specifies one or more directories, separated by colons, that
                contain copies of the distrobution CD-ROMs. The default on a
                Linux system is /mnt/cdrom and the default on an AIX system is
                /dev/cd0. This flag may only be used with the --copy flag.
       -q | --query [-c | --common]
                Query all the RPMs installed on the target machines and report
                the RPMs installed that are not common to every node.
           -c | --common
                Also report the common set of RPMs (installed on every target
                node).
       -r "remote shell path"
                Path to use for remote commands. If this is not set, the
default
                is determined by dsh.
       -t | --test
                Report  what would be done by this command without making any
                changes to the target system(s)
       -v | -V    Verbose mode.
```

SMS writes logs to /var/log/csm/smsupdatenode.log files.

Kernel packages are updated as normal RPM packages using SMS. Once upgraded, kernel cannot be backed out, so use caution while running **smsupdatenode** command with any kernel packages (kernel* prefix).

Also, make sure to run **lilo** to reload the boot loader if you upgrade kernel and wants to load the new kernel.

### 5.3.6 CSM Monitoring

CSM uses Reliable Scalable Cluster Technology Infrastructure (RSCT) for event monitoring. RSCT has been proven to provide highly available and scalable infrastructure in applications such as GPFS and PSSP.

CSM Monitoring uses a condition and response-based system to monitor system resources such as processes, memory, CPU and file systems. A condition can be a quantified value of a monitored resource attribute, and is based on a defined event expression. If an event expression is true, then an event is generated.

File system utilization(/var) is a resource to be monitored, and “condition” can be THE percent utilization on that resource. For example, /var >90% means if the /var file system increases above a 90% threshold value, then the event expression is true and an event is generated. To prevent flooding of generating events, a re-arm expression can be created. In this case, no event will be generated until the re-arm expression value is true.

A response can be one or more actions performed when an event is triggered for a defined condition. Considering the file system resource example, if we define that a response action is to increase the file system by 1 MB if /var reaches above 90% and to notify the system administrator, then after monitoring is started, whenever /var goes above 90%, a response action is performed automatically.

A set of predefined conditions and responses are available at CSM install. See the *IBM Cluster Systems Management for Linux: Administration Guide* SA22-7873, for more information.

**Resource Monitoring and Control (RMC) and Resource Managers (RMs)**

Resource Monitoring and Control (RMC) and Resource Managers (RM) are components of RSCT and are critical for monitoring.

- RMC provides monitoring of system resources. The RMC daemon monitors resources and alerts RM.

- RM can be defined as a process that maps resources and resource classes to commands for one or more resources. Resource classes contain resource attributes and descriptions and are available for query through the command line.

Table 5-3 lists available resource managers and their functions.

*Table 5-3 Resource managers*

| Resource manager | Function |
|---|---|
| IBM.AuditRM | Audit Logging |
| IBM.ERRM | Event resource manager |
| IBM.HWCTRLRM | Hardware control |
| IBM.DMSRM | Domain node management |
| IBM.HostRM | Hostname management |
| IBM.SensorRM | |
| IBM.FSRM | File System management |

Table 5-4 lists predefined resource classes and can be obtained with the command `lsrsrc.`

*Table 5-4 Predefined resource classes*

| Resource class | Description of attribute |
|---|---|
| IBM.Asociation | Persistent Resources |
| IBM.Auditlog | Event Audit logging |
| IBM.AuditlogTemplate | Template for audit logging |
| IBM.Condition | Pre-defined conditions |
| IBM.EthernetDevice | Primary Ethernet device |
| IBM.EventResponse | Pre-defined responses |
| IBM.Host | Management server host |
| IBM.FileSystem | File system attributes |
| IBM.Program | |
| IBM.TokenRing | Token ring device |
| IBM.Sensor | CFM root and MinManaged |

| Resource class | Description of attribute |
|---|---|
| IBM.ManagedNode | Managed node |
| IBM.ManagementServer | Management server on nodes |
| IBM.NodeAuthenticate | Node authentication |
| IBM.PreManagedNode | PreManaged node classification |
| IBM.NodeGroup | Nodegroups |
| IBM.NetworkINterface | Defined network interface |
| IBM.DmsCtrl | Domain control |
| IBM.NodeHwCtrl | Node Hardware control point attributes |
| IBM.HwCtrlPoint | Hardware Control point (HMC) |
| IBM.HostPublic | |

`lsrsrc` -l Resource_class will list detailed attributes of each resource class. Check the man page of `lsrsrc` for more details.

## Customizing event monitoring

As explained, custom conditions and responses can be created and custom monitoring can be activated on one or more nodes as follows:

- Create a custom condition or event expression such as monitor file system space used on node lpar1 only in the management domain:

```
#mkcondition -r IBM.FileSystem -e “PercentTotUsed > 90” -E “PercentTotUsed
< 85” -n lpar1 -m d “File system space used”
```

  wherein:

  -r option is for resource class

  -e is for creating an event expression

  -E is for re-arm expression

  -n is for specifying a node

  -m is for management domain. Is a must if -n is used.

  d is for description of condition

- Create a custom response, such as e-mail root using a notification script to run Sunday, Monday and Tuesday.

```
#mkresponse -n “E-mail root” -d “ 1+2+3 -s “/usr/sbin/rsct/bin/notifyevent
root” -e b “E-mail root any time”
```

wherein,

-n is for action

-d days of week

-e type of event to run and b is for both event and re-arm event

- Link the created file system space used condition and e-mail notification response

  ```
  #mkcondresp -”File system space used” E-mail root any time”
  ```

- Start the created condition and responses linked above:

  ```
  #startcondresp “File system space used” “E-mail root any time”
  ```

- List the condition and responses to check the status. State is listed as “Active” if started and “Not Active” if not started.

  ```
  #lscondresp
  ```

Example 5-31 shows the output of lscondresp.

*Example 5-31 lscondresp output*

```
#lscondresp
Displaying condition with response information:
Condition                       Response                          Node          State
"NodeFullInstallComplete"       "RunCFMToNode"                    "mgmt_server" "Active"
"NodeManaged"                   "GatherSSHHostKeys"               "mgmt_server" "Active"
"UpdatenodeFailedStatusChange"  "UpdatenodeFailedStatusResponse"  "mgmt_server" "Active"
"NodeChanged"                   "rconsoleUpdateResponse"          "mgmt_server" "Active"
"NodeFullInstallComplete"       "removeArpEntries"                "mgmt_server" "Active"
"FileSystem Space Used"         "E-mail root any time"           "mgmt_server" "Active"
```

If any file system size exceeds 90% on lpar1, our newly created event triggers an an event as a response action by e-mailing root. Monitoring resumes once the file system is fixed back to 85%.

Multiple response actions can be defined to a single condition, and a single response can be assigned to multiple conditions. For Example 5-31, an action such as increasing the file system or deleting files older than 60 days from the file system to claim space could be other actions.

## 5.3.7 Diagnostic probes

CSM diagnostic probes help you diagnose system problems using programs called probes.The CSM command **probemgr** is useful in sending custom probes to determine problems; users write their own diagnostics scripts and call probemgr.

All predefined probes are located in the /opt/csm/diagnostics/probes directory, and probemgr can access the user-defined directory before reading the predefined probes called with a **-D** option. System problem diagnostics can be dependent on each other, and probes take a defined order while running. Example 5-32 shows usage of probemgr.

*Example 5-32 Probemgr usage*

```
probemgr [-dh] [-c {0|10|20|127}] [-l {0|1|2|3|4}]
         [-e prb,prb,...] [-D dir] [-n prb]
    -h        Display usage information
    -d        Show the probes dependencies and the run order
    -c        Highest level of exit code returned by a probe that the
              probe manager permits before terminating.The defaule value
              is 10
                0   - Success
                10  - Success with Attention Messages
                20  - Failure
                127 - Internal Error
    -l        Indicates the message output level. The default is 3
                0 - Show probe manager messages, probe trace messages,
                    probe explanation and suggested action messages, probe
                    attention messages and probe error messages
                1 - Show probe trace messages, probe explanation and
                    suggested action messages, probe attention messages
                    and probe error messages
                2 - Show probe explanation and suggested action messages,
                    probe attention messages and probe error messages
                3 - Show probe attention messages and probe error
                    messages
                4 - Show probe error messages only
    -e prb,.. List of probes to exclude when creating the probe dependency
              tree. This also means that those probes will not be run
    -D dir    Directory where user specified probes reside
    -n prb    Run the specified probe
```

Table 5-5 lists the default pre-defined probes available and the probe dependencies.

*Table 5-5 Probes and dependencies*

| Probe | Dependent probe |
|---|---|
| dsh | ssh-protocol |
| nfs | network |
| rmc | network |

| Probe | Dependent probe |
|---|---|
| errm | rmc |
| fs-mounts | none |
| network-ifaces | network-enabled |
| dmsrm | rmc |
| network-routes | network-enabled<br>network-ifaces |
| network-hostname | none |
| network-enabled | none |
| network-ping | network-enabled<br>network-ifaces<br>network-routes |
| network | network-enabled<br>network-hostnamr<br>network-ifaces<br>network-routes<br>network-ipforward<br>network-ping |
| network-ipforward | none |
| ssh-protocol | none |
| rsh-protocol | none |

All probes are run from the management server using the `probemgr` command. For detailed information on each probe, refer to probemgr man page.

## 5.3.8 Querying the CSM database

CSM stores all cluster information, such as nodes, attributes of nodes, and so on, in a database at a centralized location in the /var/ct directory. This database is accessed and modified using tools and commands, but not directly with a text editor.

Table 5-6 on page 268 lists commands you can use to access the CSM database.

*Table 5-6 CSM database commands*

| **Command** | **Purpose** |
|---|---|
| lsnode | Lists nodes defined |
| definenode | Add/define nodes |
| chnode | Change node definitions |
| rmnode | Remove defined nodes |
| smsupdatenode | Software update the node |
| installnode | Install node |
| csmsetupyast | Setup config for the node to be installed |
| cfmupdatenode | Distribute files |
| rpower | Remote power |
| rconsole | Remote Console |

## 5.3.9 Un-installing CSM

CSM is un-installed by using the **`uninstallms`** command on the management server. Not all packages are removed while running uninstallms. Table 5-7 identifies what is removed and what is not removed with uninstallms.

*Table 5-7 Uninstallms features*

| | |
|---|---|
| Node definitions | Removed |
| Node group definitions | Removed |
| Predefined conditions | Removed |
| CSM packages | Removed |
| CSM log files | Removed |
| CSM Package prerequisites | Not removed |
| RSCT packages when rsct.basic is present | Not removed |
| RSCT packages when csm.client is present on mgmt. server | Not removed |
| RSCT packages when no rsct.basic installed | Removed |
| /cfmroot | Not removed |
| /csminstall | Not removed |

| | |
|---|---|
| /opt/csm | Removed |
| SSH public keys | Not removed |

Clean up manually to remove all the packages and directories that are not removed with the uninstallms command to completely erase CSM. Refer to *IBM Cluster Systems Management Guide for Linux: Planning and Installation Guide-Version 1.3.2,* SA22 7853, for detailed information

## 5.3.10 Distributed Command Execution Manager (DCEM)

DCEM is a Cluster Systems Management GUI interface used to run a variety of tasks on networked computers. Currently this is not available for pSeries machines.

## 5.3.11 Backing up CSM

Currently CSM backup and restore features are not available for pSeries Linux management server version 1.3.2. These will be available in the near future.

## 5.3.12 CSM problem determination and diagnostics

CSM logs detailed information in various log files on the management server and on managed nodes. These log files are useful in troubleshooting problems. In this section, we discuss some common and frequent problems which may be encountered while setting up and running CSM. For more detailed information and diagnostics, refer to the *IBM Cluster Systems Management Guide for Linux: Administration Guide*, SA22-7873.

Table 5-8 lists common CSM problems and their fixes.

*Table 5-8 Common CSM problems and fixes*

| Problem | Fix |
|---|---|
| installms fails | Make sure to copy requisites to the reqs directory on the temporary CSM package folder |
| rpower reports <unknown> status for query | Management server event subscriptions either expired or hung on HMC. Refresh openCIMOM on HMC or reboot HMC. |

| Problem | Fix |
|---|---|
| rpower -a query reports “Hardware Control Socket Error” | ► The Java path might have changed on the management server. Verify that the Java search path is “/usr/lib/IBMJava2-1.3.1/jre/bin/java”.<br>Restart HWCTRL as follows:<br>– stopsrc -s IBM.HWCTRLRM;<br>– startsrc -s IBM.HWCTRLRM -e “HC_JAVA_PATH=/usr/lib/IBMJava2-1.3.1” |
| Any other rpower or rconsole errors | ► Stop/start RMC daemons such as IBM.HWCTRLRM on the management server<br>► Stop/Start openCIMOM on the HMC by running “initCIMOM stop” and “initCIMOMstart”<br>► Reboot the HMC<br>► Start IBM.HWCTRLRM with trace hooks on to collect more data by running “startsrc -s IBM.HWCTRLRM -e “HC_JAVA_VERBOSE=/tmp/jni.txt”<br>► Run “rmcctl -A” and rmcctl -Z” to stop/remove and add/start RMC daemons<br>► Check *IBM Cluster Systems Mgmt for Linux: Hardware Control Guide*, SA22-7856 |
| rconsole not coming up on the current window | Check the flags to make sure the `-t` option is specified |
| rconsole fails with “xinit failed to connect to console” error | ► Corrupted conserver.cf. Rewrite the config files and refresh conserver conserver<br>► chrconsolecfg -a<br>► rconsolerefresh -r |

| Problem | Fix |
|---|---|
| csmsetupyast fails with getadapters errors | ► Run getadapters command line to populate CSM node database<br>► Or use the chnode command to upload node network attributes such as InstallAdapterType, MAC address, and so on<br>► getadapaters may fail if run on multiple nodes, so check /var/log/csm/getadapters/getadapter*.* logs to check, and fix any errors with locks |
| installnode fails | ► Run lsnode and verify duplicate nodes entries are listed<br>► Verify log files to find/fix any errors<br>► Check for node attribute “Mode” to make sure it is set to PreManaged. If set to Installing and installnode fails, reset it to “PreManaged” with the chnode command<br>► Check network cables for proper connectivity<br>► Check /etc/dhcpd.conf and restart dhcpd<br>► Check /etc/inetd.conf for tftp and restart inetd<br>► Open the read only console and look for any packaging errors. Installnode waits for input if any package errors are encountered. Open the read-write console to interactively respond to install options and close the console. |
| updatenode fails | Check for hostname resolution errors |
| dsh fails | Check for SSH authentication errors |
| smsupdatenode fails | Check that RPM packages are copied to the right directories |
| Event monitoring fails | Check for proper network connectivity |

Refer to *IBM Cluster Systems Management for Linux: Hardware Control guide* SA22-7856 for more information on hardware control and HMC connectivity and RMC issues.

## 5.4 CSM hostname changes

Changing hostnames and IP addresses of the management server and management node are inevitable in a large management domain. However, when this happens, common features of CSM fail to function if the CSM database is not updated properly. In this section, we explain how to perform these updates.

### Changing the management server hostname

If the management servers hostname is changed, follow these steps to update CSM:

- Update DNS or the /etc/hosts file on the management server to reflect the new hostname.
- Run the **chnode** command and reset the ManagementServer node attribute to the new hostname on all nodes.

  #chnode -a ManagementServer=newHostname

- Run the **updatenode** command to re-exchange keys between the management server and the managed node.

  ```
  #updatenode -a -k
  ```

- Run the **cfmupdatenode** command to push the /etc/hosts file if a common hosts file is used.

### Changing the managed node hostname

If any of the managed node's hostnames are changed, follow these steps to update CSM:

- Update DNS or the /etc/hosts file to reflect the new node hostname.
- Run the **chnode** command to reset the HostName node attribute.

  ```
  #chnode oldHostName HostName=newHostName
  ```

- Run the **updatenode** command to exchange keys with the node.

  ```
  # updatenode -k newHostName
  ```

- Update any custom scripts or event monitoring definitions to reflect the new hostname.

## 5.5 CSM interoperability

This section describes Cluster Systems Management offerings from IBM and IBM product positioning of CSM with respect to the Parallel Systems Support Program (PSSP) and CSM on IBM xSeries running Linux and pSeries running AIX operating systems.

Specifically, we discuss the following products for interoperability, common features, and when to use each:

- CSM for pSeries AIX
- CSM for pSeries Linux
- CSM for e325 and xSeries Linux
- Parallel Systems Support Program PSSP-3.5
- Matrix of CSM products

## 5.5.1 CSM for pSeries AIX

Cluster Systems Management for AIX primarily provides management server functionality to a mixture of managed nodes running on different hardware platforms such as pSeries AIX, pSeries Linux, and xSeries Linux, in addition to serving other cluster functions.

CSM for AIX modular architecture allows independent use of features and functions on each set of managed nodes. It is easy to set up and manage a mixture of clustered nodes. It is mandatory to have a pSeries server running AIX 5.2 to be a management server to support the environment. The following hardware and software is supported as nodes.

**Hardware**

- pSeries (p615, p630, p650, p655, p690) in standalone or LPAR mode. Cluster 1600. HMC connection is a must for hardware control
- Any RS6000 without HMC connection with no Hardware Control support
- xSeries (x330, x335, x342, x345, x360)

**Software**

- AIX 5.2, AIX 5.1-ML03 on pSeries
- SLES 8 on pSeries
- RHEL AS 2.1, 3.0 and SLES8 on xSeries

Refer to *IBM Redbook: An Introduction to CSM 1.3 for AIX 5L*, SG24-6859, for detailed information about CSM on the AIX platform.

## 5.5.2 CSM for pSeries Linux

Cluster Systems management for Linux on pSeries hardware only supports Linux-managed nodes and cannot support heterogeneous managed nodes. So the management server and managed nodes all must be running Linux in the cluster.

The following hardware and software requirements outline the requisites:

**Hardware**

- pSeries (p615, p630, p650, p655, p690) in standalone or LPAR mode. Cluster 1600. HMC connection is a must for hardware control
- Any RS6000 without HMC connection with no Hardware Control support

**Software**

- SuSE SLES 8

## 5.5.3 CSM for xSeries Linux and e325 (Cluster 1350)

Cluster Systems management for cluster 1350 hardware provides a management environment for cluster 1350 hardware running Intel or AMD Opteron processor-based Linux nodes.

The management server must be xSeries, IntelliStattion or Blade Center hardware; managed nodes can be mixed on hardware, but the Linux operating system cannot be mixed and has to be only one type on all managed nodes.

CSM is integrated with IBM Director software for interoperability and common management functions. The following hardware and software is supported:

**Hardware**

- IBM eServer xSeries (335, 345, 360)
- IBM eServer325(AMD Opteron)
- IBM eServer Blade Center

**Software**

- RHEL AS 2.1 (not on e325)
- RHEL AS 3.0
- SLES 8
- RH 9

## 5.5.4 Parallel Systems Support Program (PSSP)-3.5

PSSP version 3.5 is the latest and final developed version of software mainly developed and used for IBM SP2® environment. PSSP's rich features, such as Reliable Scalable Clustering Technology (RSCT), were the backbone of Cluster Systems Manager (CSM) software as well.

PSSP 3.5 is a cluster management solution for IBM SP2 Power Parallel clusters running High Speed Switch 2 and is widely in use for high performance computing environments. PSSP 3.5 is supported on AIX 5.x and SP high speed Switch and Switch 2 technologies. Version 3.5 is the announced final development version of PSSP and upcoming new SP switch technologies are supported on CSM software.

Following is a list of PSSP-supported hardware and software.

**Hardware**

- IBM SP nodes
- pSeries Servers
- Any RS6000 supporting AIX 5.x

**Software**

- AIX 5.2
- AIX 5.1

## 5.5.5 Matrix of CSM products

A comparative matrix of CSM products and product positioning is provided outlining major features, limitations, supported environments and when and where to use a specific version. Table 5-9 shows a comparative assessment of CSM product positioning on different hardware and software.

*Table 5-9 CSM product positioning*

| **Feature** | **pSeries AIX** | **pSeries Linux** | **xSeries/e325 Linux** |
|---|---|---|---|
| Management server | Must be pSeries AIX 5.2 | Must be pSeries SLES8 | Must be xSeries, Intellistation or Blade Center |
| Managed nodes | Mix of nodes on different hardware and software. AIX and Linux LPARs can co-exist | All managed nodes must be only SLES8 | Mix of hardware allowed, but only one type Linux such as SLES8, RHEL AS3.0 is required |
| DCEM Admin GUI interface | Available | Not available | Available |
| CSM database Backup and Restore | Available | Not available | Not available |
| Diagnostic probes | All probes are available | All except Hardware control are available | All available |

| Feature | pSeries AIX | pSeries Linux | xSeries/e325 Linux |
|---|---|---|---|
| Software Maintenance | Supported | Supported | Supported |
| Event Monitoring | Supported | Supported | Supported |
| Hardware Control | Only for HMC and Management process or adapter card-connected servers | Only for HMC-connected servers | Only for Management Processor adapter Card-connected servers |
| Dynamic LPAR | Available only for AIX-managed nodes | Not available | Not available |
| CSM Image Cloning | Supported (mksysb-NIM) | Not supported | Supported (SIS and csmsetupsis) |
| csmstat (Status command for nodes) | Available | Not available | Not available |

| Feature | pSeries AIX | pSeries Linux | xSeries/e325 Linux |
|---|---|---|---|
| When/Where best suited | ► Mix of managed nodes running different hardware<br>► Coexistence of operating systems such as AIX, Linux on different LPARs<br>► Centralized management and administration<br>► Latest SP high speed switch environments<br>► Nodes with remote hardware control<br>► High availability cluster can be part of management domain | ► Group of Linux servers running on pSeries hardware<br>► Centralized management and administration<br>► Nodes with remote hardware control feature<br>► Environments where only one flavor of operating systems required<br>► High availability clusters can be part of management domain | ► Mix of managed nodes running different xSeries, IntelliStation®, e325 and Bladecenter hardware<br>► A group of Linux nodes<br>► Centralized management and administration<br>► Nodes with remote hardware control<br>► High availability clusters can be part of management domain |

## 5.6 Future of CSM

Cluster Systems Management for pSeries Linux is a dynamic environment to centralize management of a cluster of pSeries Linux systems; some of the new features which might be available in the near future are:

- Highly Available Management Server to eliminate the management server being a single point of failure. Once set up, managed nodes can function independently, but the management server is important for using features such as event monitoring.
- Interoperability and mix of managed nodes running on xSeries, IntelliStation, e325 Linux clusters and pSeries servers running AIX.
- Red Hat Enterprise Advanced Server version 3.0 support for the management server and nodes.

- A higher number managed nodes. With CSM version 1.3.2, the current limit is a maximum of 128 pSeries Linux nodes. Future versions will support a higher number of nodes.
- Support for new High Speed Switch offerings.
- Better cross-support on xSeries systems using IBM Director software.

# 6

# Getting started with GPFS

In this chapter, we provide a brief introduction to GPFS version 2.2 for Linux on pSeries. GPFS is a cluster file system that provides users with fast, consistent, and reliable shared access to files, cluster-wide. Implemented as the Virtual File System (VFS), GPFS supports the UNIX file interface so that applications need not be changed to take advantage of the new file system. Initially found only on AIX clusters, GPFS is now available across the whole cluster offering from IBM: AIX/pSeries, Linux/pSeries, and Linux/xSeries.

This chapter contains:

- 6.1, “GPFS description” on page 280. This section describes the GPFS architecture and its components. The most striking features of GPFS are presented.
- 6.2, “RSCT peer domain setup” on page 285. GPFS relies on RSCT for configuration, control and operation. We show how to define a RSCT peer domain on which the GPFS cluster is defined. A few RSCT commands are introduced and described here.
- 6.3, “GPFS installation and basic configuration” on page 290. This section describes the installation, configuration, and startup of GPFS on a very simple setup. It is useful to understand the steps you need to take to create and operate a GPFS file system. In real situations, more of the GPFS features that contribute to its unique strength and resilience can be used. But regardless of configuration (number of disks, high availability, data redundancy), the steps and commands are the same.

# 6.1 GPFS description

GPFS is a major component of the IBM clusters offering. Its purpose is to satisfy the needs of the most demanding applications in terms of I/O throughput and resilience. This is obtained by combining the processing power and disk bandwidth of selected nodes and making them available to the whole cluster.

There are other file systems that can be used to provide shared and consistent access to files inside a cluster. Network File System (NFS) is still very much in use. However, if an NFS can be mounted on all the nodes in a cluster, the files are still served by a single node, and this quickly becomes a bottleneck. This single server architecture does not scale well when the number of client nodes increases.

The open source community provides Parallel Virtual File System (PVFS[1]), which uses concepts similar to GPFS. It can use any number of server nodes, each node using its own disks and disk adapters. It is known to work on pSeries Linux clusters, but lacks the high availability features of GPFS.

Here we describe the GPFS architecture and components and how they operate. However, this description is by no means exhaustive and the interested reader should refer to the following documentation for more information:

- *Linux Clustering with CSM and GPFS*, SG24-6601
- The complete GPFS documentation can be viewed at:

  `http://www.ibm.com/servers/eserver/clusters/software/gpfs.html`

## 6.1.1 Principles of operation

GPFS version 2.2 supports two modes of operation, depending on the type of attachment of the storage devices to the nodes: direct-attached disks, and Network Shared Disks. We describe these as follows:

- Direct-attached disks

  In direct attachment mode, all nodes see all the disks allocated to GPFS, generally through a SAN. In this mode, the path for the file data between the GPFS nodes is through the fiber channel attachments. All the nodes participating to the GPFS cluster must have an attachment to the SAN.

- Network Shared Disk

  A subset of the GPFS nodes are designated as the GPFS servers. They “export” their locally attached disks, through a network disk abstraction called Network Shared Disk (NSD), to the other cluster nodes.

[1] http://www.parl.clemson.edu/pvfs/

This time, the file data flows between the GPFS server nodes and the client nodes over a TCP/IP network connection. The constraint here to have a fast network connection between the nodes. The network of choice in this mode of operation is Gigabit Ethernet.

Figure 6-1 illustrates the direct-attached disks mode of operation.

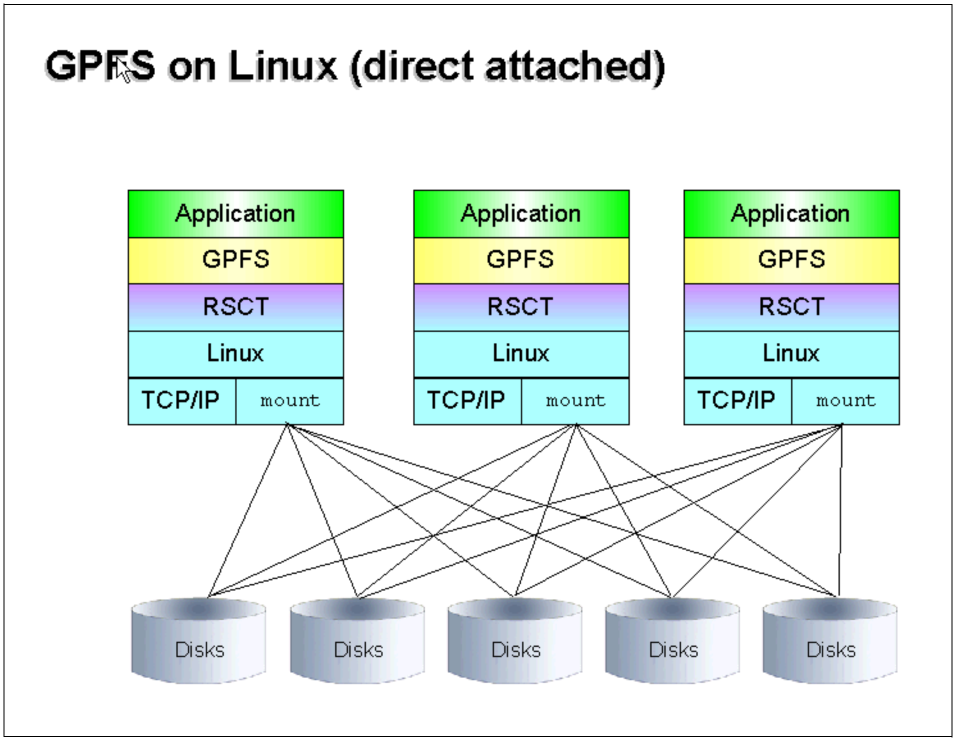


*Figure 6-1 GPFS with direct-attached disks*

The Network Shared Disks mode is shown in Figure 6-2 on page 282.

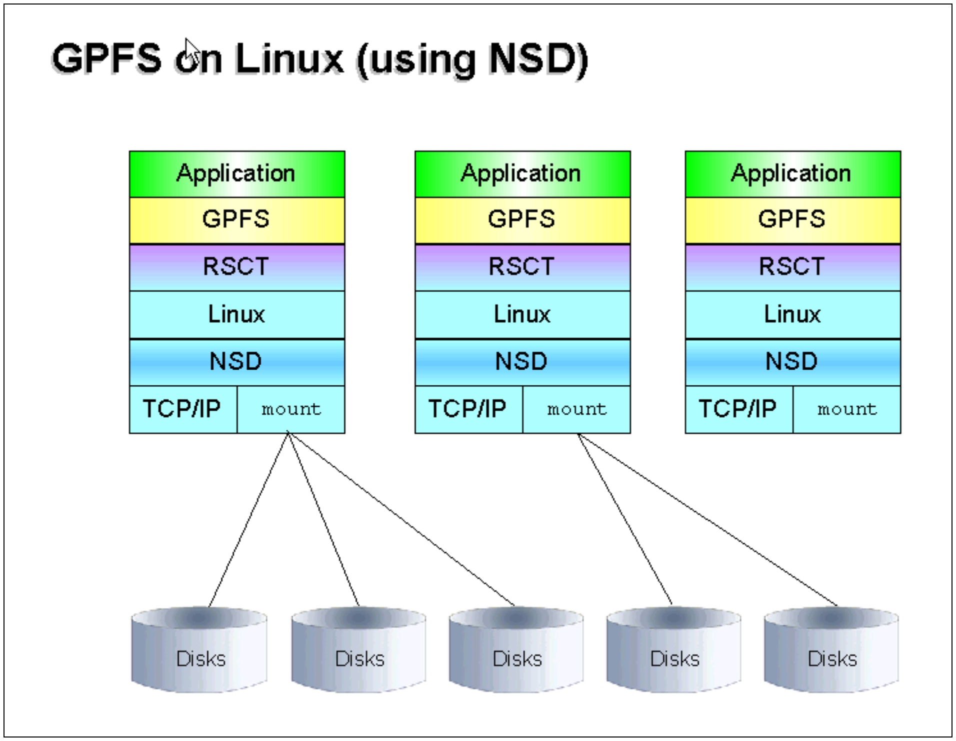


*Figure 6-2 GPFS with Network Shared Disks*

GPFS is implemented at the VFS level so that applications can use it transparently just like any other file system. GPFS also provides an API for applications that would like to benefit from some of the features that are not accessible through the UNIX file system interface.

On AIX pSeries clusters, IBM Parallel Environment 4.1 implements the I/O part of the MPI 2 standard with GPFS, providing a very high performance parallel I/O environment for large parallel applications.

## 6.1.2 Terminology

GPFS defines various entities, as described here:

### GPFS cluster

GPFS cluster identifies the collection of nodes that you intend to run GPFS on. GPFS clusters can be of different types, but the only type supported on Linux is the loose cluster type or “lc”. AIX GPFS clusters can be of other types: hacmp in

HACMP environments, or sp if defined on AIX clusters equipped with SP Switch or SP Switch 2.

With the lc type, GPFS clusters can be interoperable between AIX and Linux on xSeries. Nodes in a GPFS cluster must use the *same* communication adapters. In the case of Linux on pSeries, only Ethernet is supported.

### GPFS nodeset

A *nodeset* is a subset of the GPFS cluster, and this is where the file systems are defined. A node can only belong to one nodeset at a time. It is really where all the configuration takes place.

Examples of attributes that are private to a GPFS nodeset are quorums, and the maximum size of the caching attributes of the file systems, and the communication protocol to use between the participating nodes.

### Failure group

A *failure group* identifies the set of disks that have a common point of failure. For example, two disks on the same adapter on the same node are part of the same failure group.

This information is used by GPFS to decide on data placement in order to improve data availability. Indeed, there is no point to having two replicas on the same data in the same failure group, as they would both become unavailable in case of a failure.

## 6.1.3 Components

GPFS consists of the following components:

- Administrative commands

  These reside under /usr/lpp/mmfs/bin. You should update your PATH to include this directory.

- The GPFS daemon( mmfsd)

  This is a multi-threaded daemon that runs on all the nodes in the nodeset. It is responsible for the following:

  - Performing all I/O operations on files
  - Managing directories
  - Maintaining data integrity
  - Satisfying disk space allocation requests

- The kernel module (mmfs)

This is used for interfacing with the VFS layer. System calls to access files in a GPFS file system will be handed by the Linux kernel to this module.

- The portability layer module (mmfslinux)

  This open source module has to be compiled during GPFS installation. It provides a portable interface between the kernel and the rest of GPFS, which is provided as binaries only.

- tracedev

  This module provides a trace facility for GPFS. Under normal circumstances, tracing is disabled, and it only needs to be enables in the course of troubleshooting under the guidance of IBM support personnel.

  However, the tracedev module will be loaded every time at GPFS startup. It also ships as source code and needs to be built together with mmfslinux (the same set of makefiles build both).

### 6.1.4 Advantages

As a general purpose file system, GPFS has several unparalleled features that we detail now.

#### *I/O performance*

By aggregating the individual I/O performance and processing power of many cluster nodes, GPFS can dramatically improve the disk throughput of serial and parallel applications. A sound design should make sure that you match the performance characteristics of all the components (nodes, disks, network or fiber adapters) to avoid bottlenecks. GPFS is at its best for large sequential I/O patterns, and it is the file system of choice for distributed applications that need to operate concurrent read and write operations on files.

#### *High availability*

When properly configured, GPFS provides a highly resilient file system. It can recover from failures at the disk, adapter, or node level. Node failure recovery is facilitated with three components of RSCT:

- System Resource Controller (SRC ) to start, stop, and interact with RSCT daemons
- Topology Services, which constantly monitors the status of nodes and network adapters in the cluster
- Group Services, which provides coordination between processes executing on distributed nodes by maintaining group-wide information and informing members of the group of events affecting the group.

With data replication enabled, not only can the structure and operation of the file system be continuous, but the data can also be made highly available.

### *Flexibility*

Disks and nodes can be added on the fly to a GPFS nodeset. When it becomes more convenient from an operational point of view, the data can then be rebalanced between the various disks and nodes.

## 6.2 RSCT peer domain setup

A GPFS cluster requires an underlying RSCT peer domain. In this section, we describe the commands we used to create an RSCT peer domain on our experimental setup.

There are currently two types of domains in RSCT:

- A management domain, with one management server and managed nodes. This is typically a CSM domain. The RSCT communication in this model flows between the management server and the nodes; it never flows between the nodes themselves.
- A peer domain, where all nodes have the same role as far as RSCT is concerned. The information about the domain is available to all participating nodes through the use of Topology Services and Group Services, which are started upon creation of the peer domain.

Note that we do not cover every aspect of peer domains here. For a complete description of RSCT, refer to the RSCT documentation:

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

Chapter 2 of the Administration Guide contains information about creating and administering peer domains. Section 4.3 in the IBM Redbook *A Practical Guide for Resource Monitoring and Control (RMC),* SG24-6615, contains an example of a peer domain setup.

Our experimental setup for GPFS, and the underlying RSCT peer domain, consists of two nodes connected by a 10/100 Mbit Ethernet and a Gigabit Ethernet. One node will be primarily for control messages, and the second node will be used for GPFS communication.

Table 6-1 summarizes the names and interfaces of our nodes.

*Table 6-1 Network interfaces and corresponding names*

| Node | eth0: 10/100 Mbit | eth1: 1000 Mbit |
|---|---|---|
| node1 | r01n33 | gr01n33 |
| node2 | r01n34 | gr01n34 |

## 6.2.1 RSCT requisites

In our testing, we used RSCT version 2.3.2-1. These filesets must be installed on the nodes prior to defining the peer domains:

- src-1.2.0
- rsct.core.utils-2.3.2
- rsct.basic-2.3.2
- rsct.64bit-2.3.2
- rsct.core-2.3.2

## 6.2.2 Overview of peer domain operations

RSCT provides commands for adding, removing, starting, and stopping nodes inside a peer domain and commands to create, start, stop, and remove peer domains. Commands acting on nodes end in **rpnode**. Commands for domains end in **rpdomain**.

## 6.2.3 Creating a peer domain

In case the GPFS nodes are part of a CSM cluster, you must tell RSCT that the commands need to operate on a peer domain. To do this, issue the command shown in Example 6-1.

*Example 6-1 Setting the RSCT management scope*

```
lpar1:~ # export CT_MANAGEMENT_SCOPE=2
```

### *Prepare the nodes*

Before being incorporated in a peer domain, the security environment has to be prepared for all the nodes. During this, the nodes will exchange public keys. This is done with the **preprpnode** command, as shown in Example 6-2.

As an argument to the **preprpnode** command, a list of all the nodes from which this node can accept RSCT messages from. In a GPFS cluster, this command needs to be run on all the participating nodes.

*Example 6-2 preprpnode command*

```
r01n33:~ # preprpnode -V r01n33 r01n34
Beginning to prepare nodes to add to the peer domain.
Completed preparing nodes to add to the peer domain.
```

### *Create the peer domain*

Once the nodes are prepared we can proceed to creating the domain. This is done with the **mkrpdomain** command. A peer domain is characterized by a name, the names of the constitutive nodes, and the UDP ports that Topology Services and Group Services will use. The command shown in Example 6-3 creates the peer domain named `itso`. We run this command once on one of the nodes.

*Example 6-3 Create a peer domain*

```
r01n33:~ # mkrpdomain -V itso r01n33 r01n34
Making the peer domain "itso".
Completed making the peer domain "itso".
```

The command **lsrpdomain** shows the status of the peer domain we have just created. Example 6-4 shows as “Offline” because we have not yet started it.

*Example 6-4 Check the status of the peer domain*

```
r01n33:~ # lsrpdomain
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
itso Offline 2.3.2.0           No            12347  12348
```

## 6.2.4 Starting and stopping a peer domain

To bring the peer domain online, we use the **startrpdomain** command as shown in Example 6-5 on page 288. This command can be run on any node in the domain, but it must be issued on only *one* node.

Immediately after, we check the domain status with **lsrpdomain**. It can take a while, up to one minute, for the domain to come up. The status should change from “Pending online” to “Online”.

*Example 6-5 Start the peer domain and watch it come online*

```
r01n33:~ # startrpdomain itso
r01n33:~ # lsrpdomain
Name OpState        RSCTActiveVersion MixedVersions TSPort GSPort
itso Pending online 2.3.2.0           No            12347  12348

After a while ...

r01n33:~ # lsrpdomain
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
itso Online  2.3.2.0           No            12347  12348
```

Upon startup of the peer domain, the Topology Services and Group Services subsystems will be started on the nodes. To check the proper operation of these services, use the **lssrc** command as shown in Example 6-6.

The **lssrc** command is part of System Resource Controller (SRC), which has been ported from AIX to Linux. It is a way to control subsystems. Subsystems must register to SRC in order to be controlled.

*Example 6-6 Use lssrc to query Topology/Group Services*

```
r01n33:~ # lssrc -a
Subsystem         Group            PID     Status
   IBM.ConfigRM     rsct_rm          781     active
   ctcas            rsct             4376    active
   cthats           cthats           10839   active
   cthags           cthags           10843   active
   ctrmc            rsct             10933   active
   IBM.ERRM         rsct_rm          10947   active
   IBM.AuditRM      rsct_rm          10968   active
   IBM.HostRM       rsct_rm                  inoperative
```

The status of individual nodes can be displayed with the **lsrpnode** command, as shown in Example 6-7.

*Example 6-7 lsrpnode to check each node*

```
r01n33:~ # lsrpnode
Name    OpState RSCTVersion
gr01n34 Online  2.3.2.0
gr01n33 Online  2.3.2.0
```

To stop a peer domain, the nodes can be stopped individually with the **stoprpnode** command as shown in Example 6-8. We use the **lsrpnode** to verify that the node was stopped correctly.

*Example 6-8 Stopping a node in the peer domain*

```
r01n33:~ # stoprpnode gr01n34
r01n33:~ # lsrpnode
Name    OpState RSCTVersion
gr01n34 Offline 2.3.2.0
gr01n33 Online  2.3.2.0
```

We can use the **stoprpdomain** command to stop all nodes at once. This is shown in Example 6-9.

*Example 6-9 Stopping the whole peer domain*

```
r01n33:~ # lsrpnode
Name    OpState RSCTVersion
gr01n34 Online  2.3.2.0
gr01n33 Online  2.3.2.0
r01n33:~ # stoprpdomain itso
r01n33:~ # lsrpnode
lsrpnode: There are no nodes in the peer domain or an online peer domain does
not exist.
```

### 6.2.5 Removing a peer domain

To remove the peer domain, you must bring it online first and then use the **rmrpdomain** command; an example is shown in Example 6-10.

For some unknown reason, the peer domain may survive. To remove it permanently, use the **-f** (force option).

*Example 6-10 Removing a peer domain*

```
r01n33:~ # rmrpdomain itso
rmrpdomain: There are no nodes in the peer domain or an online peer domain does
not exist.
r01n33:~ # rmrpdomain -f itso
```

# 6.3 GPFS installation and basic configuration

Here we describe the steps we took to install GPFS on our cluster. We used the NSD model, using one internal disk per server node. Both nodes act as client and server. Although our cluster was not a realistic one, bringing up a real production GPFS system would follow the same path.

## 6.3.1 Planning

GPFS is becoming more and more dynamic in the way it handles its components. You can add or remove disks and nodes and change file system settings without stopping GPFS. However, careful planning is still a very important step.

You need to consider the following areas:

- Hardware

  Verify that the hardware (especially storage devices) you are using is supported by GPFS.

  Refer to the FAQ page for more information:

  `http://publib.boulder.ibm.com/clresctr/library/gpfs_faq.html`

- Software

  Ensure that you have the right levels of GPFS at hand. GPFS 2.2 for pLinux requires SLES8 Service Pack 3 or later. (RHAS 3 is not yet supported for GPFS.)

- Performance

  This is the time to design your GPFS setup to satisfy the planned file system performance. The following areas should be examined carefully:

  - The number, capacity and speed of the disks
  - The number of disk adapters
  - The number and capacity of server nodes
  - The number and types of network adapters and switches, if you plan to use the Network Shared Disk (NSD) model

  An understanding of future file access patterns, as well as of the file size distribution in the file system, is also invaluable.

  GPFS stripes file data across the disks and uses large block size to maximize I/O efficiency. If you can change the block size of your application, make it a multiple of the GPFS block size.

- Resilience

  You must define the level of resilience that you need to satisfy your operational requirements, and then proceed accordingly. Carefully review the failure groups and the number of replicas that you need to configure in order to satisfy your high availability requirements.

### 6.3.2 Software installation

The following RPMs must be installed:

- gpfs.gpl-2.2.0
- gpfs.msg.en_US-2.2.0
- gpfs.docs-2.2.0
- gpfs.base-2.2.0

The kernel source, development tools, and cross-compilers need to be installed in order to be able to compile the portability layer.The **imake** command, which is part of the xdevel package, is also needed.

If all your nodes are identical, you may install these development tools on one node only and then copy the binaries to the other nodes; this technique is described in 6.3.3, “Compiling the portability layer” on page 291.

### 6.3.3 Compiling the portability layer

An explanation of how to compile the portability layer is detailed in a README file located under /usr/lpp/mmfs/src. You can build as a regular user (which requires this user to have write permission in the /usr/lpp/mmfs/src directory and read permission for all files under the /usr/src/linux/ directory).

All the action takes place in the /usr/src/mmfs/src/config directory. The first step in compiling the portability layer is to set the environment, as shown in Example 6-11.

*Example 6-11 Compilation of the portability layer*

```
r01n33:/usr/lpp/mmfs/src/config # export SHARKCLONEROOT=/usr/lpp/mmfs/src
r01n33:/usr/lpp/mmfs/src/config # cp site.mcr.proto site.mcr
```

Use the site.mcr.proto file as a template. The real file is site.mcr, and it needs to be edited to suit your needs; this is well documented inside the file.

Example 6-12 on page 292 shows the differences between the template file and a file that worked for us.

*Example 6-12 diff site.mcr site.mcr.proto*

```
r01n33:/usr/lpp/mmfs/src/config # diff site.mcr.proto site.mcr
13c13
< #define GPFS_ARCH_I386
---
> /* #define GPFS_ARCH_I386 */
15c15
< /* #define GPFS_ARCH_PPC64 */
---
> #define GPFS_ARCH_PPC64
34c34
< LINUX_DISTRIBUTION = REDHAT_LINUX
---
> /* LINUX_DISTRIBUTION = REDHAT_LINUX */
36c36
< /* LINUX_DISTRIBUTION = SUSE_LINUX */
---
> LINUX_DISTRIBUTION = SUSE_LINUX
55c55
< #define LINUX_KERNEL_VERSION 2041900
---
> #define LINUX_KERNEL_VERSION 2042183
```

The differences are quite easy to see. The architecture needs to be changed to PPC64, and the name of the distribution and the Linux kernel version need to be changed, too.

To determine which kernel version you are running, use the command shown in Example 6-13; note that the version number is highlighted.

*Example 6-13 Get to know your kernel version*

```
r01n33:/usr/lpp/mmfs/src/config # cat /proc/version
Linux version 2.4.21-83-pseries64 (root@PowerPC64-pSeries.suse.de) (gcc version
3.2.2) #1 SMP Tue Sep 30 11:30:48 UTC 2003
```

Next, you need to configure the installed kernel source tree. The source tree is not configured properly when the kernel-source RPM is installed—this is true for both SLES8 and RHAS 3 distributions.

Several commands are needed to rectify the situation; these commands are shown in Example 6-14 on page 293 for SLES8 (RHAS 3 is not supported at the time of writing).

Adjust the names of the files in the /boot directory to your environment, if needed. Root authority is required.

*Example 6-14 Configure the kernel source tree under SLES8*

```
r01n33:/usr/lpp/mmfs/src/config # cd /usr/src/linux-2.4.21-83
r01n33:/usr/src/linux-2.4.21-83 # sh make_ppc64.sh distclean
r01n33:/usr/src/linux-2.4.21-83 # cp /boot/vmlinuz-2.4.21.config .config
r01n33:/usr/src/linux-2.4.21-83 # sh make_ppc64.sh oldconfig
$(/bin/pwd)/include/linux/version.h update-modverfile
```

Once this is done, move back to the /usr/lpp/mmfs/src directory to build and install the portability layer, as shown in Example 6-15.

*Example 6-15 Build and install the portability layer*

```
r01n33:/usr/lpp/mmfs/src # make World
...
Checking Destination Directories....
\c
\c
\c
touch install.he
\c
\c
\c
touch install.ti
make[1]: Leaving directory `/usr/lpp/mmfs/src/misc'
r01n33:/usr/lpp/mmfs/src # make InstallImages
cd gpl-linux; /usr/bin/make InstallImages;
make[1]: Entering directory `/usr/lpp/mmfs/src/gpl-linux'
mmfslinux
mmfs25
lxtrace
dumpconv
tracedev
make[1]: Leaving directory `/usr/lpp/mmfs/src/gpl-linux'
```

The five files highlighted in Example 6-15 are copied into the /usr/lpp/mmfs/bin directory. If all the nodes in your cluster are identical and run the same Linux kernel, you can simply copy over these files to the /usr/lpp/mmfs/bin directory on all nodes.

## 6.3.4 Configuring and bringing up GPFS

This proceeds in seven steps, for NSD-based configurations:

1. Create the GPFS cluster.
2. Create a GPFS nodeset inside this cluster.
3. Start up the GPFS nodeset.
4. Create the local disk partitions.

5. Create the Network Shared Disks.
6. Create a GPFS file system.
7. Mount the GPFS file system.

### *Create the GPFS cluster*

We defined the GPFS cluster on top of the RSCT peer domain created during 6.2, “RSCT peer domain setup” on page 285. First, you need to create the file that describes the nodes that will be part of the GPFS cluster. Its contents are shown in Example 6-16. The syntax is one node per line.

*Example 6-16 Example of a GPFS nodes file*

```
r01n33:~ # cat /tmp/gpfsnodes
gr01n33
gr01n34
```

To create the GPFS cluster, use the **mmcrcluster** command as shown in Example 6-17 on page 294. You do not need to specify the RSCT peer domain; the GPFS cluster will be defined on the current peer domain, which must be online when you issue the command.

Note that, in the example, we specified the use of **ssh** and **scp**. In order for the command to succeed, root must be able to execute **ssh** from all nodes, to all nodes, using all interfaces, without being prompted for a password. Refer to 3.12, “ssh” on page 144 for a discussion on how to set up **ssh**.

You have to designate a primary and a backup server. What is meant here by “primary” and “secondary” servers are nodes that hold the *cluster data*, not the subsequent file data that will indeed be spread over all the nodes in the nodesets that will be defined later.

Note also the use of the lc type for the cluster. lc (loose clusters) is the only type supported for Linux on pSeries.

In Example 6-17, we also show the use of the **mmlscluster** command to list the properties of the newly created cluster. **mmlscluster** is correct in reporting that the two nodes do not belong to any nodeset.

*Example 6-17 mmcrcluster command*

```
r01n33:~ # mmcrcluster -t lc -p gr01n33 -s gr01n34 -r /usr/bin/ssh -R
/usr/bin/scp -n /tmp/gpfsnodes 2>&1
Thu Oct 30 15:12:35 PST 2003: mmcrcluster: Processing node gr01n33
Thu Oct 30 15:12:36 PST 2003: mmcrcluster: Processing node gr01n34
mmcrcluster: Command successfully completed
mmcrcluster: Propagating the changes to all affected nodes.
This is an asynchronous process.
```

```
r01n33:~ # mmlscluster

GPFS cluster information
========================
  GPFS cluster type:         lc
  GPFS cluster id:           gpfs1067555555
  RSCT peer domain name:     itso
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp

GPFS cluster data repository servers:
-------------------------------------
  Primary server:    gr01n33
  Secondary server:  gr01n34

Cluster nodes that are not assigned to a nodeset:
-------------------------------------------------
   1   (1)   gr01n33     10.10.10.33      gr01n33
   2   (2)   gr01n34     10.10.10.34      gr01n34
```

### *Create a GPFS nodeset*

To create a GPFS nodeset, you need a simple file that lists, one line per node, the name of the nodes to include in the nodeset. This has to be a subset of the nodes composing the cluster. To create the nodeset, use the **mmconfig** command as shown in Example 6-18 on page 295. Use **mmlsconfig** to display the properties of the nodeset.

*Example 6-18 Creating the GPFS nodeset*

```
r01n33:~ # cat /tmp/gpfsnodeset
gr01n33
gr01n34
r01n33:~ # mmconfig -n /tmp/gpfsnodeset
mmconfig: Command successfully completed
mmconfig: Propagating the changes to all affected nodes.
This is an asynchronous process.

r01n33:~ # mmlsconfig
Configuration data for nodeset 1:
---------------------------------
clusterType lc
comm_protocol TCP
multinode yes
autoload no
useSingleNodeQuorum no
useDiskLease yes
group Gpfs.1
```

```
recgroup GpfsRec.1
maxFeatureLevelAllowed 700

File systems in nodeset 1:
--------------------------
(none)
```

**mmlsconfig** shows the GPFS nodeset id that was assigned (in our case, it was 1).You can tailor this at **mmconfig** time, but the important thing is to remember the id, as it will be used to start up GPFS on the nodeset.

**Note:** Be aware that the nodeset id is only significant if you want to have multiple nodesets (which is optional, and not common for lc clusters). If there is only one nodeset in the cluster, there is no reason to know the nodeset, as you never need to supply it (GPFS, by default, picks the id of the nodeset on which the command is being executed).

### *Start up the GPFS nodeset*

We are now ready to start up GPFS. This is done with the **mmstartup** command, to which we must give the GPFS nodeset id as returned by mmlsconfig. This is described in Example 6-19.

It is useful to check the status of Topology Services and Group Services after GPFS has been started. To do so, use the **lssrc** command with the **-ls** flag on each of the two subsystems.

*Example 6-19 Starting up the GPFS nodeset*

```
r01n33:~ # mmstartup -C 1
Thu Oct 30 15:14:31 PST 2003: mmstartup: Starting GPFS ...
gr01n33:  0513-059 The mmfs Subsystem has been started. Subsystem PID is 18224.
gr01n34:  0513-059 The mmfs Subsystem has been started. Subsystem PID is 8145.

r01n33:~ # lssrc -ls cthats
Subsystem         Group            PID     Status
 cthats           cthats           17262   active
Network Name   Indx Defd Mbrs St Adapter ID      Group ID
CG1            [ 0]    2    2  S 10.10.10.33     10.10.10.34
CG1            [ 0] eth1         0x47a19a84      0x47a19a7e
HB Interval = 1 secs. Sensitivity = 4 missed beats
Missed HBs: Total: 0 Current group: 0
Packets sent    : 261 ICMP 0 Errors: 0 No mbuf: 0
Packets received: 317 ICMP 0 Dropped: 0
NIM's PID: 17321
CG2            [ 1]    2    2  S 129.40.34.33    129.40.34.34
CG2            [ 1] eth0         0x47a19a82      0x47a19a7d
HB Interval = 1 secs. Sensitivity = 4 missed beats
```

```
Missed HBs: Total: 0 Current group: 0
Packets sent    : 262 ICMP 0 Errors: 0 No mbuf: 0
Packets received: 317 ICMP 0 Dropped: 0
NIM's PID: 17324
  2 locally connected Clients with PIDs:
 rmcd( 17378) hagsd( 17266)
  Configuration Instance = 1067555418
  Default: HB Interval = 1 secs. Sensitivity = 8 missed beats
  Daemon employs no security
  Segments pinned: Text Data Stack.
  Text segment size: 131611 KB. Static data segment size: 595 KB.
  Dynamic data segment size: 939. Number of outstanding malloc: 130
  User time 0 sec. System time 0 sec.
  Number of page faults: 1100. Process swapped out 0 times.
  Number of nodes up: 2. Number of nodes down: 0.
r01n33:~ # lssrc -ls cthags
Subsystem         Group            PID     Status
 cthags           cthags           17266   active
3 locally-connected clients.  Their PIDs:
17200(IBM.ConfigRMd) 17378(rmcd) 18344(mmfsd)
HA Group Services domain information:
Domain established by node 1
Number of groups known locally: 5
                   Number of   Number of local
Group name         providers   providers/subscribers
GpfsRec.1                2           1           0
Gpfs.1                   2           1           0
rmc_peers                2           1           0
NsdGpfs.1                2           1           0
IBM.ConfigRM             2           1           0
```

If the portability layer was not properly compiled and installed on all the nodes, **mmstartup** will fail.

You may also see, in the GPFS log file (/var/mmfs/gen/mmfslog), entries such as those shown in Example 6-20.

*Example 6-20   mmfsd will not start without the portability layer*

```
# cat /var/mmfs/gen/mmfslog
Wed Oct 29 08:07:10 PST 2003 runmmfs starting
Removing old /var/adm/ras/mmfs.log.* files:
/bin/mv: cannot stat `/var/adm/ras/mmfs.log.previous': No such file or
directory
Unloading modules from /usr/lpp/mmfs/bin
Error: /usr/lpp/mmfs/bin/mmfslinux kernel extension does not exist.
       Please compile a custom mmfslinux module for your kernel.
       See /usr/lpp/mmfs/src/README for directions.
Error: unable to verify kernel/module configuration
```

```
Loading modules from /usr/lpp/mmfs/bin
Error: /usr/lpp/mmfs/bin/mmfslinux kernel extension does not exist.
       Please compile a custom mmfslinux module for your kernel.
       See /usr/lpp/mmfs/src/README for directions.
Error: unable to verify kernel/module configuration
Wed Oct 29 08:07:10 PST 2003 runmmfs: error in loading or unloading the mmfs
kernel extension
Wed Oct 29 08:07:10 PST 2003 runmmfs: stopping SRC ...
0513-044 The mmfs Subsystem was requested to stop.
Wed Oct 29 08:07:10 PST 2003 runmmfs: received an SRC stop request; exiting
```

In our case, it was compiled and installed properly, and the GPFS kernel modules show up as illustrated in Example 6-21.

*Example 6-21 GPFS kernel modules loaded*

```
r01n33:~ # lsmod
Module                  Size  Used by    Tainted: PF
mmfs                 1106392   1
mmfslinux             219800   1  [mmfs]
tracedev               14880   1  [mmfs mmfslinux]
ipv6                  481800  -1  (autoclean)
key                   102936   0  (autoclean) [ipv6]
e1000                 152368   1
e100                  106696   1
lvm-mod               110912   0  (autoclean)
```

### Create the local disk partitions (optional)

GPFS will accept anything that looks like a block device, whether it is a partition or an entire disk.

In our example, we set aside a partition on each node, as listed in Example 6-22. On the first node, we used /dev/sda4, and on the second node, we used /dev/sdb4.

*Example 6-22 Local disk partitions*

```
r01n33:~ # fdisk /dev/sda

The number of cylinders for this disk is set to 34715.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help): p

Disk /dev/sda: 64 heads, 32 sectors, 34715 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1         4      4080   41  PPC PReP Boot
/dev/sda2             5      1029   1049600   82  Linux swap
/dev/sda3          1030     10246   9438208   83  Linux
/dev/sda4         10247     26631  16778240   83  Linux

r01n34:~ # fdisk /dev/sdb

The number of cylinders for this disk is set to 34715.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdb: 64 heads, 32 sectors, 34715 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1   *         1         4      4080   41  PPC PReP Boot
/dev/sdb2             5      1029   1049600   82  Linux swap
/dev/sdb3          1030      9222   8389632   83  Linux
/dev/sdb4          9223     25607  16778240   83  Linux
```

### *Create the Network Shared Disks (NSDs)*

It is now time to create the Network Shared Disks (NSDs) that we will use to store our file data (this step could have been done after cluster creation).

We used the **mmcrnsd** command to perform this operation. We gave it a description file, listed in Example 6-23. The **mmlsnsd** is used to display the NSDs just created.

The purpose of this step is to prepare all the NSDs, and to assign, to each NSD, a unique name across the cluster; a PVID is stored in the NSD itself. We need a unified naming scheme, because each node names its local partitions irrespective of the other nodes.

As Example 6-23 on page 300 shows, we have two NSDs (one per node) capable of storing file system data and file system meta data, each one served by a single server (we have no twin-tailed disks here). Each disk belongs to its own failure group, as there is no common point of failure for the two disks, which are on separate machines. Refer to GPFS documentation for a full description of the syntax.

*Example 6-23 Create the NSDs and check them*

```
r01n33:~ # cat /tmp/gpfsnsd
/dev/sda4:gr01n33::dataAndMetadata:1
/dev/sdb4:gr01n34::dataAndMetadata:2

r01n33:~ # mmcrnsd -F /tmp/gpfsnsd
mmcrnsd: Propagating the changes to all affected nodes.
This is an asynchronous process.

r01n33:~ # mmlsnsd

 File system   Disk name    Primary node            Backup node
---------------------------------------------------------------------------
 (free disk)   gpfs1nsd     gr01n33
 (free disk)   gpfs2nsd     gr01n34
```

### *Create a file system*

GPFS is in operation, and we have NSDs available for receiving data. Now we can create a file system. We do this using the **mmcrfs** command, as shown in Example 6-24 on page 300.

**mmcrnsd** is clever enough to modify the disk description file, which we gave it in “Create the Network Shared Disks (NSDs)” on page 299, and convert it to a format suitable to the mmcrfs command. The local partition names are replaced by the global, cluster-wide NSD names. Refer to the **mmfscrfs** man page for a detailed explanation of its syntax. In our example, we use no data replication and we chose to name the file system /dev/gpfs0 and to mount it automatically (-A 1) under /bigfs.

The **mmlsdisk** command lists the current usage of the NSD disks by GPFS. **df -h** shows that the file system is indeed mounted and the **dd** command demonstrates that we can write into the GPFS file system. The file system is mounted on all the nodes in the nodeset.

*Example 6-24 GPFS file system creation*

```
r01n33: # cat /tmp/gpfsnsd
# /dev/sda4:gr01n33::dataAndMetadata:1
gpfs1nsd:::dataAndMetadata:1
# /dev/sdb4:gr01n34::dataAndMetadata:2
gpfs2nsd:::dataAndMetadata:2

r01n33: # mmcrfs /bigfs gpfs0 -F /tmp/gpfsnsd -C 1
```

```
The following disks of gpfs0 will be formatted on node r01n33:
    gpfs1nsd: size 16778240 KB
    gpfs2nsd: size 16778240 KB
Formatting file system ...
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Flushing Allocation Maps
Completed creation of file system /dev/gpfs0.
mmcrfs: Propagating the changes to all affected nodes.
This is an asynchronous process.
r01n33:~ # mmlsdisk gpfs0
disk         driver   sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- -------- ----- ------------- ------------
gpfs1nsd     nsd         512       1 yes      yes   ready         up
gpfs2nsd     nsd         512       2 yes      yes   ready         up

r01n33:~ # df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             9.1G  4.5G  4.6G  50% /
shmfs                 7.3G     0  7.3G   0% /dev/shm
/dev/gpfs0             33G   43M   32G   1% /bigfs

r01n33:~ # dd if=/dev/zero of=/bigfs/junk bs=1024k count=1024
1024+0 records in
1024+0 records out
```

### 6.3.5 Shutting down and restarting GPFS

To shut down GPFS, the first step is to unmount the GPFS file systems on all the nodes where they are mounted. Then the **mmshutdown** command is issued on one of the nodes as shown in Example 6-25 on page 301.

*Example 6-25 Shutting down GPFS*

```
r01n33:~ # mmshutdown -a
Thu Oct 30 14:57:19 PST 2003: mmshutdown: Starting force unmount of GPFS file
systems
Thu Oct 30 14:57:24 PST 2003: mmshutdown: Shutting down GPFS daemons
r01n33:  0513-044 The mmfs Subsystem was requested to stop.
r01n33:  Shutting down!
r01n33:  Unloading modules from /usr/lpp/mmfs/bin
r01n33:  Unloading module mmfs
r01n33:  Unloading module mmfslinux
r01n33:  Unloading module tracedev
```

```
r01n34:  0513-044 The mmfs Subsystem was requested to stop.
r01n34:  Shutting down!
r01n34:  Unloading modules from /usr/lpp/mmfs/bin
r01n34:  Unloading module mmfs
r01n34:  Unloading module mmfslinux
r01n34:  Unloading module tracedev
Thu Oct 30 14:57:33 PST 2003: mmshutdown: Finished
```

To restart GPFS, simply issue the **mmstartup -C nodesetID** command as shown in Example 6-19 on page 296. If you created your GPFS file systems with the automatic mount on start up option, they should get mounted now.

7

# High Performance Computing case studies

With the rapid advances in semiconductor technology, most computing tasks that used to require supercomputers can now be performed on desktop workstations. However, there are always areas and applications that challenge and push the limit of computing power. Such applications include the “grand challenge problems” of weather forecasting, nuclear testing, drug designing, automobile designing, and so on. For these challenges, High Performance Computing (HPC) clustering is often the best solution.

An HPC cluster often consists of the following components:

- A large number of powerful computing nodes
- A high speed network that interconnects the computing nodes
- A high performance global file system that is accessible from all the computing nodes
- An efficient job scheduling system to make effective use of the cluster
- A rich set of compilers and performance libraries for various applications

In this chapter, we address these issues for an IBM pSeries cluster running Linux, and discuss the performance of some HPC benchmarks.

The chapter includes:

- 7.1, “Hardware and software overview” on page 305, a hardware and software overview from the HPC perspective.
- 7.2, “Myrinet” on page 305 describes Myrinet installation and set up.
- 7.3, “Gigabit Ethernet” on page 312 contains a brief discussion of Gigabit Ethernet.
- 7.4, “Compilers and ESSL libraries” on page 313 discusses available compilers and their installation. We also describe how to install IBM ESSL and PESSL libraries. We present case studies for frequently used industrial standard benchmarks that also serve as a test of the installed system.

## 7.1 Hardware and software overview

Hardware and software supported for pSeries running Linux are discussed in 2.1, “Before you install” on page 7. You can also find similar information from the following Web site:

```
http://www.ibm.com/servers/eserver/pseries/hardware/linux_facts.pdf
```

From an HPC point of view, the following points may be of special interest:

- All POWER4 or POWER4+-based systems are supported. This includes p615, p630, p650, p655, p670, and p690, providing a whole range of systems from entry level workstations to enterprise servers.
- All of the following Gigabit Ethernet adapters are supported on all supported pSeries hardware:
  - 5700 Gigabit Ethernet - SX PCI-X (Fibre)
  - 5701 Gigabit Ethernet - Base-TX PCI-X (UTP)
  - 5706 Gigabit Ethernet (UTP) 2-port
  - 5707 Gigabit Ethernet (Fibre) 2-port (in our case study for HPC)
- Myrinet PCI-X-based adapters (the most recent generation) are supported, providing low latency, high bandwidth network connectivity.
- IBM’s proven Fortran compiler (XLF 8.1), C and C++ compiler (VACPP 6.0), Engineering and Scientific Subroutine Library (ESSL 4.1) and its parallel distributed memory version (PESSL3.1) are supported. This not only provides high performance to pSeries Linux applications due to its POWER4 architecture affinity and its Symmetrical Multiple Processor (SMP) scalability, but also provides great convenience as follows:
  - For AIX users transitioning to Linux, since all these products are originally from AIX and their usage is almost the same as in an AIX environment.
  - For 32-bit application programmers embracing a 64-bit world, because these compilers and libraries support both 32-bit and 64-bit application space.
- GNU gcc/g++/g77 for 32-bit and cross-compilers for 64-bit are also available. In addition, much open source software popular in the HPC world has been enabled and is available as RPM packages.

## 7.2 Myrinet

Myrinet is the brand name for all network products from Myricom Inc.; see:

```
http://www.myri.com
```

The products range from PCI- and PCI-X based network adapters, switches, cables, to software support such as device drivers and middleware. In this section, we describe the installation of Myrinet products, from the device driver to message passing interface library.

## 7.2.1 Myrinet hardware

Table 7-1 lists characteristics of the latest Myrinet adapters, the single-ported D-card (M3F-PCIXD-2), and the dual-ported E-card (M3F-PCIXE-2), which are supported on pSeries running Linux (ppc64).

*Table 7-1 Myrinet adapters supported on pSeries running Linux*

| Adapters | PCI-X width and speed | Number of Ports | Interface processor speed | Device drivers |
|---|---|---|---|---|
| M3F-PCIXD-2 | 64bit/133 MHz | 1 | 225 MHz | GM 2 or MX |
| M3F-PCIXE-2 | 64bit/133MHz | 2 | 333 MHz | GM 2.1 or MX |

Figure 7-1 shows what each adapter looks like.

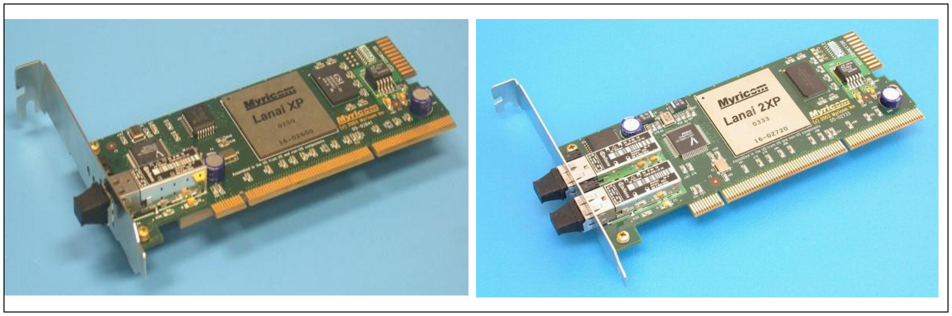


*Figure 7-1 Myrinet D-card (left) and E-card (right)*

It should be noted that:

- Both cards work with the Myrinet 2000 switch.
- If only one port is used, the dual-ported card will deliver the same bandwidth as the single-ported card. However, the E-card will have lower latency because its interface processor has higher clock speed.
- The two ports on the E-card can be on the same Myrinet fabric or on separate, independent Myrinet fabrics. In the former case, you can plug in and out one cable without losing network connection; thus, some level of high performance and high availability are provided by using both ports.

- The new MX driver has lower latency than GM; further details are listed in Table 7-2.

*Table 7-2 User-level short-message latency of GM and MX*

| GM-2 | 5μs |
|---:|---|
| MPICH-GM over GM-2 | 5.5μs |
| MX | 3.5μs |
| MPICH-MX over MX | 3.5μs |

This information is derived from:

http://www.myri.com/news/03a14/

## 7.2.2 GM installation and enablement

MX was not available at the time of writing (and GM version 2.0.8 became available while the project was running). Although we tested GM versions 2.0.3 and 2.0.6, we use only version 2.0.6 in our redbook examples.

The GM installation requires root privileges. It consists of the following steps.

1. Compile and install the device driver.
2. Configure the GM mapper.
3. Load and start the driver.

### Compile and install the device driver

Before starting compilation, you need to do some planning—and a major planning consideration is the addressing space. With the ability to support both 32-bit and 64-bit at the hardware level of pSeries processors, and at the software level with, for example, XLF and VACPP compilers and the ESSL/PESSL libraries, it would be useful to HPC users to have the flexibility of being able to work on both 32-bit and 64-bit applications. Therefore, it would be beneficial to have both 32-bit and 64-bit support from GM drivers and other libraries, such as MPICH_GM.

#### *32-bit build*

Example 7-1 on page 308 lists commands to build and install a 32-bit device driver (gm.o) and application library (libgm.so). The device driver will not be used because the only Linux kernel supported is 64-bit.

However, the 32-bit application library is needed for building 32-bit applications.

*Example 7-1 32-bit GM driver build*

```
/bin/bash
install_dir=/opt/gm-2.0.6/32
mkdir -p $install_dir
export CC=gcc
./configure --host=ppc64-unknown-linux --prefix=$install_dir
make
cd binary # if make has been successful
./GM_INSTALL $install_dir
```

### *64-bit build*

Example 7-2 shows a script for a 64-bit build. (Note that the default installation dir is /opt/gm for either a 32-bit or 64-bit build.)

*Example 7-2 64-bit GM driver build*

```
/bin/bash
install_dir=/opt/gm-2.0.6/64
mkdir -p $install_dir
export PATH=/opt/cross/bin:$PATH
export LD=/opt/cross/powerpc64-linux/bin/ld
export RANLIB=/opt/cross/powerpc64-linux/bin/ranlib
export STRIP=/opt/cross/powerpc64-linux/bin/strip
export CC=powerpc64-linux-gcc
./configure --host=ppc64-unknown-linux --prefix=$install_dir
make
cd binary # if make has been successful
./GM_INSTALL $install_dir
```

**Note:** Configuring and compiling GM requires a Linux source tree to be available, and the default location is /usr/src/linux. The kernel header files must match the running kernel exactly.

If the source tree is at a different place, you must provide the option **`--with-linux=<linux-src-dir>`** to the **`configure`** command.

The GM_INSTALL script copies GM binaries to the specified installation directory $install_dir (which is /opt/gm-2.0.6/32 for the 32-bit build, and /opt/gm-2.0.6/64 for the 64-bit build).

## Configure the GM mapper

The GM mapper is responsible for discovering hosts on the Myrinet network, and then computing the route to each host. There are two ways to accomplish this, depending on the level of high availability required for the Myrinet network: by using manual mapping, or by using active/passive mapping. We explain both techniques in the following section.

### *Manual mapping (map-once)*

This is the favored mapping technique for small clusters where the possibility of nodes leaving the Myrinet is low. To enable this mapping mode, the following command is needed for *each* node:

```
/opt/gm-2.0.6/64/sbin/gm_install_drivers --manual-mapper
```

After the kernel module is loaded (see below), you start the mapping with this command:

```
/opt/gm-2.0.6/64/sbin/gm_mapper --map-once -verbose
```

### *Active/passive mapping (dynamic mapping)*

This is the preferred mapping for large clusters where some nodes (active) continuously query the Myrinet network to build the map table for other (passive) nodes. Since mapping is created dynamically, the cluster is able to recover itself from falling nodes. This is done by running the following command on *every* node:

```
/opt/gm-2.0.6/64/sbin/gm_install_drivers --active-mappers=”lp05 lp08”
```

where nodes lp05 and lp08 serve as active mappers. Verify that these are the names returned by the **hostname** command.

## Load and start the driver

The last step in making Myrinet available to applications is to load the GM module on all participating nodes. This is done by running the following command on *every* node:

```
/etc/init.d/gm start
```

As with other network devices, you can restart or stop the device by providing **restart** or **stop** to the command **/etc/init.d/gm**.

**Tip:** You may be wondering, should you use manual mapping or dynamic mapping? Since overhead is not a concern here, dynamic mapping seems more attractive because it heals itself without human interference.

However, there are potentially serious problems with dynamic (active/passive) mapping. For example, with very intensive MPI applications, the GM mapping may break. The high communication rate may take all the bandwidth, causing some GM mapping control messages to be delayed, or lost. This makes GM think that some nodes are down and it will therefore terminate the MPI application.

Even worse, in some cases, the application exhibits unpredictable behavior and gives wrong answers, simply based on the interaction between application messages and GM mapping control messages.

To prevent this from happening and yet still use the active/passive mapping during mapping configuration and kernel module loading, run the following command on all nodes:

```
gm_install_drivers --active-mappers=”lp05 lp08”
```

Then load the GM module, using **/etc/init.d/gm start**. After this is done, execute **killall gm_mapper** on all nodes.

This technique works, although it is not supported by Myricom; the README-linux file states “Stopping the mapper while GM is running is not supported. The gm_mapper should be left running at all times, and it will not interfere with the performance of jobs running over Myrinet”.

### 7.2.3 IP over Myrinet

GM by default supports 16 ports, but only 13 ports (ports 2, 4-15) are available to non-privileged user applications. Port 0 is for internal GM use and is never available. Port 1 is used by the GM mapper when running. Port 3 is used by the IP-over-Myrinet driver.

The install procedure does not enable IP mode of the Myrinet card. In order to IP-over-Myrinet, run this command:

```
ifconfig myri0 <ip_addr> up
```

### 7.2.4 Checking GM status and performance

Myrinet provides a set of scripts and test programs to show the status and performance of the Myrinet network. We assume /opt/gm-2.0.6/bin on the system PATH.

To check the status of the GM ports, issue the following command:

```
gm_board_info
```

To test DMA rate between memory and Myrinet adapter:

```
gm_debug -L
```

The performance should then be compared with the published numbers, if available, at:

http://www.conservativecomputers.com/myrinet/perf.html

To test point-to-point communication, issue the following command on one node, for example, lp07:

```
gm_allsize --slave --size=15
```

And also on another node, for example, lp08:

```
gm_allsize --unidirectional --bandwidth --remote-host=lp07 --size=15
--geometric
```

### 7.2.5 MPICH_GM installation

Any user can install MPICH_GM. The installation of MPICH_GM does not require root access, as long as the installation directory (/opt/gm-2.0.6/32, in our example) does not require it.

After downloading the source code from the Myricom Web site and untarring it, you can build a 32-bit MPICH_GM library with the script in Example 7-3.

A 64-bit library can be built with minor modifications (refer to Example 7-2 on page 308 and Example 7-3 for hints). Note that we used XLF for Fortran and Fortran90 support.

*Example 7-3 Script to build a 32-bit MPICH_GM*

```
/bin/sh

CC=gcc
CXX=g++
FC=”xlf_r -qnosave -q32 -qpic=large”
F90=”xlf90_r -q32 -qpic=large”
```

```
GM_HOME=/opt/gm-2.0.6/32
CFLAGS=”-I${GM_HOME}/include”
export CC CXX FC F90 GM_HOME CFLAGS
PREFIX=/opt/mpich-1.2.5..10/32
mkdir -p $PREFIX
./configure --with-device=ch_gm -prefix=$PREFIX -opt=”-O2” \
-lib=”-lgm -lpthread” --enable-sharedlib
make
make install
```

## 7.3 Gigabit Ethernet

Though with larger overhead to computing nodes and lower bandwidth for inter-node communication compared with Myrinet, Gigabit Ethernet (GigE), with its advantage in price, can sometimes be the choice for an HPC environment. This is especially true if applications to be run on such a system are:

- Embarrassingly Parallel. In this case, there is no communication, or very little communication, between processes. Examples are sequence searching programs in life sciences (BLAST, FASTA), low level image processing, Mandelbrot set, and Monte Carlo simulations.
- Hybrid message passing and multi-threading. In this case, one can adjust the number of processes and the number of threads to achieve a balanced computing and communication performance. For example, for a cluster of two SMP nodes each with 32 processors, you can run a hybrid MPI/OpenMP job with two MPI processes and 32 OpenMP threads per MPI process, making full use of the total 64 processors. You can also start four MPI processes each with 16 threads, and so on.

In our case study, discussed in the following section, we have a configuration of two p655 nodes interconnected by Gigabit Ethernet.

In this case, the GM driver and MPICH_GM are not needed. You can use ANL’s MPICH to compile and run MPI programs. The installation is more or less the same as MPICH_GM. It is worth noting that in order to use the SMP feature, it is better to have two kinds of installations:

- One with the following configuration option:

  ./configure --with-device=ch_p4 --with-comm=shares
- And the other with:

  ./configure --with-device=ch_shared

The first one will work with a cluster of SMP nodes in which the intra-node communication will be done through a shared memory mechanism, while the inter-node communications go to the network adapters.

The second one will only work on a single SMP node using shared memory for communications. However, somehow it performs better than the first one.

So, it's useful to have both installed.

## 7.4 Compilers and ESSL libraries

As of today, in addition to open source compilers, the following compilers have recently been added to the family of pSeries running Linux.

- IBM XL Fortran V8.1 for Linux on pSeries, see:

`http://www.ibm.com/software/awdtools/fortran/xlfortran/features/xlf-linux.html`

- IBM VisualAge® C++ V6.0 for Linux on pSeries, see:

`http://www.ibm.com/software/awdtools/vacpp/features/vacpp-linux.html`

- Absoft Pro Fortran 8.2 for PPC/Linux , see:

`http://www.absoft.com/newppcproductpage.html`

Some features of the compilers are listed in Table 7-3.

*Table 7-3 Linux on pSeries - available compilers and their features*

| | IBM XLF/VACPP | GNU gcc/g++/g77 | Cross Compilers gcc/g++/g77 | Absoft Pro Fortran |
|---|---|---|---|---|
| **C/C++** | yes | yes | yes | no |
| **Fortran** | F77/F90/F95 | F77 | F77 | F77/F90/F95 |
| **App Space** | 32- and 64-bit | 32-bit | 64-bit | 32-bit |
| **OpenMP** | yes | no | no | no* |
| **Auto Parallelization** | yes | no | no | no* |

*You need a third-party pre-processor to support these features.

### 7.4.1 Installing GNU and cross-compilers

XLF, VACPP and Pro Fortran compilers all require gcc to be installed. In addition, many HPC applications in the Linux world also need gcc for compiling either whole program suites or some of the utility programs. 64-bit applications need a 64-bit environment, which is provided by cross-compilers.

GNU g77 compilers are also often needed even with the existence of other, more advanced compilers such as the XLF compiler. Therefore, it is highly recommended that the whole set of GNU and cross compilers gcc, g++, and g77, be installed.

The best time to do this is during operating system installation so that dependencies and so forth are taken care of by installation software. Nevertheless, we list here a script to do rpm installation.

The command lines to install the whole set of GNU compilers (gcc, g++, and g77, all 32-bit) are listed in Example 7-4.

*Example 7-4 Installation of complete set of GCC (32-bit) and libelf.*

```
# gcc
rpm -ivh binutils-2.12.90.0.15-67.ppc.rpm
rpm -ivh cpp-3.2.2-35.ppc.rpm
rpm -ivh libgcc-3.2.2-35.ppc.rpm
rpm -ivh glibc-devel-2.2.5-139.ppc.rpm
rpm -ivh gcc-3.2.2-35.ppc.rpm
# g++
rpm -ivh libstdc++-3.2.2-35.ppc.rpm
rpm -ivh libstdc++-devel-3.2.2-35.ppc.rpm
rpm -ivh gcc-c++-3.2.2-35.ppc.rpm
# g77
rpm -ivh gcc-g77-3.2.2-35.ppc.rpm
# libelf
rpm -ivh libelf-0.8.2-35.ppc.rpm
```

In Example 7-5, we list command lines to install 64-bit cross compilers that include c/c++/f77 components. We chose a lower version of the compiler (3.2-49, instead of 3.2.2-44, which was also available at the time of installation) since the higher version does not work well; the c++ compiler does not even compile a “Hello World” code.

*Example 7-5 Installation of 64-bit cross compilers*

```
# Install packages
rpm -ivh cross-ppc64-binutils-2.13.90.0.4-81.ppc.rpm
rpm -ivh cross-ppc64-glibc-2.2.5-129.ppc.rpm
rpm -ivh cross-ppc64-gcc-3.2-49.ppc.rpm
# Add to the $PATH, assuming sh or bash.
cat > /etc/profile.d/cross_compilers.sh << EOF
PATH=/opt/cross/bin:$PATH
export PATH
EOF
```

### 7.4.2 Installing XL Fortran and VisualAge C++ compilers

Installing XL Fortran V8.1 (XLF) for Linux on pSeries and VisualAge C++ (VACPP) V6.0 for Linux on pSeries is straightforward as long as GCC and libelf have been properly installed. The 64-bit cross compilers are necessary only for 64-bit support of XLF and VACPP.

Example 7-6 on page 315 lists command lines to install the packages and to configure the environment. The "new_install" script is an interactive script which asks two simple questions about language choice and license acceptance and then does the post-installation configuration.

*Example 7-6 Installation of IBM XL Fortran 8.1 VisualAge C++ 6.0*

```
# XLF installation
rpm -ivh xlsmp.msg.rte-1.3.7-0.ppc64.rpm
rpm -ivh xlsmp.rte-1.3.7-0.ppc64.rpm
rpm -ivh xlsmp.lib-1.3.7-0.ppc64.rpm
rpm -ivh xlf.msg.rte-8.1.0-0.ppc64.rpm
rpm -ivh xlf.rte-8.1.0-0.ppc64.rpm
rpm -ivh xlf.cmp-8.1.0-0.ppc64.rpm
rpm -ivh xlf.help-8.1.0-0.ppc64.rpm
rpm -ivh xlf.samples-8.1.0-0.ppc64.rpm
/opt/ibmcmp/xlf/8.1/bin/new_install

# For VACPP - xlsmp.rte and xlsmp.lib are needed
rpm -i  vacpp.rte-6.0.0-0.ppc64.rpm
rpm -i  vacpp.help-6.0.0-0.ppc64.rpm
rpm -i vacpp.samples-6.0.0-0.ppc64.rpm
rpm -i vac.cmp-6.0.0-0.ppc64.rpm
rpm -i vacpp.cmp-6.0.0-0.ppc64.rpm
/opt/ibmcmp/vac/6.0/bin/new_install

# Set up PATH
cat > /etc/profile.d/ibmcmp.sh << EOF
PATH=/opt/ibmcmp/xlf/8.1/bin:/opt/ibmcmp/vacpp/6.0/bin\
:/opt/ibmcmp/vac/6.0/bin:$PATH
export PATH
EOF
```

In most cases, the order of the command lines matters because of dependencies. After the installation, adding the path is the only thing needed to use the compilers. 32-bit and 64-bit support are done through compiler options (-q32, -q64) or through the use of an environment variable: OBJECT_MODE.

## 7.4.3 Installing Absoft Pro Fortran 8.2 compiler

Installing Absoft Pro Fortran 8.2 for LinuxPPC involves three steps, as shown in Example 7-7.

*Example 7-7 Installation of Absoft Pro Fortran for LinuxPPC*

```
rpm -ivh absoft_profortran-8.2-1.ppc.rpm
cp license.dat /opt/absoft
ABSOFT=/opt/absoft
PATH=/opt/absoft/bin:$PATH
export ABSOFT PATH
```

**Important:** In our example listings, we always put the PATH of the current software first to make sure the current software will be used if there are name conflicts.

However, in a production environment, you must be careful about the order of the PATH. For example, there are f77 in /usr/bin, /opt/ibmcmp/xlf/8.1/bin, /opt/absoft/bin. We recommend the following settings be in the system-wide startup script:

```
/opt/ibmcmp/xlf/8.1/bin:/opt/ibmcmp/vacpp/6.0/bin:/opt/ibmcmp/vac/6.0/bin:/o
pt/cross/bin:$PATH
```

Absoft PATH and environment (ABSOFT) can be added on the fly by users when there is a need.

## 7.4.4 Installing ESSL 4.1 and PESSL 3.1

For more detailed information on ESSL installation, see “Engineering and Scientific Subroutine Library for Linux on pSeries, Version 4 Release 1 Installation Guide (September 2003)”. In Example 7-8, we list the installation steps, followed by a brief explanation.

*Example 7-8 Installation of ESSL and PESSL*

```
# ESSL
rpm -ivh essl.license-4.1.0-0.ppc64.rpm
cd /opt/ibmmath/essl/4.1/bin
./install_essl -y -d /mnt/install/essl/
# PESSL
```

```
rpm -ivh pessl.license-3.1.0-0.ppc64.rpm
cd /opt/ibmmath/pessl/3.1/bin
./install_pessl -y -d /mnt/install/essl/
```

In Example 7-8, `-y` is an optional flag indicating that you are accepting the license agreement without being prompted. `-d rpmpath` is an optional flag specifying the directory that contains the ESSL packages. To use the default directory (/media/cdrom), do not specify this flag.

Header files and libraries (or soft links of them) are placed at the system default places /usr/include and /usr/lib after successful installation. However, the use of the libraries for compiling, linking, running, and so on depends on the availability of the XLF compiler, XLF runtime, GCC; refer to the documentation for further details.

## 7.5 More packages ported and installed

In addition to the compilers, communication packages (MP), and ESSL/Parallel ESSL library that we have discussed so far, we have also ported/installed these packages, which are frequently used in HPC community: BLAS, CBLAS, LAPACK, MPICH, LAM/MPI, FFTW, BLACS, ScaLAPACK, WSMP, SuperLU, PETSC.

## 7.6 Benchmarks

In this section, we discuss industry standard benchmarks performed on pSeries running Linux.

### 7.6.1 System configurations

#### Hardware

Eight p655 nodes (lp01-08) interconnected by the Myrinet 2000 switch. Each node has eight 1.1 GHz POWER4 processors, private 32 KB L1 Data and 64 KB L1 Instruction cache per processor, 4x1.44 MB L2 cache, 32 GB memory, two Myrinet D-cards.

Two of the nodes (lp07, lp08) have direct peer-to-peer Gigabit Ethernet connection. The schematic diagram is shown in Figure 7-2 on page 318.

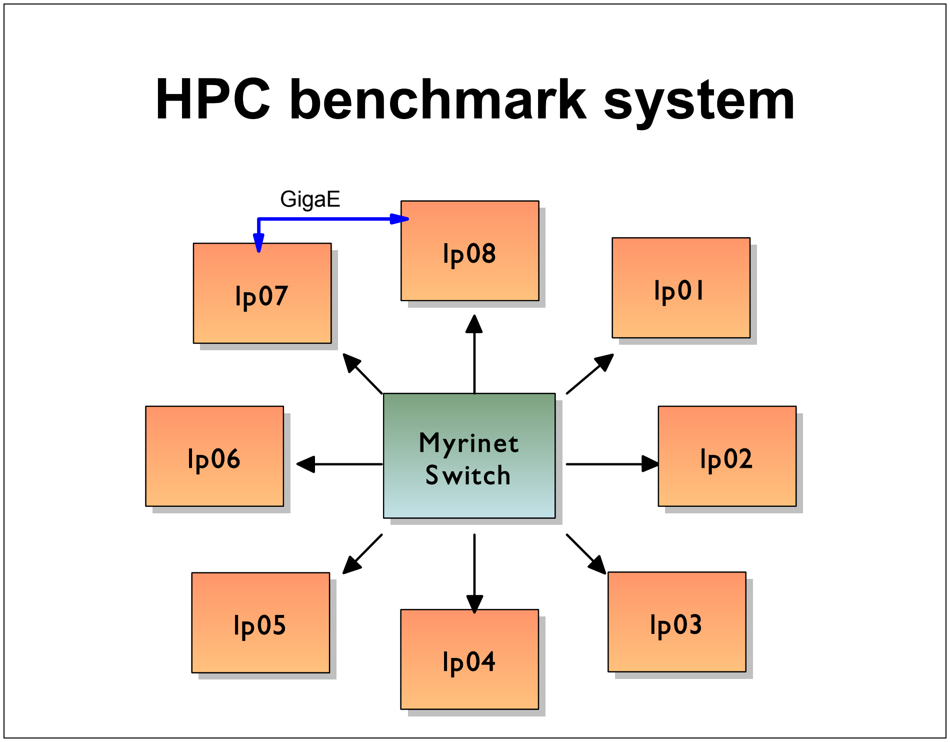


*Figure 7-2 Node configuration for HPC case study*

### Software

- SuSE SLES 8 (ppc) Linux version 2.4.19-ul1-ppc64-SMP
- GCC 3.2
- IBM XLF 8.1 for Linux on pSeries
- IBM VACPP 6.0 for Linux on pSeries
- Myrinet GM 2.06, MPICH_GM 1.2.5..10
- ANL (Argonne National Lab) MPICH 1.2.5
- IBM ESSL 4.1 for Linux on pSeries

## 7.6.2 Linpack DP n=100 Benchmark

For further details on this study, refer to the following site:

`http://www.netlib.org/benchmark/linpackd`

Linpack DP (Double Precision) is one of three Linpack benchmarks (also see 7.6.3, “Linpack TPP” on page 320 and 7.6.4, “Linpack HPC” on page 322) that measure the number of floating point operations per second. It solves a system of linear equations:

$$\boldsymbol{Ax=b}$$

where ***A*** is a dense matrix of order 100 by 100, and ***x*** and ***b*** are vectors of length 100. It is a serial source code written in Fortran double precision. The source code is not allowed to change, even comments. However, a function subroutine “second()” which returns elapsed time in seconds has to be provided by a user.

This is a very small problem for modern computers and does not reflect the best performance a modern computer can deliver. Nevertheless, it can measure, to some extent, the performance of processors and compilers. Because of this and due largely to historical reasons, it still appears in some customer benchmark requests.

In our measurement, we replaced the function subroutine “second()” by XLF runtime function “rtc()” and used the following command and options to compile and run the code:

```
xlf -O3 -qhot -qarch=pwr4 -qtune=pwr4 -o linpackd linpackd.f
time ./linpackd
```

Here is the output file of the program:

*Example 7-9 Program output*

```
    Please send the results of this run to:

     Jack J. Dongarra
     Computer Science Department
     University of Tennessee
     Knoxville, Tennessee 37996-1300

     Fax: 865-974-8296

 Internet: dongarra@cs.utk.edu

    norm. resid      resid           machep         x(1)          x(n)
    1.23217369E+00  1.36796649E-14  2.22044605E-16  1.00000000E+00  1.00000000E+00

       times are reported for matrices of order   100
    dgefa      dgesl      total     mflops       unit      ratio
 times for array with leading dimension of 201
1.010E-03  3.695E-05  1.047E-03  6.558E+02  3.050E-03  1.870E-02
9.960E-04  3.695E-05  1.033E-03  6.648E+02  3.009E-03  1.845E-02
9.960E-04  3.695E-05  1.033E-03  6.648E+02  3.009E-03  1.845E-02
9.996E-04  3.591E-05  1.036E-03  6.631E+02  3.016E-03  1.849E-02


times for array with leading dimension of 200
1.012E-03  3.600E-05  1.048E-03  6.552E+02  3.052E-03  1.871E-02
1.007E-03  3.600E-05  1.043E-03  6.584E+02  3.038E-03  1.862E-02
1.013E-03  3.695E-05  1.050E-03  6.540E+02  3.058E-03  1.875E-02
1.008E-03  3.619E-05  1.044E-03  6.575E+02  3.042E-03  1.865E-02
  end of tests -- this version dated 10/12/92
```

The resulting MFLOPS, along with time information returned by the Linux **time** command, is shown in Figure 7-4.

*Table 7-4 Linpack DP n=100 result*

| Real time (s) | User time (s) | Sys time (s) | MFLOPS |
|---|---|---|---|
| 0.061 | 0.060 | 0.000 | 664.8 |

The theoretical peak performance is 4400 MFLOPS, far above the one measured here, since the problem is too small.

## 7.6.3 Linpack TPP

For further details on this study, refer to the following site:

http://www.netlib.org/benchmark/1000d

Linpack TPP (Toward Peak Performance), the second Linpack benchmark, is for a matrix of order 1000 by 1000. It allows vendors to implement their own linear equation solver in any language, as long as the implementation computes a solution and the result satisfies prescribed accuracy.

We used ESSL subroutines *dgef* and *dges* to replace the original *dgefa* and *dgesl* respectively. These two ESSL subroutines have SMP versions in the ESSLSMP library which contains codes that have been parallelized to use all available processors within a node. The following line compiles the code:

```
xlf_r -O3 -qarch=pwr4 -qtune=pwr4 -qsmp=noauto -lesslsmp \
-o linpackt linpackt.f
```

To run (four threads):

```
nt=4
export XLSMPOPTS="parthds=$nt:spins=0:yields=0"
export OMP_NUM_THREADS=$nt
./linpackt > $nt.out
```

The resulting GFLOPS are shown in Figure 7-3, where we also include available AIX results which are published in:

http://www-1.ibm.com/servers/eserver/pseries/hardware/system_perf.pdf

This is not an exact shoulder-to-shoulder comparison since many factors such as memory configuration, large page support, compiler options, and so on may be different. Nevertheless, it can be seen that Linux is performing almost the same as AIX.

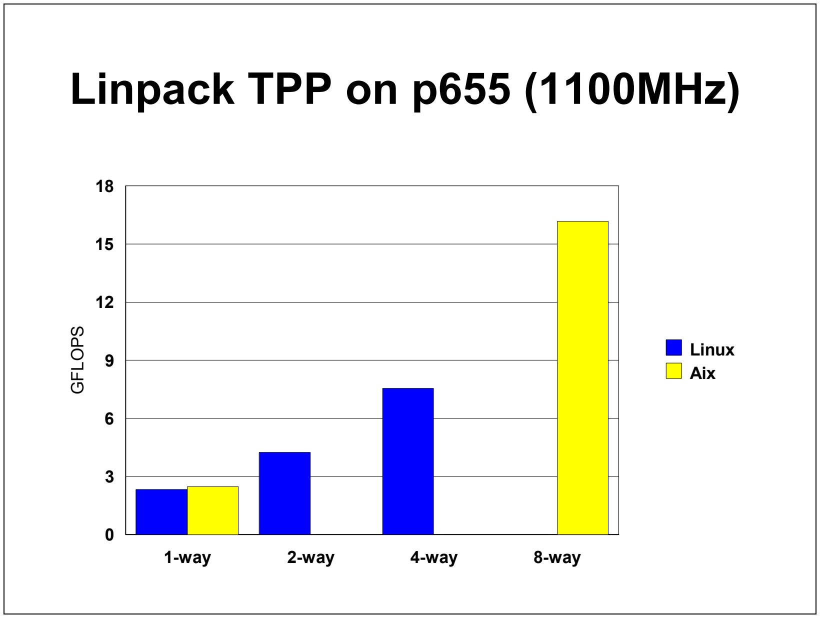


*Figure 7-3 Linpack TPP result of p655 running Linux and AIX*

Theoretical peak GFLOPS for 1.1GHz p655 are 4.4, 8.8, and 17.6 for one, two, and four processors, respectively. By comparing the actual numbers from Figure 7-3 with these peak numbers, it can be seen that the problem size (n=1000) under consideration is still not optimal to show the best performance.

In order to see how much GFLOPS can be obtained, we have tried to vary the problem size from 1100 to 30000 with an increment of 100. The results are shown in Figure 7-4 on page 322 for four-way runs.

It can be seen from Figure 7-4 that as N (the size of the matrix) increases, we get better performance overall. After around N=15000, the performance saturates at about 11GFLOPS. Actually, the best number is 10.91GFLOPS at N=16800. This is much better than Linpack TPP (N=1000), which is 7.549GFLOPS in Figure 7-3 on page 321 (the four-way result).

Another interesting observation is that the performance fluctuates even in the saturated region. So it is very important to have a dense data point to get the best result.

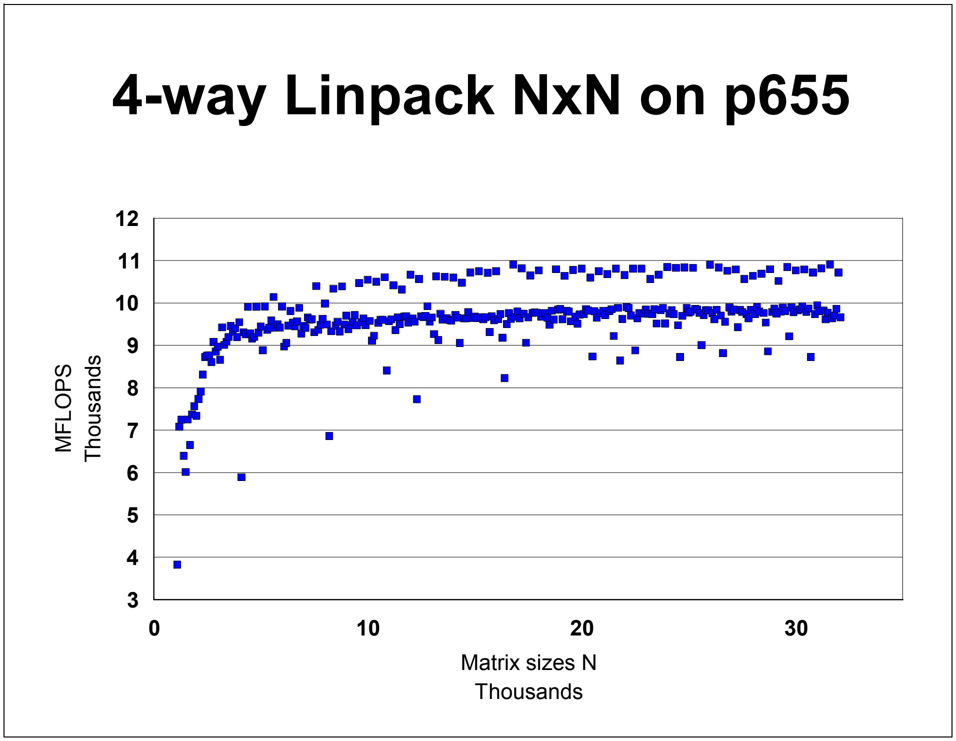


*Figure 7-4 Four-way Linpack NxN performance as N varies*

## 7.6.4 Linpack HPC

For further details on this study, refer to the following site:

http://www.netlib.org/benchmark/hpl/

Linpack HPC (Highly Parallel Computing), also known as HPL (High Performance Linpack, is the benchmark used for world's top 500 supercomputers list (top500). Unlike Linpack DP and Linpack TPP, which work only on shared memory systems, Linpack HPC works on distributed memory systems as well. Because of this, Linpack HPC requires an implementation of MPI (version 1.1-compliant). Another prerequisite is either Basic Linear Algebra Subprograms (BLAS) or Vector Signal Image Processing Package Library (VSIPL).

## About BLAS

BLAS consists of three levels.

- Level 1 performs vector-vector operations.
- Level 2 performs matrix-vector operations.
- Level 3 performs matrix-matrix operations.

BLAS is available from different sources.

- At the netlib Web site, source code and some binary builds are available for downloading:

  http://www.netlib.org/blas/

  Performance is usually lacking, compared with vendor-provided or specially optimized ones.

- The Automatically Tuned Linear Algebra Software (ATLAS) package contains BLAS and some routines from Linear Algebra PACKage (LAPACK). ATLAS uses an empirical method to tune performance. After extensive tuning, it can achieve very high performance. ATLAS is maintained at:

  http://math-atlas.sourceforge.net/

- The GOTO library, a high performance BLAS library developed by Kazushige Goto, available at:

  http://www.cs.utexas.edu/users/flame/goto/

  Only binaries for some platforms are available. This is definitely a site to check if you do not have a tuned BLAS.

- Vendor-provided; many hardware vendors (and some compiler providers) have highly tuned BLAS for their products. For IBM, it comes with ESSL libraries.

**Important:** There may be more than one copy of a BLAS library on your system and they may be from different sources. Therefore, avoid putting **`-lblas`** in your makefile without knowing which BLAS it is linking to.

For ESSL users, it is safe to replace any appearance of **`-lblas`** by **`-lessl`**. If you intend to use multi-thread versions, use this linking order: **`-lesslsmp`**.

## About MPI

Three implementations of MPI are frequently used on Linux clusters:

- Local Area Multicomputer (LAM) MPI includes a full implementation of the MPI-1 standard, as well as many elements of the MPI-2 standard. It often comes with Linux installation CDs as optional packages. LAM/MPI supports the following communications.
  - Native Myrinet message passing through GM
  - Asynchronous UDP-based message passing
  - Message passing using TCP
  - Message passing using shared memory and TCP

  LAM MPI is available from:

  `http://www.lam-mpi.org/`
- ANL (Argonne National Laboratory) MPICH:

  `http://www-unix.mcs.anl.gov/mpi/mpich`

  This is probably the most popular MPI implementation. It supports many devices and communication protocols. It can be built to suit for:
  - Single SMP box for shared memory communication only.
  - A cluster of uniprocessors for TCP/IP communication.
  - A cluster of SMP boxes for combined TCP/IP and shared memory communication
- Myricom MPICH_GM ported by Myricom from MPICH to exploit lower latency and higher data rates of Myrinet networks. It requires Myrinet hardware and device drivers.

  `http://www.myri.com/scs/index.html`

## To build

After downloading the hpl.tgz file, untar it to a directory, for example, /bench. Examine the available makefiles in directory /bench/hpl/setup/ and find the one that closely matches your system. Then copy the file to /bench/hpl/ and rename it to something like *Make.MyArch*. Modify this file appropriately to suit your system environment and then issue the command **make arch=MyArch** from /bench/hpl/. This will build an executable xhpl at /bench/hpl/bin/MyArch/, where a sample input file HPL.dat will also be available.

For pSeries running Linux, the closest match of makefiles is Make.PWR3_FBLAS, so we took this file and modified the definitions shown in Example 7-10, which uses the ESSL library.

*Example 7-10 Linpack HPC build options with ESSL*

```
ARCH = $(arch)
TOPdir = /bench/hpl
CC = /usr/bin/mpicc
CCFLAGS = -qnosmp -qarch=pwr4 -qtune=pwr4 -qcache=auto -O3
LINKER = /usr/bin/mpif77
LINKFLAGS = $(CCFLAGS) -lessl
```

Example 7-11 shows settings for the ESSLSMP library. In both cases, we used 64-bit as the application space by setting the environment variable OBJECT_MODE=64.

Note that /usr/bin/mpicc and /usr/bin/mpif77 are wrappers for the “real” mpicc and mpif77.

*Example 7-11 Linpack HPC build options with ESSLSMP*

```
ARCH = $(arch)
TOPdir = /bench/hpl
CC = /usr/bin/mpicc
CCFLAGS = -qsmp=noauto -qarch=pwr4 -qtune=pwr4 -qcache=auto -O3
LINKER = /usr/bin/mpif77
LINKFLAGS = $(CCFLAGS) -lesslsmp
```

Change of source code is normally not necessary since most of the time would be spent in the matrix-matrix operation which is handled by user-supplied performance library.

However, there is one place which you can change to see how performance changes. In line 158 of the file hpl/testing/ptest/HPL_pddriver.c, the parameter to call to HPL_grid_init is HPL_ROW_MAJOR. It can be changed to HPL_COLUMN_MAJOR. This change allows more of the MPI communication to be done within each SMP.

### The input file HPL.dat

The HPL executable xhpl only needs one input file HPL.dat to be in the current working directory. The input file contains parameters controlling the size of the problem, the number of problems per job, the algorithm features and processor grid, and so on.

Actually, one of the major challenges to get the best performance number is to try different combinations of the parameters. Example 7-12 lists the input file we use in our 8-way runs (note that it requires about 16 GB of total memory).

*Example 7-12 Linpack HPC input file HPL.dat*

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
42000
1            # of NBs
400
1            # of process grids (P x Q)
2            Ps
4            Qs
16.0         threshold
1            # of panel fact
2            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
8            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
0            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
1            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
0            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

Parameters N, NB, P, Q in Example 7-12 are of particular interest for performance tuning and are discussed below.

- N - The size of the matrix. The bigger the problem size, the bigger the ratio of computing time to communication time and therefore normally the better the performance. However, the problem cannot be too big to cause system paging.

A rule of thumb is to have a problem size which is 80% of the total memory of all nodes (assuming a uniform amount of memory on each node). So you can start with the N given by the following formula:

N=sqrt(0.8*Total_Mem_in_Bytes/8)

Then vary it to see how performance changes.

- NB - The block size of the matrix used by HPL for data distribution and computational granularity. It is better to be a small multiple of the block size used by matrix-matrix multiply routine (*dgemm*). Again, an optimal one should be big enough for better data reuse and small enough for better load balance. Normally NB should be bigger for bigger N.
- P, Q - These two parameters determine processor grid. P*Q=Number of MPI processes. Start with a “square” grid and then decrease P (increase Q so P*Q is constant) to see how performance changes.

**Note:** There is another layer of tuning if hybrid MPI/OpenMP is used—the combination of the number of MPI processes and the number of OpenMP threads.

In this case, P*Q*T=total number of processors, where T is the number of OpenMP threads per MPI process. For example, for a system of two nodes each with four processors, you can also run with P=Q=T=2, in addition to P=2,Q=4,T=1. Hybrid MPI/OpenMP has the advantage of achieving better balance between computing time and communication time.

### To run

Copy files xhpl and HPL.dat to a directory. Then the rest will depend on your hardware configurations and MPI packages. We give examples for combinations of Gigabit Ethernet, Myrinet GM, pure MPI, hybrid MPI/OpenMP.

Note that in the following examples (Example 7-13 to Example 7-16), we assume the following:

- We have two compute nodes whose Myrinet interfaces are lp01 and lp02, and whose Gigabit interfaces are lg01 and lg02.
- Four processors on each are used in all examples.

We have put large values (in bytes) for two important MPICH environment variables, P4_GLOBMEMSIZE and P4-SOCKBUFSIZE. Their default values are 4 MB and 16 KB, respectively. P4_GLOBMEMSIZE specifies the amount of memory reserved for communication with shared memory (when MPICH has been configured with the -comm=shared option). The job hangs if the value is too low. This is a dynamic run-time behavior and you need to set it appropriately according to the communication size of the running job.

P4_SOCKBUFSIZE sets the socket buffer size in bytes. Increasing this value usually improves performance.

### Pure MPICH_GM over GM

*Example 7-13 HPL run script for pure MPICH_GM over GM*

```
#!/bin/bash
cat > host.list << EOF
lp01
lp01
lp01
lp01
lp02
lp02
lp02
lp02
EOF
mpirun_ch.gm -np 8 -machinefile host.list xhpl > log.log
```

### Hybrid MPICH_GM/OpenMP over GM

*Example 7-14 HPL run script for hybrid MPICH_GM/OpenMP over Myrinet*

```
#!/bin/bash
cat > host.list << EOF
lp01
lp01
lp02
lp02
EOF
mpirun_ch.gm OMP_NUM_THREADS=2 -np 4 -machinefile host.list xhpl > log.log
```

### Pure MPICH over IP (Gigabit Ethernet)

*Example 7-15 HPL run script for pure MPICH over IP (GigE)*

```
#!/bin/bash
cat > host.list << EOF
lg01:4
lg02:4
EOF
cat >> ~/.bashrc << EOF
P4_GLOBMEMSIZE=660000000
P4_SOCKBUFSIZE=512000
export P4_GLOBMEMSIZE P4_SOCKBUFSIZE
EOF
mpirun -nolocal -np 8 -machinefile host.list xhpl > log.log
```

### *Hybrid MPICH/OpenMP over IP (Gigabit Ethernet)*

*Example 7-16 HPL run script for hybrid MPICH/OpenMP over GigE*

```
#!/bin/bash
cat > host.list << EOF
lg01:2
lg02:2
EOF
cat >> ~/.bashrc << EOF
OMP_NUM_THREADS=2
P4_GLOBMEMSIZE=660000000
P4_SOCKBUFSIZE=512000
export OMP_NUM_THREADS P4_GLOBMEMSIZE P4_SOCKBUFSIZE
EOF
mpirun -nolocal -np 4-machinefile host.list xhpl > log.log
```

Notes on Example 7-13 to Example 7-16:

- Myrinet MPICH_GM always uses shared memory for intra-node communication by default, and it passes environment variables to other MPI processes by command line options.
- The **`-nolocal`** option and the format *lp01:2* of the host file for MPICH is critical to make sure shared memory communication is used for intra-node traffic. In order to have this capability, the MPICH has to have been configured with options: **"`--with-device=ch_p4 --with-comm=shared`"**
- Those two P4 environment variables are important to run large case and to run faster. The values can, of course, be changed.

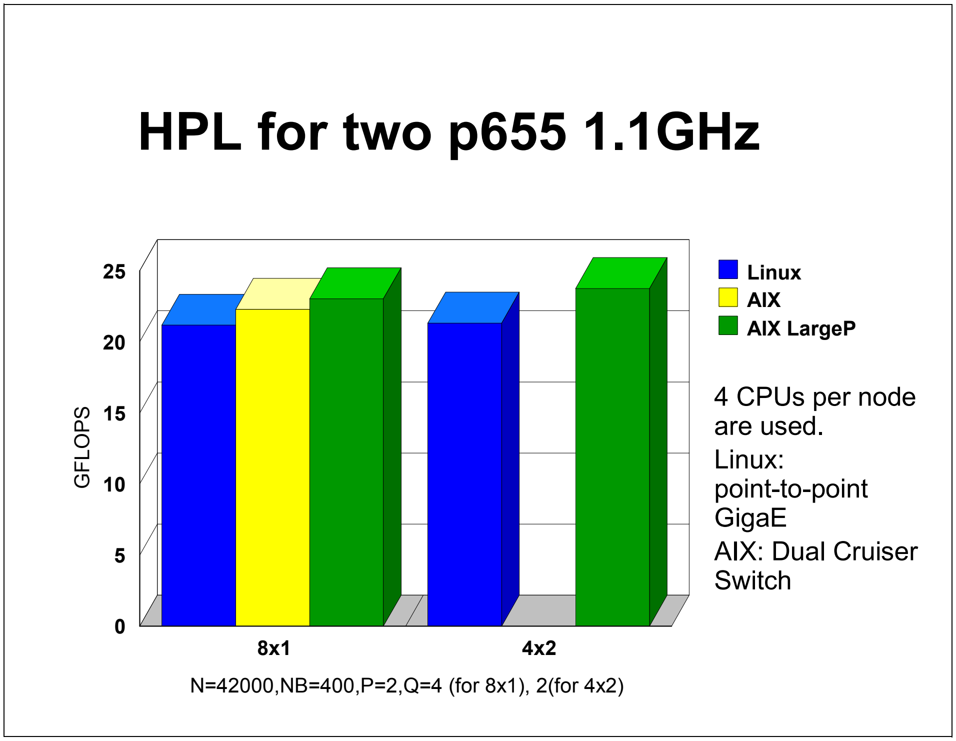
## The result

Figure 7-5 on page 330 shows an AIX and Linux comparison of HPL results on p655. Note that this is for illustration purposes only. More effort should be made to get the best performance in both cases.

We can see from Figure 7-5 that AIX has slightly better performance, especially if the AIX large page is enabled. (In addition to the standard 4 KB paging space, AIX 5.1 also supports a large paging space which is 16 MB per memory page. HPC applications that involve large sequential access of memory and files will benefit from the large paging space.)

The hybrid MPI/OpenMP (4x2 means 4 MPI processes, each with 2 OpenMP threads) performance better than pure MPI (the 8x1 case in Figure 7-5).

In addition to the standard 4 KB paging space, AIX 5.1 also supports a large paging space which is 16 MB per memory page. HPC applications that involve large sequential access of memory and files will benefit from the large paging space.


HPL for two p655 1.1GHz

4 CPUs per node are used.
Linux: point-to-point GigaE
AIX: Dual Cruiser Switch

*Figure 7-5 AIX and Linux HPL results for two p655 at N=42000*

Figure 7-6 on page 331 shows comparisons of MPICH over Gigabit Ethernet and MPICH_GM over Myrinet in the smaller case (N=7000). It can clearly be seen that Myrinet performs better than Gigabit Ethernet.

However, keep in mind that this is a very small problem (only takes about 15 seconds), so the communication time is relatively more important than with large problems, where computational time becomes dominant.

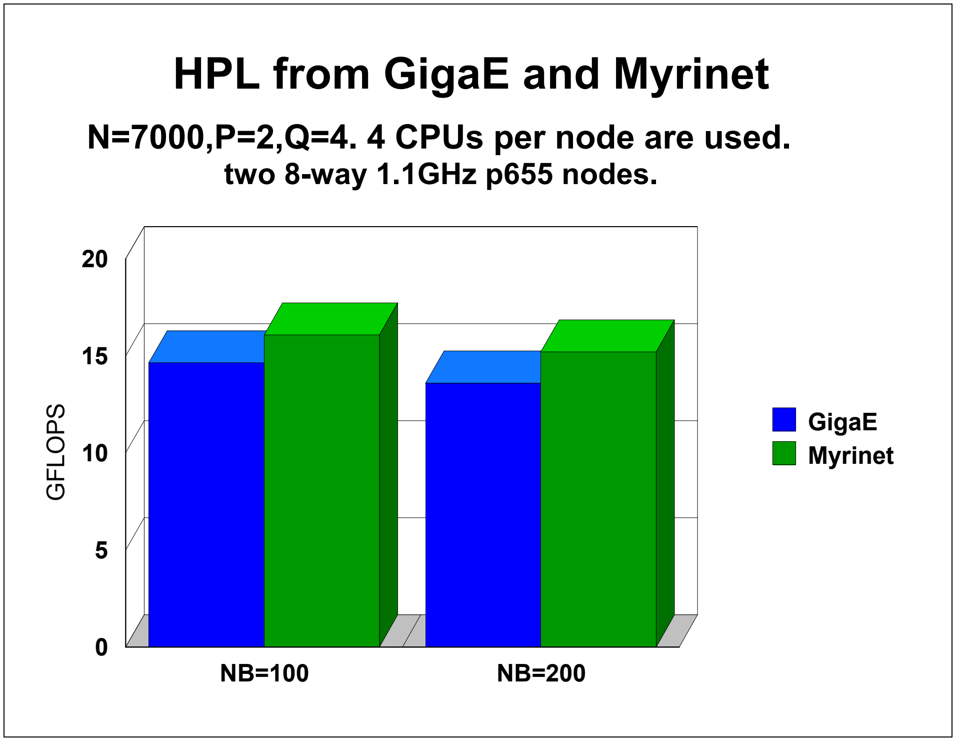


*Figure 7-6 Eight-way HPL results on two p655 nodes: GigE and Myrinet comparison*

### 7.6.5 NetPIPE - point-to-point communication

For further details on this study, refer to the following site:

`http://www.scl.ameslab.gov/netpipe/`

Network Protocol Independent Performance Evaluator (NetPIPE) measures point-to-point communication between two processes. Message sizes are increased in regular intervals and also varied with small perturbations in order to see communication throughput.

NetPIPE is designed to handle many different protocols and communication middleware such as MPI, MPI-2, PVM, SHMEM, TCP, GM, LAPI, and so on. Here we show the usage and performance of two modules, TCP and MPI, under different situations.

## To build

Building executables for TCP and MPI modules is very straightforward. No change to source files and makefile is necessary if compilers and MPI are on the system PATH. Simply execute these commands and the executables NPtcp and NPmpi will be created:

```
make tcp
```

```
make mpi
```

## TCP across nodes - Gigabit Ethernet

Figure 7-7 shows NetPIPE TCP communication throughput over Gigabit Ethernet connection. Two curves are shown, one with default parameters, and the other with a buffer size of 256 KB, which is set through the command line option **`NPtcp -b 262144`**. The default case corresponds to a buffer size of 16 KB.

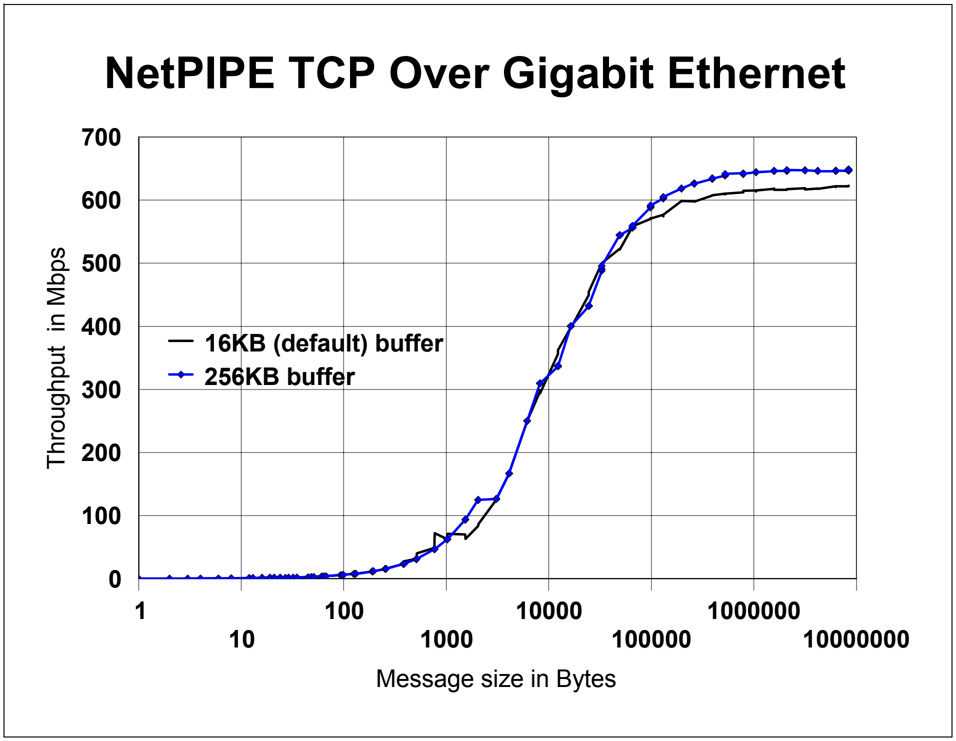


*Figure 7-7 NetPIPE TCP on two p655 nodes connected by Gigabit Ethernet*

It can be seen that the throughput increases as the message size increases, reaching the top at around 1 MB in both cases. The one with the 256 KB buffer has higher bandwidths for large messages.

### TCP within a node

To see how TCP performs within an SMP box, we also run the same jobs on a single p655 node; the results are shown in Figure 7-8. Much higher bandwidths are achieved than in the cross-node communication case (Figure 7-7 on page 332).

Again, comparing the two curves in this figure, you can see that the default buffer size of 16 KB does not yield the best performance for large messages.

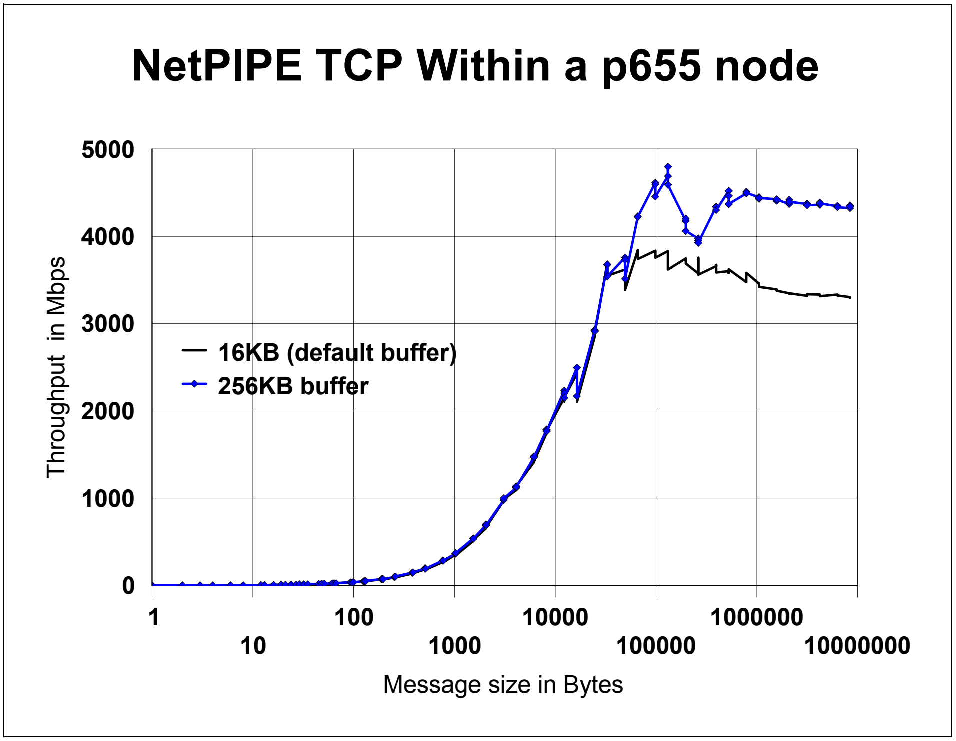


*Figure 7-8 NetPIPE TCP within a single p655 node*

### MPI across nodes - GigE and Myrinet

Figure 7-9 on page 334 shows three cases of NetPIPE MPI communications across two p655 nodes.

- MPICH over Gigabit Ethernet for a default buffer size of 16 KB.
- MPICH over Gigabit Ethernet for a buffer size of 256 KB.
- MPICH_GM over Myrinet. In this case, the buffer size parameter does not have any effect because it runs in “user space” mode, not IP mode.

In this case, the change of buffer size helped improve bandwidth from 394Mbps to 596Mbps for MPICH, a 51% improvement. This is much larger than in the TCP case (Figure 7-7 on page 332), where the bandwidth improvement for inter-node communication is (648-623)/623=4%.

A more impressive observation is the large gap between MPICH_GM over Myrinet and MPICH over Gigabit Ethernet. MPICH_GM/Myrinet is able to deliver 1691Mbps, while the “best” that MPICH/Gigabit can reach 596Mbps.

By comparing Figure 7-7 on page 332 with the MPICH part of Figure 7-9 on page 334, we can also see how much overhead MPICH has brought to the communication; the maximum bandwidth is 648Mbps for TCP and 596Mbps for MPICH.

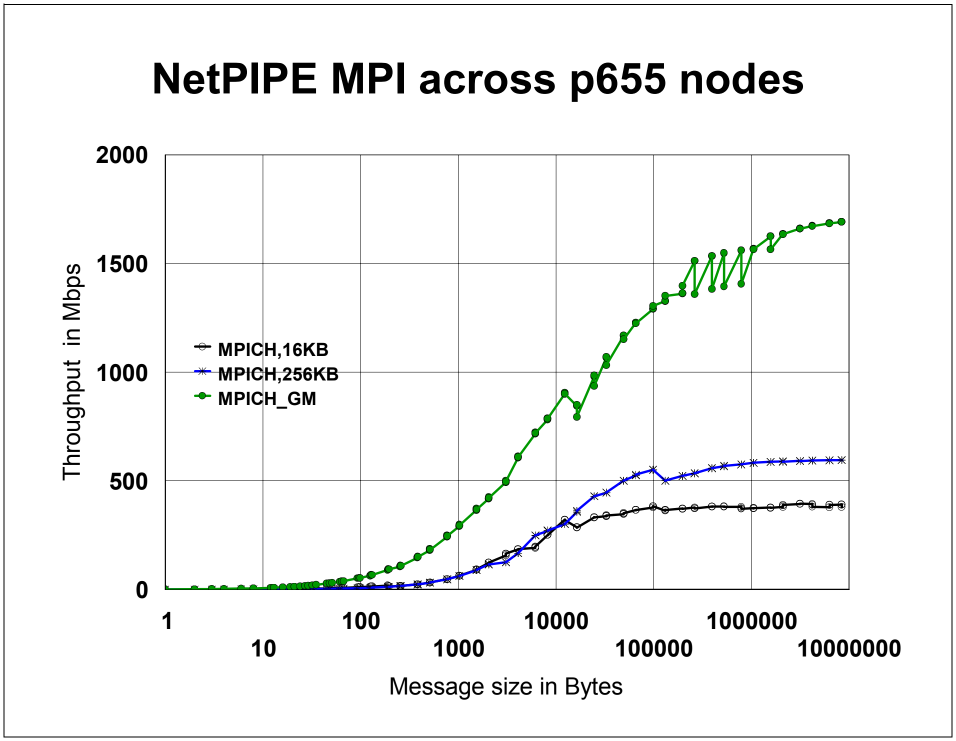


*Figure 7-9 NetPIPE MPI across p655 nodes - GigE and Myrinet comparison*

## Shared memory MPI within an SMP node

We now take a look at shared memory communication within an SMP box, which is an eight-way p655, in our case.

**Note:** NetPIPE has a separate module to do the *shmem* test for Cray systems. It is not the same as what is discussed here.

MPICH_GM (version 1.2.5..10) has shared memory support for communications within SMP nodes by using a device *ch_gm*. For MPICH, there are two implementations of shared memory support, one with the device of *ch_shmem* (configured with option **`--with-device=ch_shmem`**) and the other with device *ch_p4* and *comm* being shared (**`--with-device=ch_p4 --with-comm=shared`**) The former only works with a single SMP node, the latter works with a cluster of SMP nodes.

Figure 7-10 compares these three cases. We have chosen the buffer size to be 256 KB for the MPICH cases. It is interesting to note that MPICH/ch_shmem has the highest peak in all three cases, but MPICH_GM has better performance outside that peak area.

It is also worth noting that MPICH/ch_shmem outperforms MPICH/ch_p4 for almost all sizes of messages. This suggests that multiple MPICHs be built on a system of clustered SMP nodes for performance reasons.

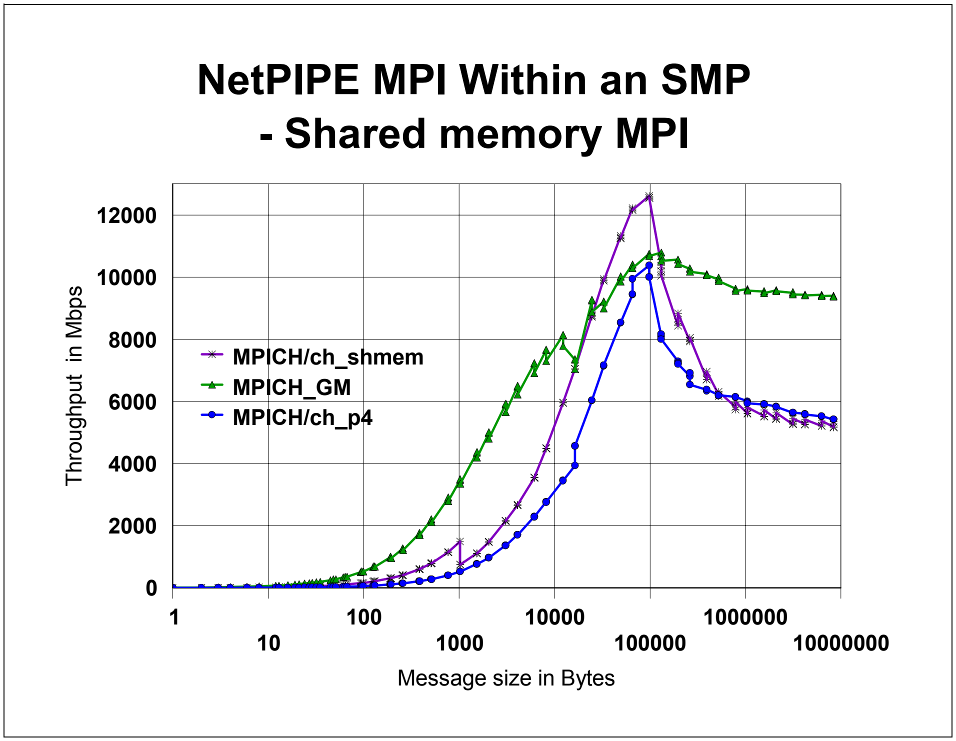


*Figure 7-10 NetPIPE MPI within an SMP - shared memory MPI*

### MPI within an SMP node - without shared memory

It is very unusual to run MPI without shared memory on a single SMP node or on a cluster of SMP nodes. However, it is still interesting to see what happens if we switch shared memory off.

Figure 7-11 illustrates such a scenario; it compares MPICH (different buffer sizes) and MPICH_GM. Surprisingly, MPICH_GM shows very bad performance, even compared with the default 16 KB buffer MPICH result.

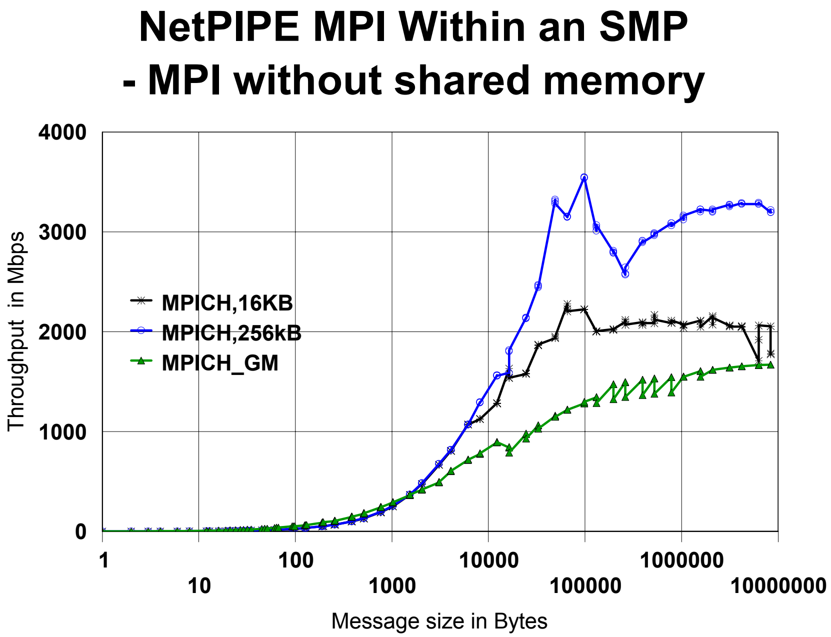


*Figure 7-11 NetPIPE MPI within an SMP - MPI without shared memory*

### Summary

So far, we have discussed TCP and MPI performance under various situations for different message sizes. To make it easier to understand communication performance differences, and therefore make better use of a system, we now provide a simplified version of performance comparisons of the maximal bandwidths of each case.

### TCP

Figure 7-12 shows maximal bandwidths of NetPIPE TCP for intra-node and inter-node communications. This gives an idea of how much difference there is between intra-node and inter-node communication. It also shows the effect of message buffer size.

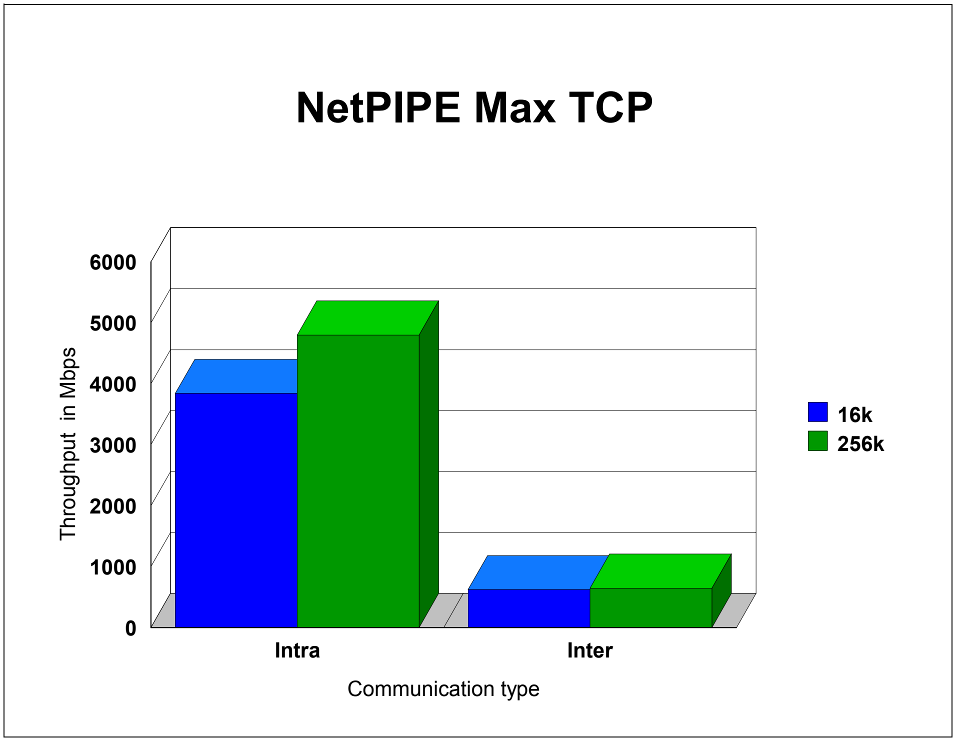


*Figure 7-12 Maximal bandwidth of NetPIPE TCP*

### MPI

Figure 7-13 on page 338 shows maximal bandwidths of NetPIPE MPI for intra-node and inter-node communications. You can see how communication bandwidths change from inter-node to intra-node non-shared, and from intra-node non-shared to intra-node shared memory.

MPICH with ch_shmem device gives the best peak bandwidth within an SMP node. MPICH_GM has the best performance for cross-node communications (it uses Myrinet adapters, not Gigabit adapters).

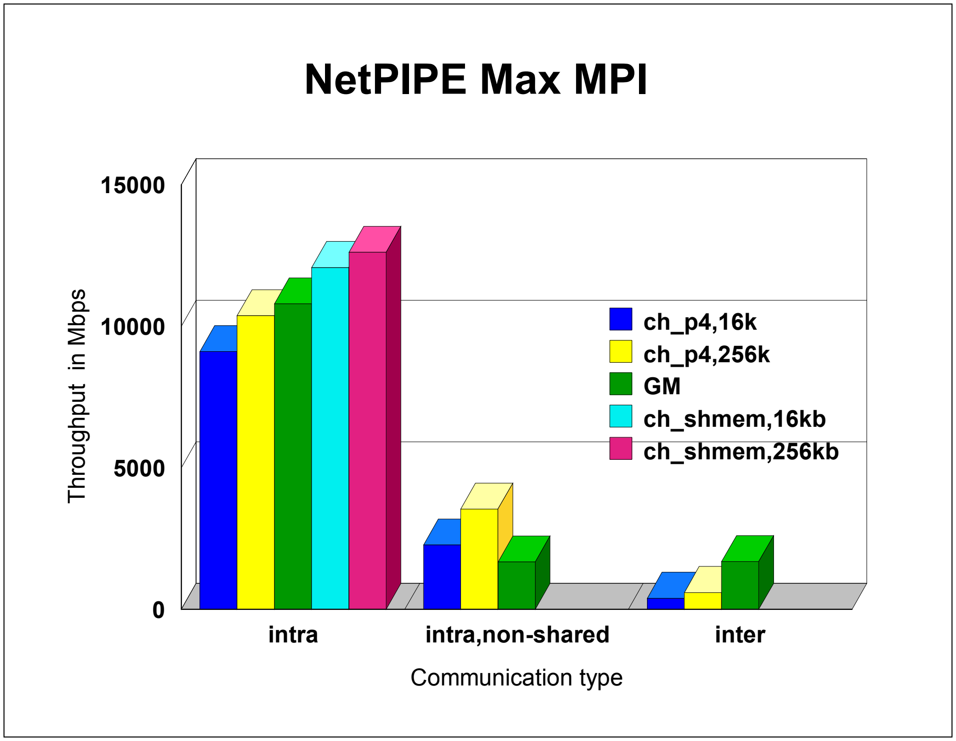


*Figure 7-13 Maximal bandwidth of NetPIPE MPI*

## 7.6.6 Pallas PMB - more on MPI

For further details on this subject, refer to the following site:

`http://www.pallas.com/e/products/pmb`

The Pallas benchmark suite is a collection of functions that measure MPI performances of various communication patterns such as one-directional point-to-point, bidirectional point-to-point, cyclic sendreceive and exchange, and collective communications.

Point-to-point communication is covered in 7.6.5, “NetPIPE - point-to-point communication” on page 331, so in this section, we show results for collective communications. Whenever possible, we make comparisons among GigaE, Myrinet, and the IBM Cruiser switch.

### To build

The version we built for testing is PMB2.2.1 and the package we used is MPI-1. Only minimal modification was done to the makefiles, and the command to make the binaries we needed is **make PMB-EXT**. There are other components, such as MPI-IO, but we only tested PMB-EXT.

## To run

The publication *Pallas MPI Benchmarks - PMB, Part MPI-1*, which comes with the tar file, describes in detail of what PMB can do and how to run the code. For our purposes, we used the following command lines to run the binary.

- For GigE:

```
mpirun -nolocal -np <np> -machinefile host.list PMB-MPI1 -npmin <np>
```

- For Myrinet:

```
mpirun.ch_gm -np <np> -machinefile host.list PMB-MPI1 -npmin <np>
```

- For AIX:

```
poe -nprocs <np> PMB-MPI1 -npmin <np>
```

**Note:** The *host.list* is different for those runs, and extra environment variables such as P4_GLOBMEMSIZE, P4_SOCKBUFSIZE for GigE runs and POE environment variables for AIX runs have to be set appropriately. <np> is the number of MPI processes to start.

## Allreduce

Figure 7-14 on page 340 shows a comparison of eight-process Allreduce among GigE (Linux), Myrinet (Linux), and Cruiser (AIX) on two p655 nodes. Both X and Y axes are in logarithmic scales in order to see full range of curves.

The first data point (corresponding to zero-sized message) for GigE and Myrinet is probably not accurate and should be discarded. It can be seen that GigE is the worst for all message sizes. Myrinet performs better than Cruiser for small messages. For large messages (>16 KB), Cruiser outperforms Myrinet.

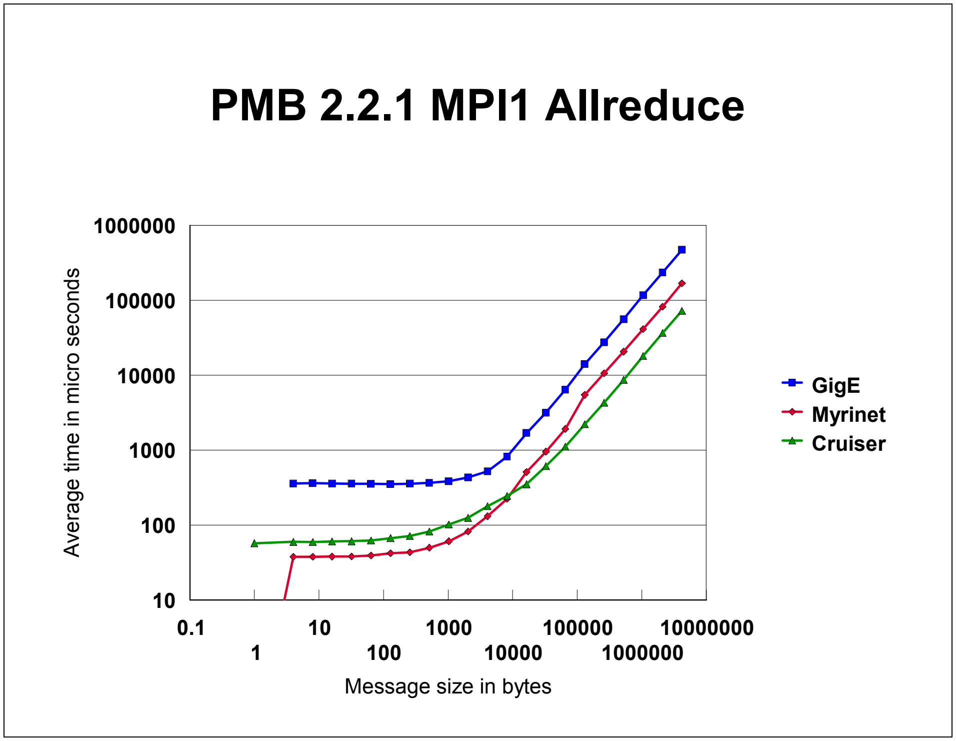


*Figure 7-14 PMB eight-way Allreduce on two p655 nodes*

Figure 7-15 on page 341 shows the same picture in linear scales for both X and Y axes. It shows that the communication time increases linearly with message size in all three cases. However, it hides the details for small messages, as we see in Figure 7-14.

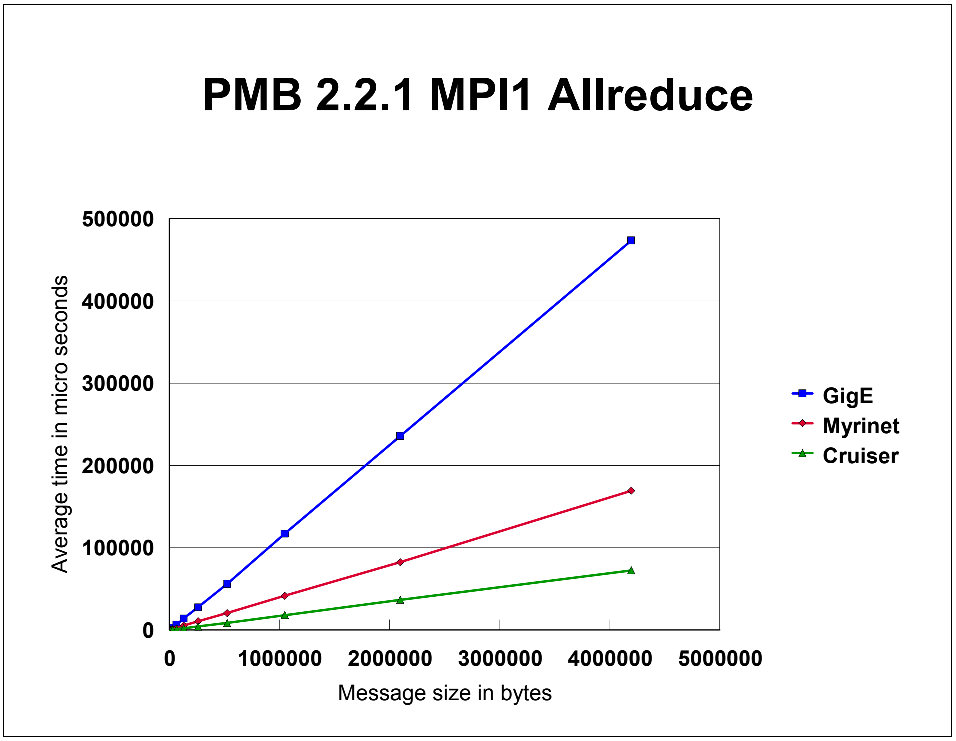


*Figure 7-15 Same as Figure 7-14, with X and Y in linear scales*

## Reduce

Figure 7-16 on page 342 shows Reduce results of the same MPI job (eight-way across two p655 nodes). The performance comparison for the three cases is the same as Allreduce; even the turning point at which Cruiser outperforms Myrinet is the same.

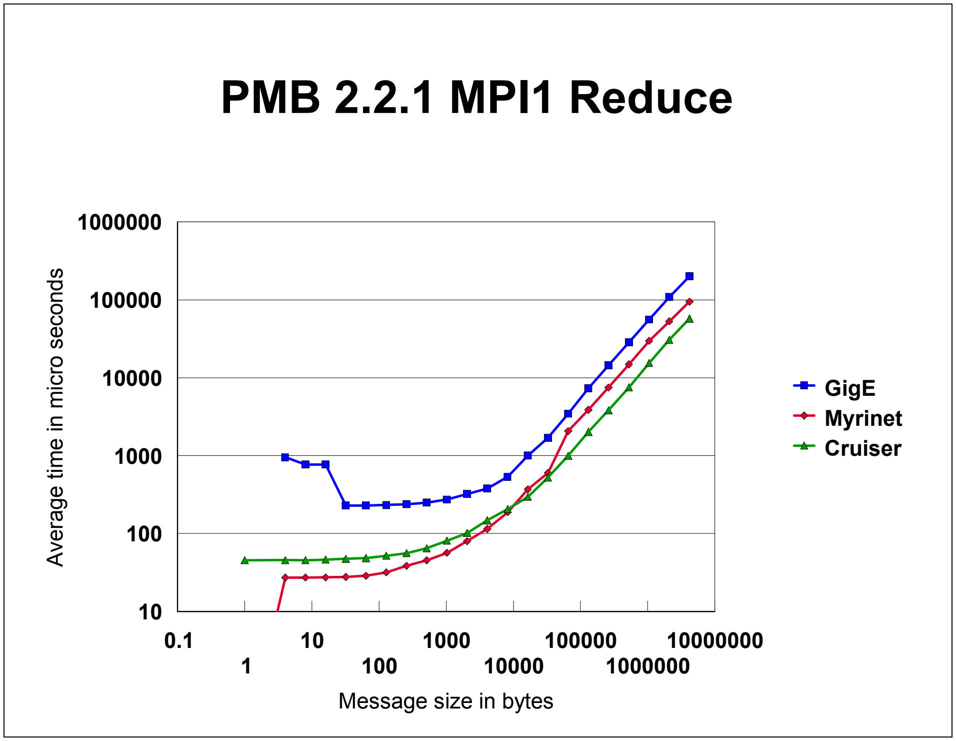


*Figure 7-16 PMB eight-way Reduce on two p655 nodes*

## Reduce_scatter

The MPI collective communication Reduce_scatter shows a slightly different picture (see Figure 7-17 on page 343); the Myrinet and Cruiser results are closer, compared with Allreduce and Reduce. The Cruiser result is better than Myrinet for sizes of messages.

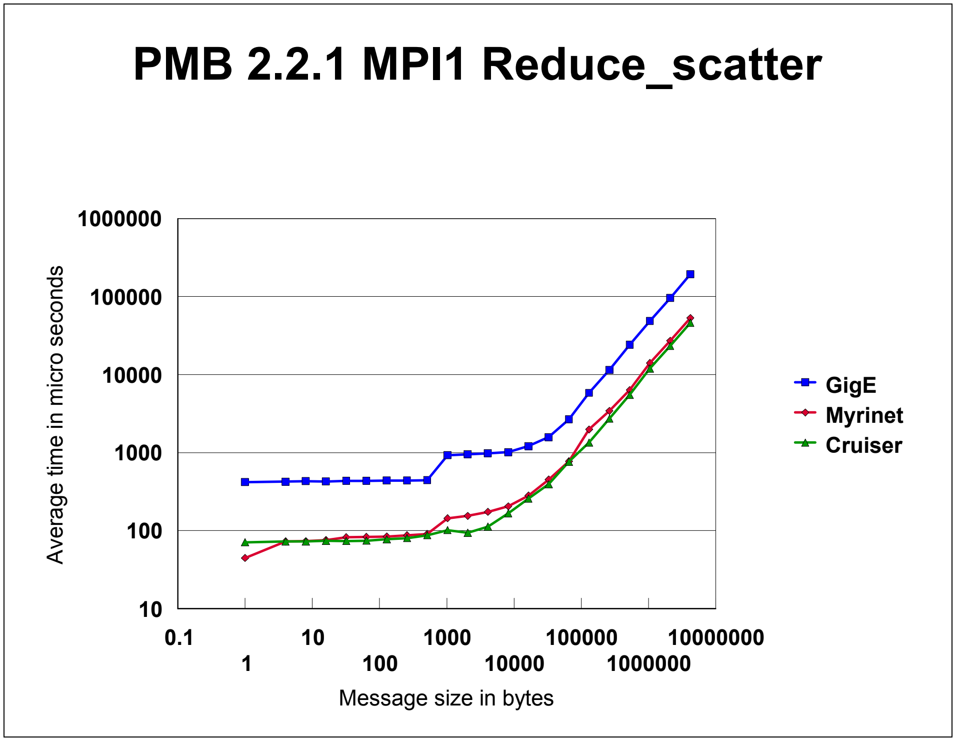


*Figure 7-17 PMB eight-way Reduce_scatter on two p655 nodes*

## Allgather

Figure 7-18 on page 344 shows Allgather performance. Myrinet has a clear advantage for small messages. For larger messages (>4KB), these two are very close to each other. However, you can still see that Myrinet performs slightly better.

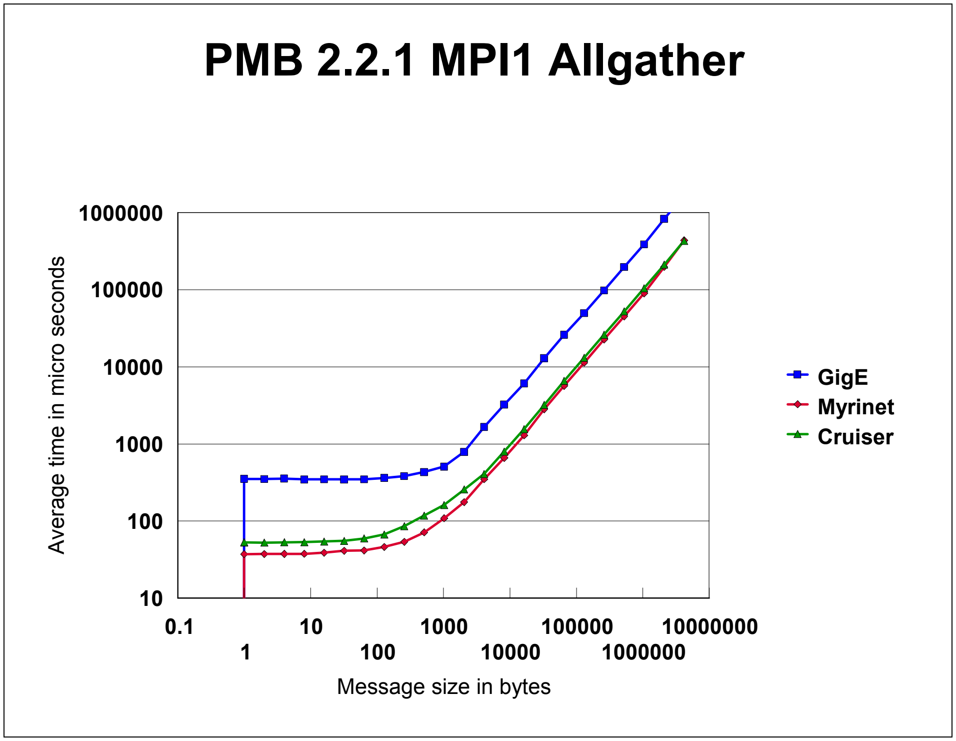


*Figure 7-18 PMB 8-way Allgather on two p655 nodes*

## Allgatherv

The overall performance trend for different sizes of messages is the same as Allreduce; that is, GigE is the worst, Myrinet is better than Cruiser for smaller messages, but worse than Cruiser for larger messages.

However, the turning point is much smaller: 512 B. Another thing about Myrinet and GigE is that the performance drops a great deal for message sizes of 8 KB, 16 KB, and 32 KB.

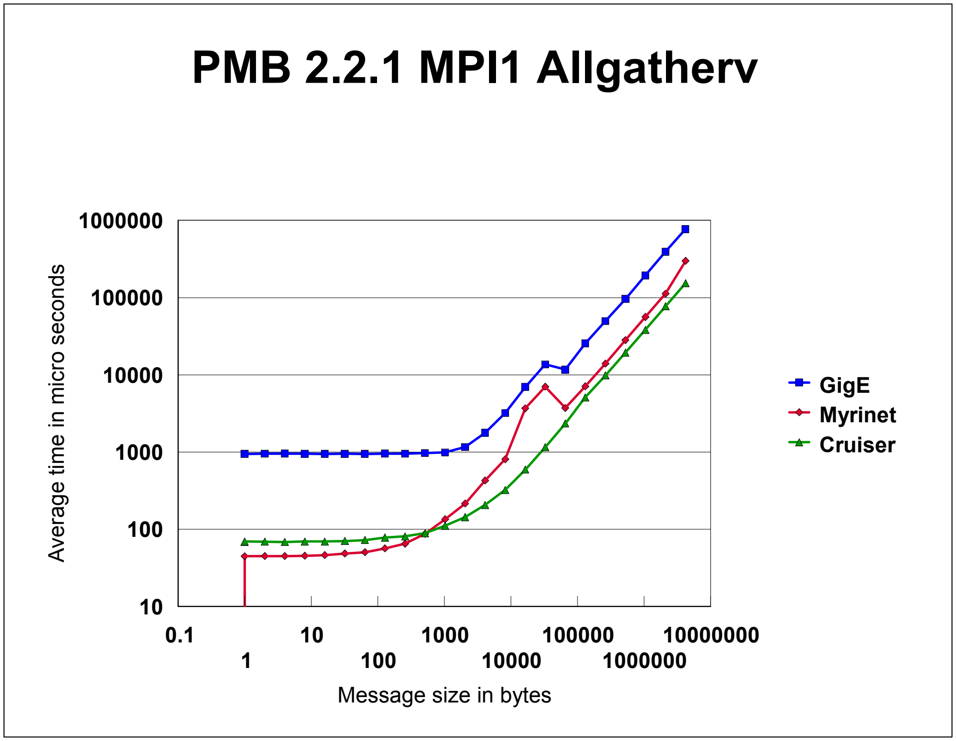


*Figure 7-19 PMB eight-way Allgatherv on two p655 nodes*

## Alltoall

Figure 7-20 on page 346 shows Alltoall performance. For large messages, Cruiser has slightly better performance than Myrinet. For small messages up to 16 KB, Myrinet and Cruiser performances are crossed to each other. The GigE performance favors message sizes of 32 B to 1 KB.

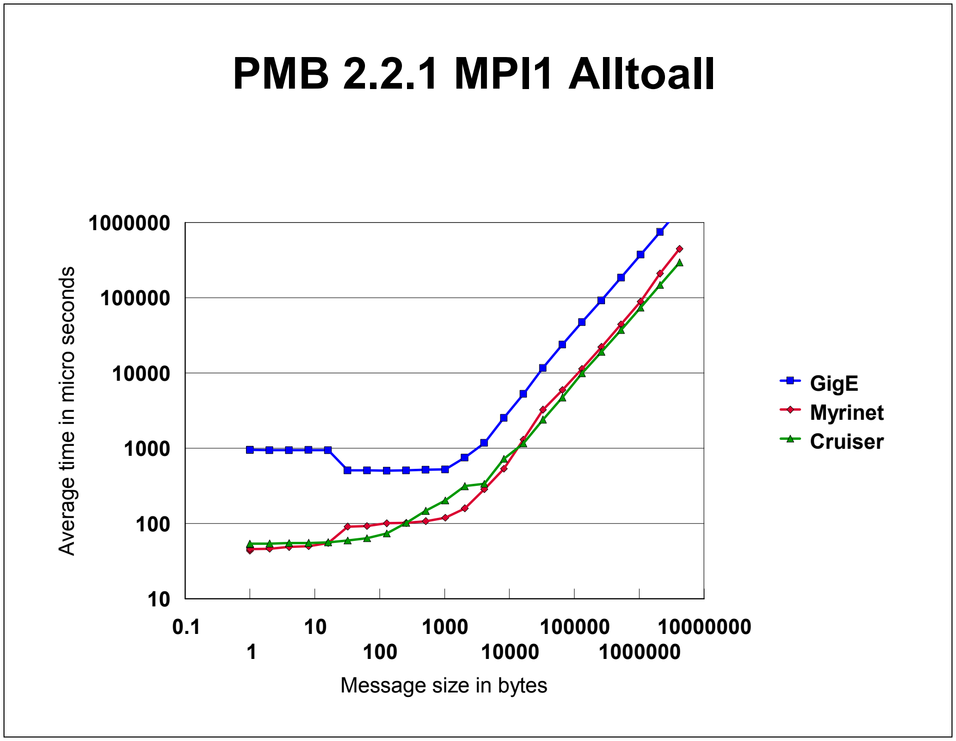


*Figure 7-20 PMB eight-way Alltoall on two p655 nodes*

## Bcast

The Bcast results are shown in Figure 7-21 on page 347. It looks from the figure that performances of all three cases have a sudden change around 8 KB or 16 KB. After that, the curves become straight. Again, Cruiser is the best one, and the GigE is the worst.

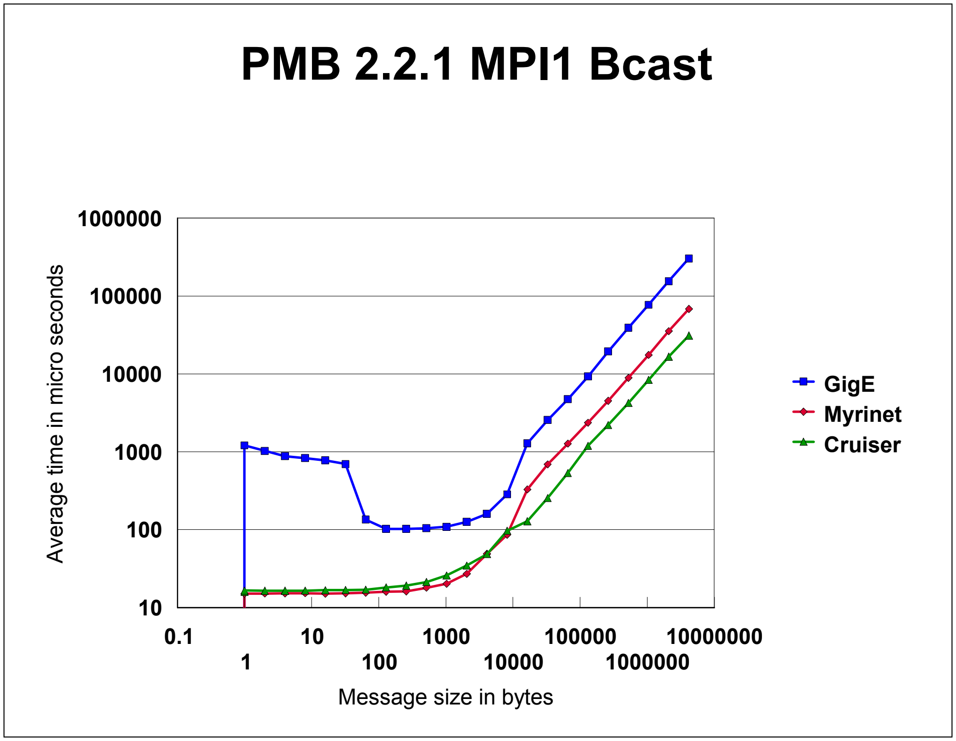


*Figure 7-21 PMB eight-way Bcast on two p655 nodes*

## 7.6.7 NAS Parallel benchmarks

For further details on this subject, refer to the following site:

```
http://www.nas.nasa.gov/Software/NPB
```

Numerical Aerodynamic Simulation (NAS) Parallel Benchmarks (NPB) are from NASA, Ames, and have been very popular in measuring parallel system performance for a number of years. These benchmarks include some of the widely used parallel algorithms in computational fluid dynamics.

NPB 2.4 consists of eight benchmark programs with several problem sizes (class A, B, C, and D):

- Class C and D are tailored for systems with more than 16 CPUs.
- Class A problems are quite small for many present-day computer systems; therefore, only class B results are presented here.

The first five of these benchmarks are *kernel* benchmarks with simple data structures. In addition to these kernels, there are three simulated applications. All of these codes are parallelized using message passing interface (MPI), and except for the Integer Sort kernel benchmark that is written in the C language, all of these are written in FORTRAN language.

## Kernel benchmarks

### *Embarrassingly Parallel (EP) problem*

The Embarrassingly Parallel problem is to generate pairs of Gaussian derivatives (also called *two independent normally distributed variables*) according to a specific scheme, and tabulate the number of pairs in successive square annuli. This problem is typical in many "Monte-Carlo" applications, and it requires very little communication between processors, as the name of the benchmark suggests.

### *The MultiGrid (MG) problem*

The MultiGrid kernel uses the V-cycle multigrid algorithms to solve approximate solution of a 3-D Poisson equation with periodic boundary conditions. Multigrid methods are used in many CFD simulations, since it improves the convergence characteristics. This kernel requires highly structured interprocessor communication.

### *The Conjugate Gradient (CG) problem*

The Conjugate Gradient benchmark computes an iterative approximation to the smallest eigenvalue of a large sparse, symmetric positive definite matrix. Within the iteration loop, the primary procedure is to solve the linear system of equations with the conjugate gradient method. This kernel is typical in unstructured grid computations, and it tests irregular long distance communication.

### *The Fast Fourier Transform (FT) problem*

The Fast Fourier Transform benchmark solves 3-D Poisson partial differential equations using 3-D forward and inverse Fourier transforms. The FFTs, key parts of turbulence simulations (large eddy simulations), use spectral methods and test rigorous long distance interprocessor communications.

### *The Integer Sort (IS) problem*

Integer Sort problem sorts keys into a number of buckets. A typical application of this method is particle in cell applications in fluid mechanics. This kernel stresses both integer computing performance and interprocessor communication.

## Pseudo Application Benchmarks

### *Block Tridiagonal (BT) problem*

This solves 3-D Navier-Stokes/Euler equations (a system of five partial differential equations) numerically using the approximate factorization scheme on a structured finite-volume mesh. In each of the three sweeps (for three directions), multiple independent systems of block tridiagonal equations (each block being a 5x5 matrix) are solved.

This method is used in many of the existing NASA flow solvers such as ARC3D, CFL3D, and OVERFLOW. This parallel implementation requires a square grid of processors.

### *Scalar Pentadiagonal (SP) problem*

This solves 3-D Navier-Stokes/Euler equations (a system of five partial differential equations) numerically using the approximate factorization scheme on a structured finite-volume mesh. In each of the three sweeps (for three directions), multiple independent systems scalar pentadiagonal equations are solved.

This method is also used in many NASA compressible flow solvers like ARC3D, CFL3D, and OVERFLOW. Like the BT problem, this parallel implementation requires a square grid of processors.

### *Lower-Upper Diagonal (LU) problem*

This solves 3-D Navier-Stokes/Euler equations numerically with the successive over relaxation procedure that involves the LU decomposition method. This algorithm is used in incompressible flow codes such as the NASA INS3D code.

## Benchmark system

The details of the hardware and software configurations are described in detail in the earlier sections of this chapter. All of these benchmarks were run from 1 through 8 CPUs (BT and SP require square grid of processors, and could not be run on 8 CPUs). Up to the 4 CPU cases, only one node was used. For the 8 CPU case, 4 CPUs from each node were used.

## To build

The NAS benchmark tar file NAS2[1].4.tar can be untarred in the benchmark directory, for example, /bench1. In the /bench1/NPB2.4/NPB2.4-MPI directory, there are several directories including the config directory.

In the config directory, there are two files that needs to be edited appropriately for setting up this benchmark: the make.def file, and the suite.def file.

In the make.def file are set the compilers to be used, the compiler flags, the linker flags, and the binary directory. For each problem, and each number of processors, a separate binary needs to be built.

A typical make.def file will have the settings shown in Example 7-17.

*Example 7-17 Typical makefile (make.def) settings for NAS benchmarks*

```
MPIF77 = mpif77
FLINK = mpif77
FFLAGS = -O3 –qarch-pwr4 –qtune=pwr4 –q64
FLINKFLAGS = -q64
MPICC = mpicc
CLINK = mpicc
CFLAGS = -O3 –q64 –qarch=pwr4 –qtune=pwr4
BINDIR = ../bin
```

The file is set to build 64-bit binaries, and put the binaries in the ../bin directory. In addition, the config directory has a file called suite.def that needs to be set for the binaries to be built.

Example 7-18 shows a suite.def  file which will build all eight applications for class B for 4 CPUs.

*Example 7-18 The suite definition (suite.def) file for NAS benchmarks*

```
# This is a sample suite config file that
# builds the class B benchmark for the entire suite for 4 CPUs

bt     B    4
sp     B    4
lu     B    4
cg     B    4
ep     B    4
is     B    4
ft     B    4
mg     B    4
```

Properly creating these two files (make.def and suite.def ) in the config directory, and issuing the following commands at the upper level directory (that is, NPB2.4-MPI) builds all eight executables:

```
make clean
make suite
```

The **`make clean`** step is important, since there may be leftover object files from some earlier build sessions with a different problem size or for different number of processes, that may result in inconsistent binaries.

## To run

Set the necessary environmental variables for Myrinet or the Gigabit Ethernet, the OBJECT_MODE (64), a host.list file with the node names appropriate for Myrinet or Gigabit Ethernet, as described in the earlier sections.

The P4_GLOBMEMSIZE environmental variable had to be set to a large value to run these benchmarks. A typical **mpirun** command, such as the following, runs the four-processor job of the application LU and creates an output file called out_lu.B.4:

```
mpirun -np 4 -machinefile host.list ../bin/lu.B.4 > out_lu.B.4
```

Each of the output files will include the verification of the results (SUCCESSFUL or FAILURE), elapsed time in seconds, total MOP/s (millions of operations per second, and MOP/s/process), the compiler, and link options that were used in the binary creation.

## The results

The results are presented as bar charts comparing Myrinet and Gigabit Ethernet interface performance. In these charts, performance is measured in terms of total MOP/s, and higher is better.

Figure 7-22 on page 352 shows results for the EP benchmark. As expected, there is no difference in performance between Myrinet and the Gigabit Ethernet interfaces for this case, because there is very little communication.

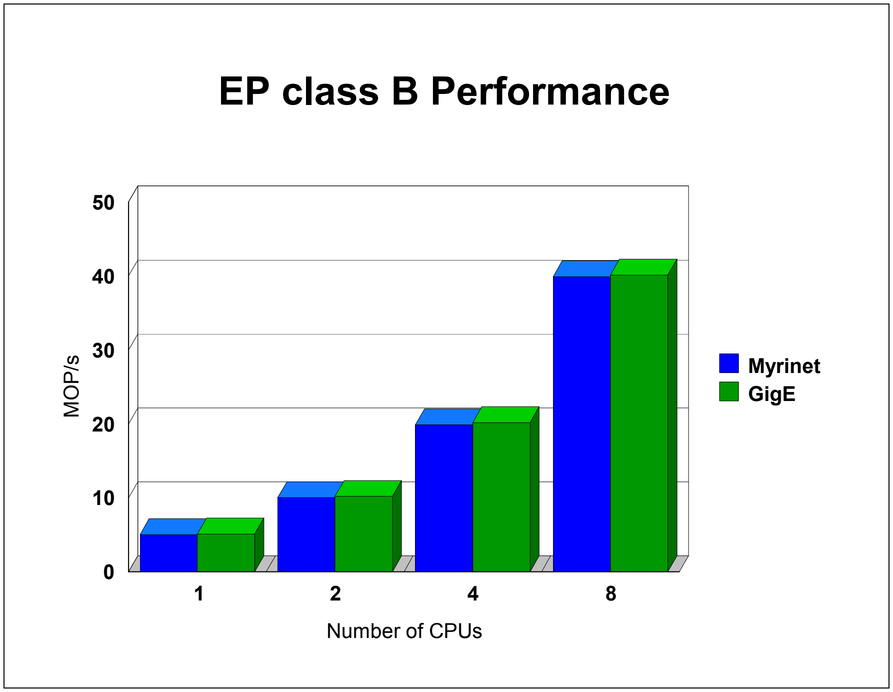


*Figure 7-22 NAS EP class B performance*

MG class B performance is shown in Figure 7-23 on page 353. For the eight-CPU case, the Myrinet interface is about 30% faster than the Gigabit Ethernet interface, since inter-node communication becomes important. The dominant message passing call is MPI_Send.

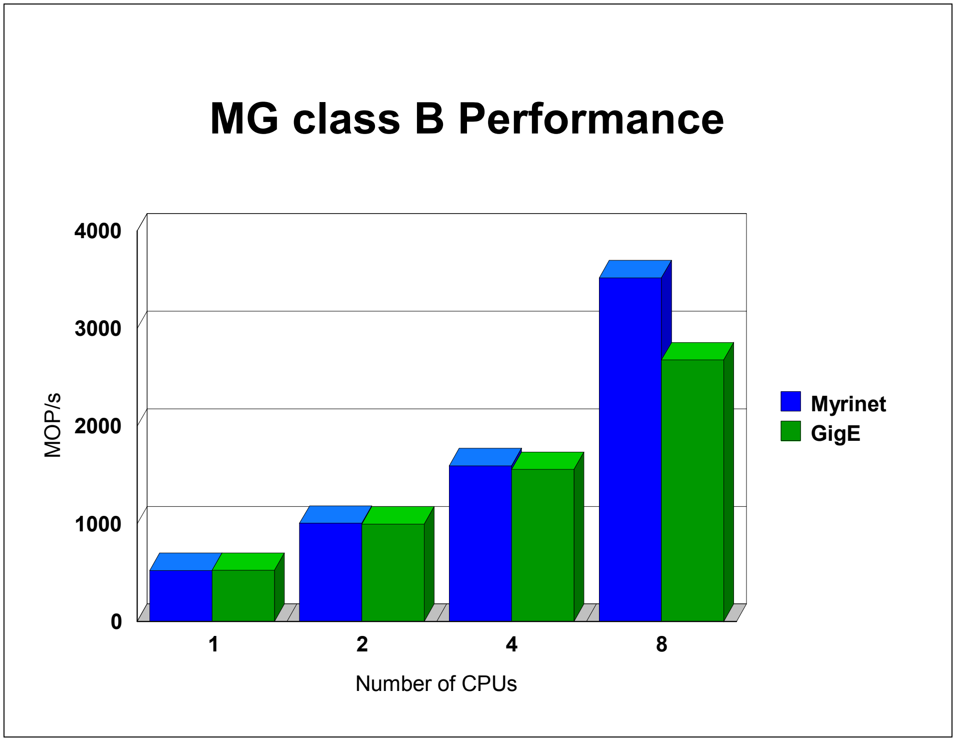


*Figure 7-23 NAS MG class B performance*

For CG (Figure 7-24 on page 354), the inter-node communication becomes even more important, and for the eight-CPU case, the Myrinet interface is 50% better than the Gigabit Ethernet interface. The dominant message passing call for this application is MPI_Send.

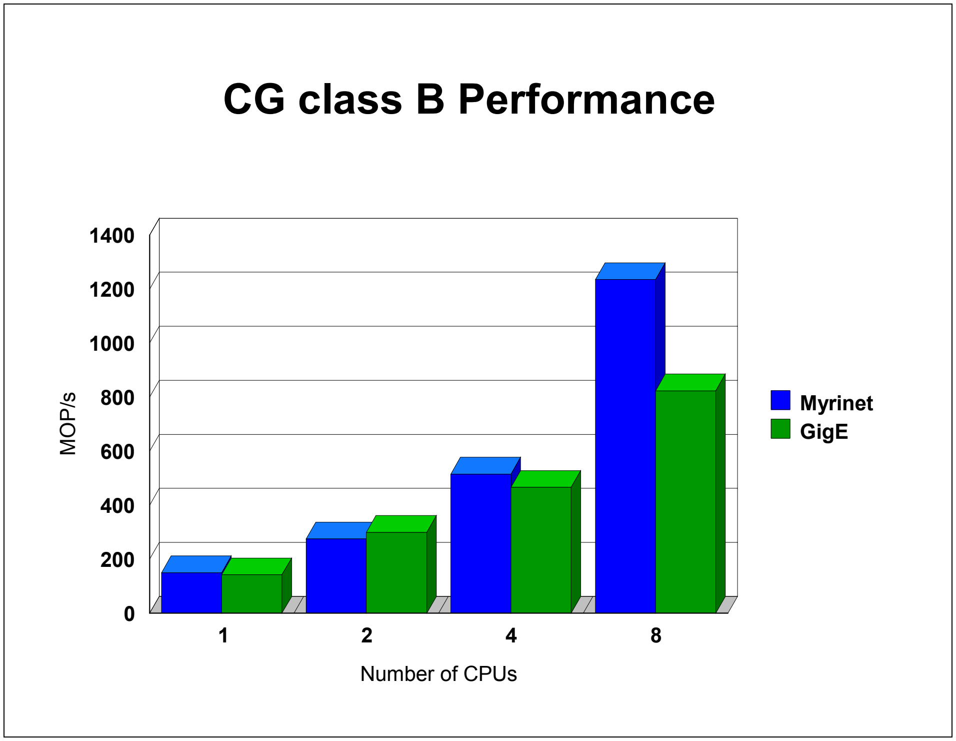


*Figure 7-24 NAS CG class B performance*

FT (Figure 7-25 on page 355) does a significant amount of MPI_Alltoall communication, requiring significant collective communication bandwidth. As a result, for the eight-CPU case, the Myrinet interface is nearly 100% faster than the Gigabit Ethernet interface.

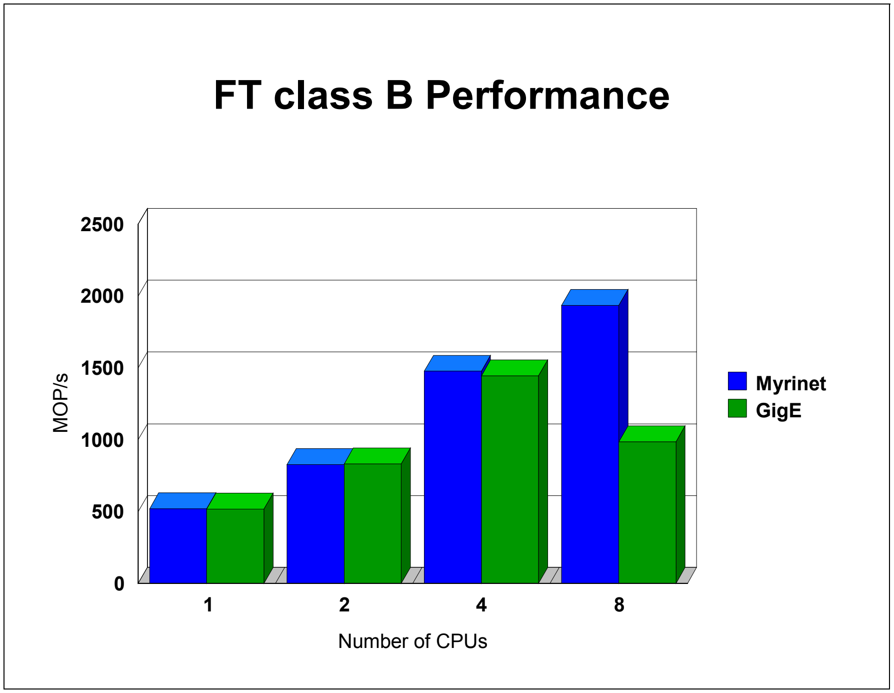


*Figure 7-25 NAS FT class B performance*

The integer sort benchmark is quite sensitive to collective communication performance (since the dominant message passing calls in this application are MPI_Allgatherv, and MPI_Alltoall). Therefore, there is a huge (nearly 400%) difference in performance for the eight-CPU case between Myrinet and Gigabit Ethernet interfaces (Figure 7-26 on page 356).

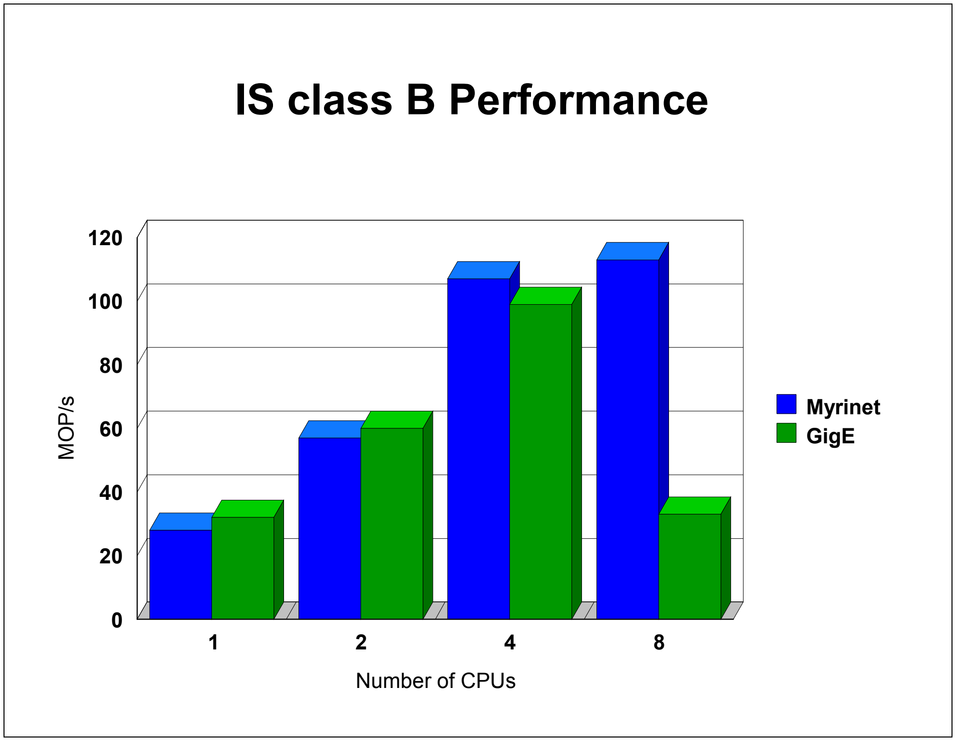


*Figure 7-26 NAS IS class B performance*

The BT results are shown in Figure 7-27 on page 357. Since BT has to be run on square grid of processors, it was run only on 1 & 4 CPUs—all within the same node. There is no inter-node communication, and there are no differences between interconnects. The primary message passing calls in this application are non-blocking sends and receives.

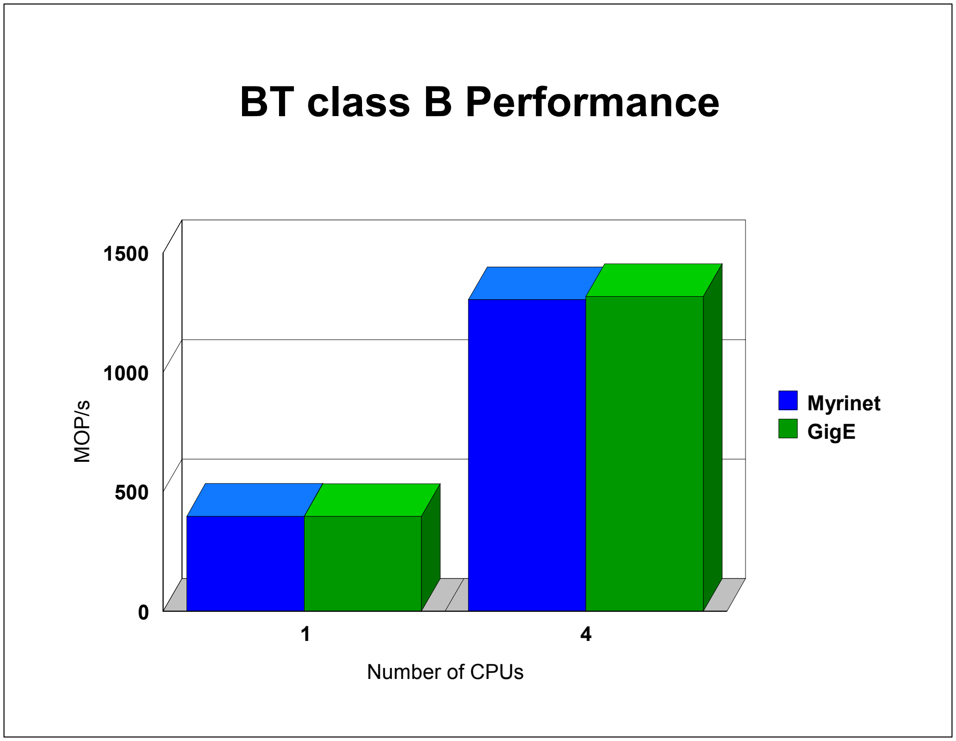


*Figure 7-27 NAS BT class B performance*

### *SP*

As in the case of BT, SP has to be run only on a square number of processors, and hence this was run within a node; there was no inter-node communication. The primary message passing calls here are non-blocking sends and receives like BT, except that SP is a much more communication-intensive application (see Figure 7-28 on page 358 for performance comparisons).

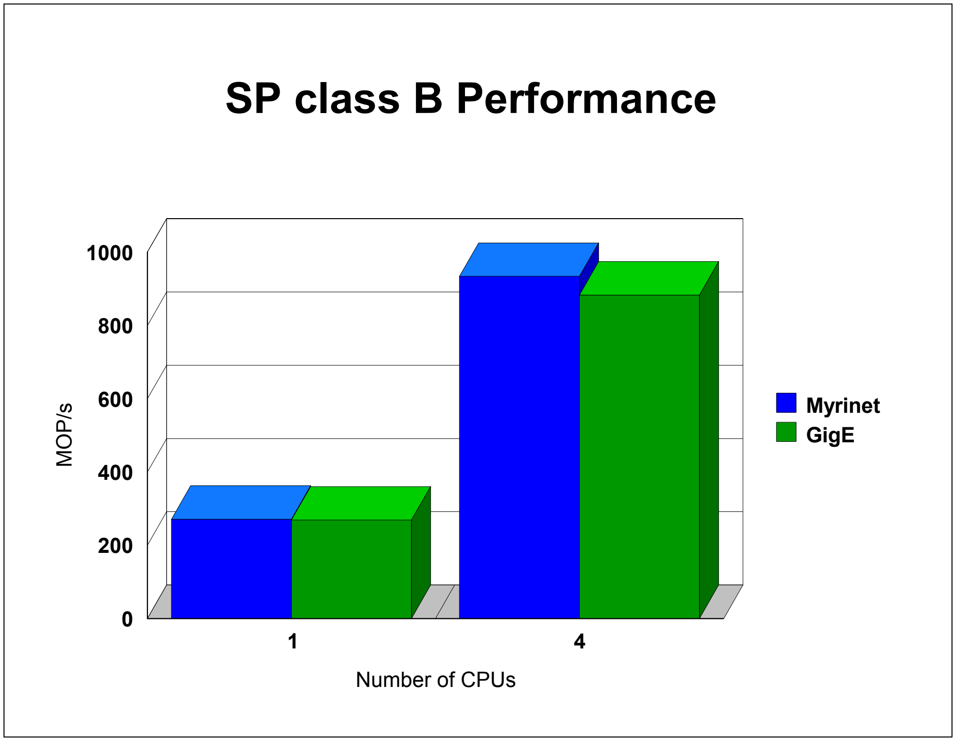


*Figure 7-28 NAS SP class B performance*

### *LU*

For the eight-way LU, the Myrinet interface has a slight advantage over the Gigabit Ethernet interface (Figure 7-29 on page 359). The primary message passing calls are non-blocking send and receive.

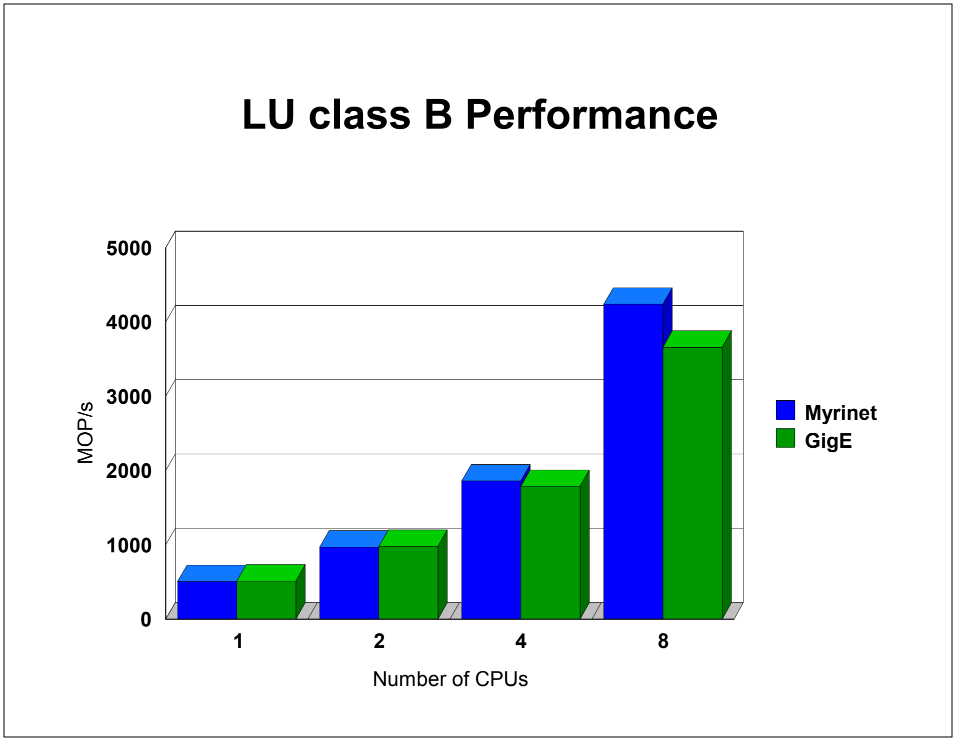


*Figure 7-29   NAS LU class B performance*

8

# Commercial application case studies

In this chapter, we discuss the following topics:

- 8.1, “High availability with heartbeat and DB2” on page 362
- 8.2, “High Availability with Red Hat ClusterSuite” on page 393
- 8.3, “Tivoli Storage Manager (TSM)” on page 406

# 8.1 High availability with heartbeat and DB2

In most environments, customers and users are very concerned about system uptime, especially regarding running their daily applications. In this case study, we discuss the high availability solution that is bundled along with SuSE Linux Enterprise Server (SLES) 8 and the IBM Data Management Solution, DB2®. While this setup is focused on how to implement database failover, it is applicable to, and similar for, other services.

## 8.1.1 Hardware and software components

In this case study, we have two logical partitions (LPARs), each with two processors (POWER4+ at 1.45 Ghz), 4 GB of memory, and one 36.4 Gb internal disk drive. In addition, the following adapters/peripherals are assigned to the system:

- 2 network interfaces (FC-4962 and FC-5700)
- RJ45 UTP cross-cable
- SCSI controller (FC-6203)
- JBOD SCSI disk drawer (2104-DU3)

The system is installed with SLES8 SP1, kernel 2.4.21-83 using the 64-bit kernel. The software stack that we used in the setup are:

- Heartbeat-1.0.4-0 clustering stack
- IBM DB2 8.1

Heartbeat is an open source clustering solution written by Alan Robertson, which provides basic clustering solutions. Heartbeat monitors the cluster resource either using network or serial adapters. It is also bundled with scripts to create cluster IP addresses and manage Linux Virtual Service (LVS) and other applications.

In this case study, we set up a cluster with a DB2 database in the external disk and cluster IP address for remote connection. Figure 8-1 shows the cluster that we will be setting up.

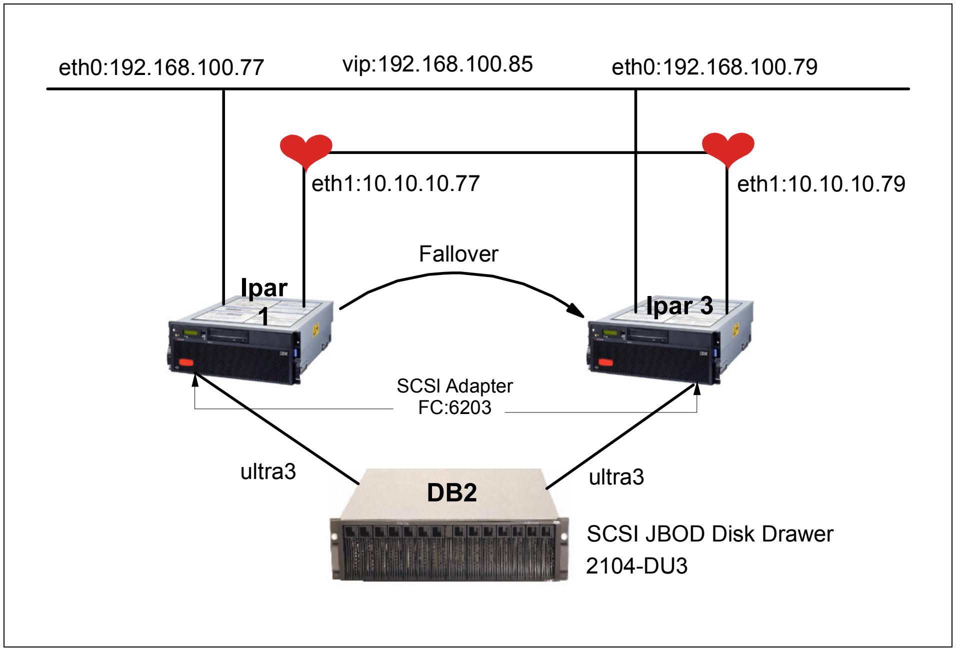


*Figure 8-1 DB2 cluster*

## 8.1.2 Preparing the nodes for cluster ready

As with most high availability clusters, we have here external storage for disk takeover, network adapters, a dedicated LAN for heartbeats, and so on. In our setup, we call the servers that we are attaching to the disk lpar1 and lpar3; lpar1 is the primary server, and lpar3 is the failover server.

### Network setup

First we assign IP addresses to the systems. We assign eth0 to be on the public LAN, and eth1 becomes the heartbeat LAN. In this setup, we are limited by the available adapters.

We recommend that you have serial connection between the two servers, as well. This ensures that you will have a non-IP -based heartbeat.

- lpar1
  - eth0 - 192.168.100.77
  - eth1 - 10.10.10.77
- lpar3
  - eth0 - 192.168.100.79
  - eth1 - 10.10.10.79

## Disk setup

Data disk is the most critical component in almost any clustering solution. Data disk ensures that the application reads the latest data from storage.

In our cluster, we are using the IBM SCSI-based external disk storage solution, also known as the 2104-DU3 with the IBM Ultra3 SCSI adapters. The 2104-DU3 storage has capability as a single bus or a twin-bus configuration. In cluster setups, we require the storage to be a single bus. Figure 8-2 explains the single bus, dual-host configuration.

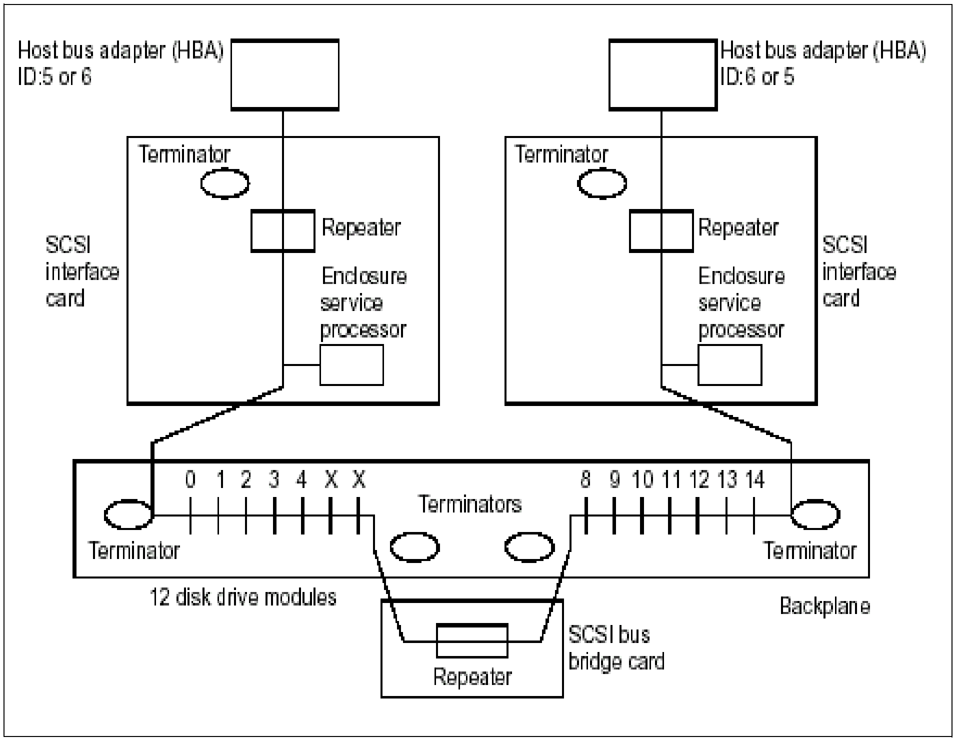


*Figure 8-2 Diagram of a single bus with dual host configuration*

For information on how to change the storage to a single bus, refer to the *2104-DU3 Installation Guide*, GA33-3311.

After setting up the storage for single bus, we need to change the SCSI ID of the adapters in the servers. The SCSI adapters defaults itself to SCSI ID 7, and that creates a conflict if both servers are booted at the same time. Therefore, we change the lpar1 SCSI ID to 5 and lpar3 SCSI ID to 6. The SCSI driver for the IBM FC-6203 SCSI adapter is “sym53c8xx”. This driver is compiled natively into the SLES8 Linux kernel.

We try out the new SCSI ID by entering the following string into the yaboot prompt:

```
yaboot : linux root=/dev/sd2 sym53c8xx=hostid:5
```

After the system is booted up, we change the SCSI of the server permanently by appending the above parameters into the kernel image so that it always is called prior to loading the kernel.

Obtain the **mkzimage_command** from the /ppc/netboot/ directory inside your SLES 8 CD:

```
# cp /boot/vmlinuz .
# mkzimage_command -c ./vmlinuz
# mkzimage_command -a 1 -s "root=/dev/sd2 sym53c8xx=hostid:5" ./vmlinuz
# cp vmlinuz/boot/vmlinuz.051103
```

Next, update the /etc/lilo.conf file with the new image, as shown in Example 8-1.

*Example 8-1 Updating LILO configuration file with new kernel*

```
# Generated by YaST2

default=test
timeout=100
boot=/dev/sda1
activate

image = /boot/vmlinuz
        label = linux
        root = /dev/sda2
        append = ""

image = /boot/vmlinuz.051103
        label = test
        root = /dev/sda2
        append = ""
```

Then we ran the **lilo** command to load the kernel and reboot the system, and did the same for lpar3 with SCSI ID6.

With both lpar1 and lpar3 capable of seeing the storage, we now create the disk partitions for storing our data. To create the new partition, we used the command **fdisk**.

In Example 8-2, we create a 10 Gb disk partition in the external storage. Once the partition is created, it is instantly visible to lpar3 as well. We run the command **fdisk -l** to check.

*Example 8-2 Creating a new 10 Gb partition in the newly added disk*

```
leecy@lpar1:~ # fdisk /dev/sdc

The number of cylinders for this disk is set to 34715.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdc: 64 heads, 32 sectors, 34715 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-34715, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-34715, default 34715): +10GB

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Now that we have the disk up and running, we create the file system and directories for our application:

```
# mkdir /data/IBM/db2inst1
# mkfs.reiserfs /dev/sdc1
# mount /dev/sdc1 /data/IBM/db2inst1
```

> **Note:** We are using IBM DB2 as our application.

Once we mount the file system, we can proceed to install our application.

## 8.1.3 Application installation

Prior to installing IBM DB2, we run through the hardware and software requirements:

- 650 Mb of disk space for full installation
- IBM JDK 1.3.1 for DB2 control center
- Flex-2.5.4a-39
- Web browser for online help

At the following site, you can check the tested kernel against the DB2 release that you are going to instal:

`http://www.ibm.com/db2/linux/validate`

Once we have all the correct software and hardware requirements, we proceed to the installation of DB2.

### DB2 installation

After mounting the CD into the CD-ROM drive, we run the command **`db2setup`** in the root of the CD-ROM directory. A screen appears, as shown in Figure 8-3 on page 368.

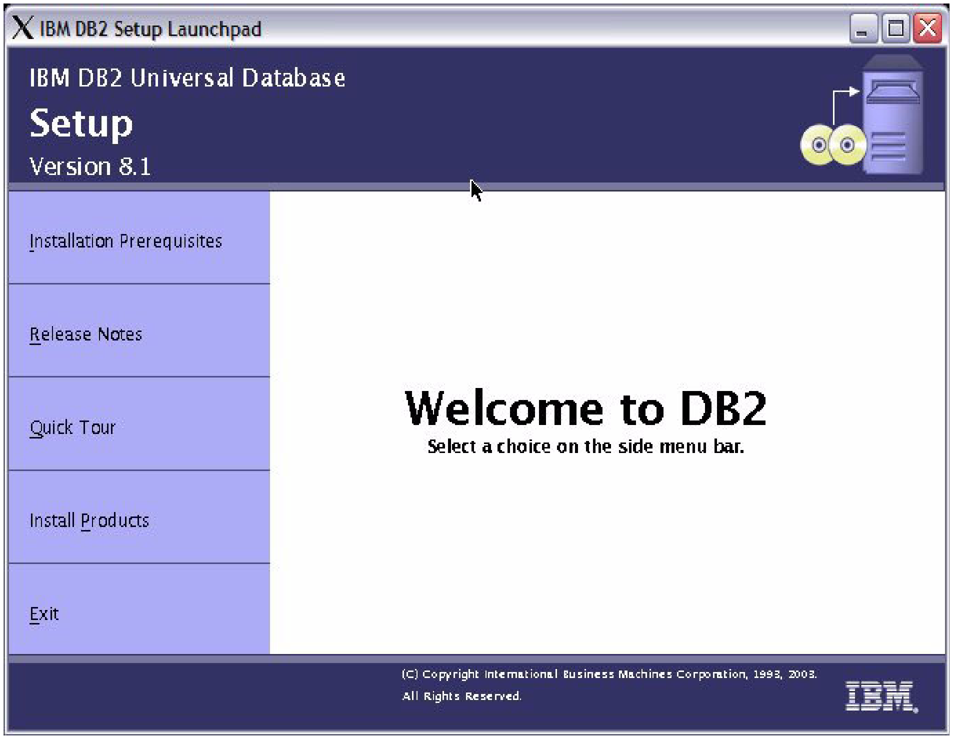
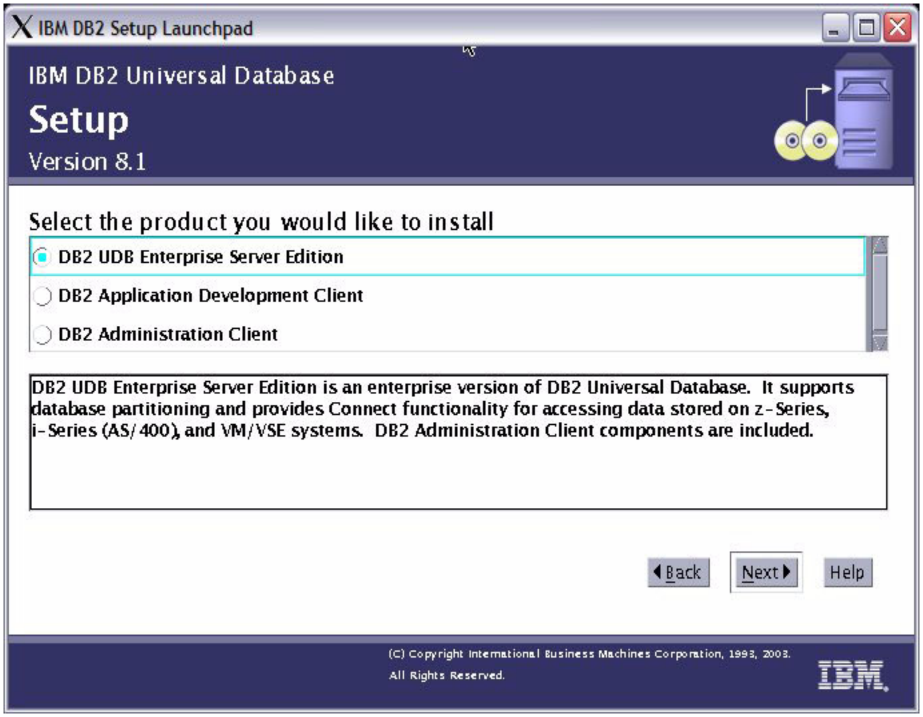

IBM DB2 Setup Launchpad
IBM DB2 Universal Database
Setup
Version 8.1
Installation Prerequisites
Release Notes
Quick Tour
Install Products
Exit
Welcome to DB2
Select a choice on the side menu bar.
(C) Copyright International Business Machines Corporation, 1993, 2003.
All Rights Reserved.
IBM

*Figure 8-3 DB2 setup*

We select **Install Products**. This presents us with choices of installation and the type of installation we want to install for the server. Figure 8-4 on page 369 shows the types of installation choices available.



Select the product you would like to install

● DB2 UDB Enterprise Server Edition
○ DB2 Application Development Client
○ DB2 Administration Client

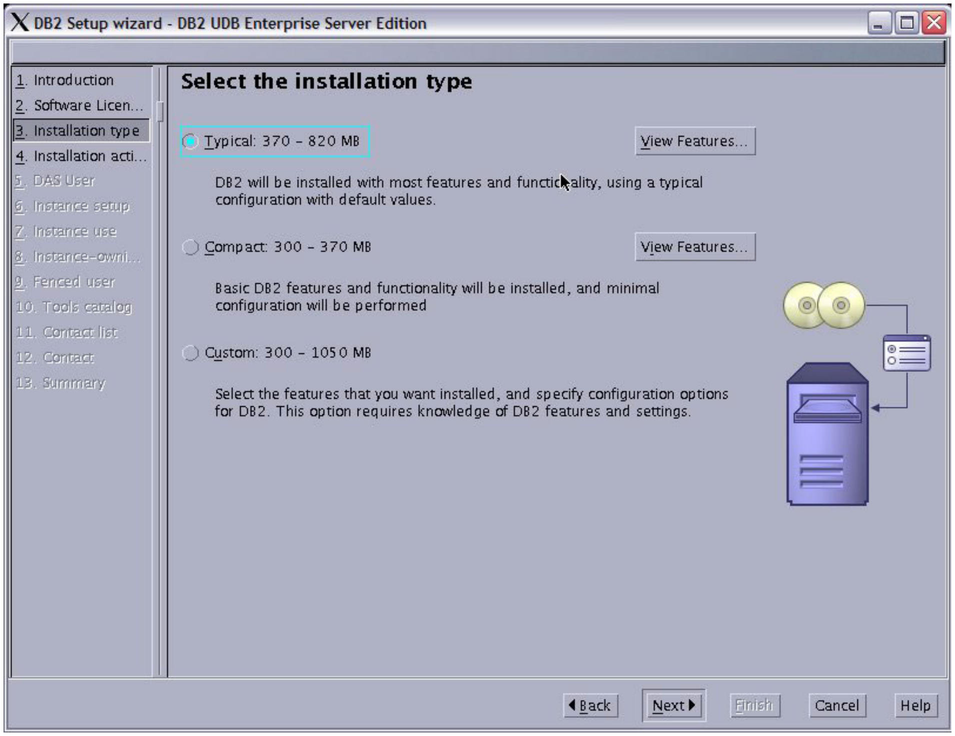
DB2 UDB Enterprise Server Edition is an enterprise version of DB2 Universal Database. It supports database partitioning and provides Connect functionality for accessing data stored on z-Series, i-Series (AS/400), and VM/VSE systems. DB2 Administration Client components are included.

◀ Back
Next ▶
Help
(C) Copyright International Business Machines Corporation, 1993, 2003.
All Rights Reserved.
IBM

*Figure 8-4 Install types*

We select **DB2 UDB Enterprise Server Edition** and click Next. In the next screen, we are asked to accept the IBM DB2 user license agreement, then we click Next.

Then we are prompted to select the type of DB2 UDB ESE we want to install (Typical, Compact, Custom), as shown in Figure 8-5 on page 370. We select **Typical** and proceed.



Select the installation type

● Typical: 370 - 820 MB

DB2 will be installed with most features and functionality, using a typical configuration with default values.

○ Compact: 300 - 370 MB

Basic DB2 features and functionality will be installed, and minimal configuration will be performed

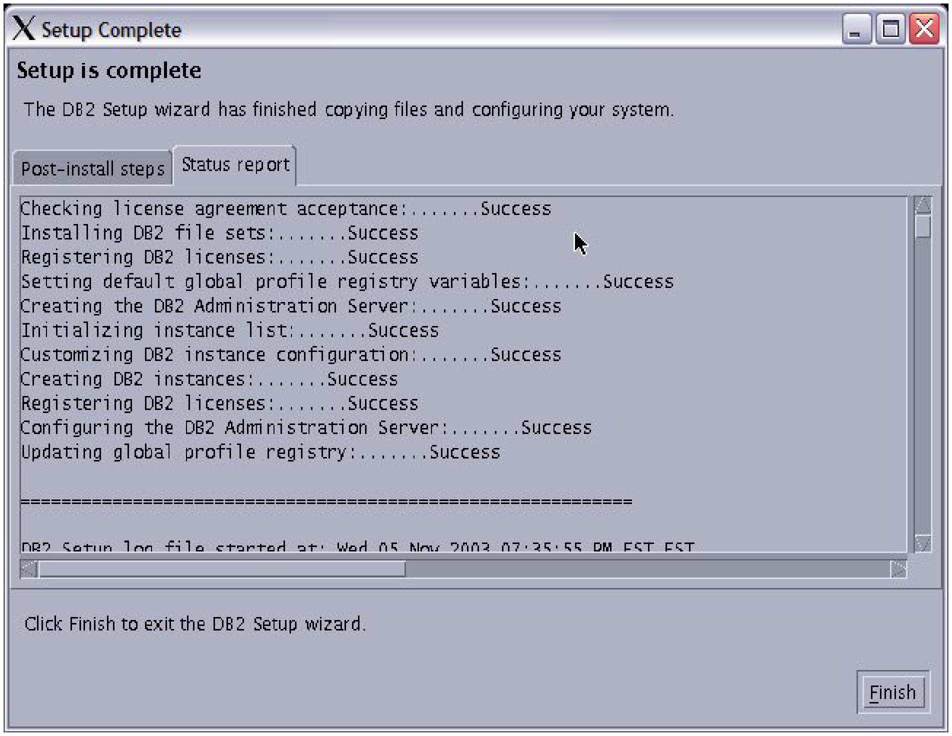
○ Custom: 300 - 1050 MB

Select the features that you want installed, and specify configuration options for DB2. This option requires knowledge of DB2 features and settings.

*Figure 8-5 DB2 installation type selection screen*

Next, when prompted, we create the necessary DB2 IDs. At this stage, we are still using the internal disk to store DB2. We will move the database to the external storage once we have both nodes installed.

Once we have done the selecting, installation will start. After it finishes, we check the post-installation report to make sure all components are installed. Figure 8-6 on page 371 shows the sample post-installation report.



Setup is complete

The DB2 Setup wizard has finished copying files and configuring your system.

```
Checking license agreement acceptance:.......Success
Installing DB2 file sets:.......Success
Registering DB2 licenses:.......Success
Setting default global profile registry variables:.......Success
Creating the DB2 Administration Server:.......Success
Initializing instance list:.......Success
Customizing DB2 instance configuration:.......Success
Creating DB2 instances:.......Success
Registering DB2 licenses:.......Success
Configuring the DB2 Administration Server:.......Success
Updating global profile registry:.......Success

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

DB2 Setup log file started at: Wed 05 Nov 2003 07:35:55 PM EST EST
```

Click Finish to exit the DB2 Setup wizard.

*Figure 8-6 DB2 post-installation*

Now that DB2 is installed, we need to create a database to use. We plan to use the DB2 sample bundled with DB2. To load the sample database, we run the **db2sampl** command as db2inst1 user.

Next, we disable DB2 from starting automatically when the system is booted up. We comment out the DB2 entry inside the /etc/inittab file. We want the clustering solution to automatically bring up DB2 for us, instead.

Now that DB2 is properly set up, we test the sample database that we loaded. We connect to the database locally and do a simple query. Example 8-3 shows a successful connection to the sample database.

*Example 8-3 Testing the connection to the DB2 sample database*

```
lpar1:/home # su - db2inst1
db2inst1@lpar1:~> db2 connect to sample

   Database Connection Information

 Database server        = DB2/LINUXPPC 8.1.2
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE
```

Next, we do a simple query to make sure that we can query the database. Example 8-4 shows a simple query of the local database.

*Example 8-4 Querying the DB2 sample database*

```
db2inst1@lpar1:~> db2 "select * from ORG"

DEPTNUMB DEPTNAME       MANAGER DIVISION   LOCATION
-------- -------------- ------- ---------- -------------
      10 Head Office        160 Corporate  New York
      15 New England         50 Eastern    Boston
      20 Mid Atlantic        10 Eastern    Washington
      38 South Atlantic      30 Eastern    Atlanta
      42 Great Lakes        100 Midwest    Chicago
      51 Plains             140 Midwest    Dallas
      66 Pacific            270 Western    San Francisco
      84 Mountain           290 Western    Denver

  8 record(s) selected.
```

We now do a similar installation in lpar3. (Select the same directory as in lpar1 to store the user directories, because this will ensure that DB2 sets up the profile and directories properly.)

After the installation of DB2 in both nodes, we move the database that we created in lpar1 to the external disk. We mount the disk from external storage, copy the entire DB2 instance folder into the external disk, and create a soft link. When the DB2 instance (db2inst1) is loaded, it is using the external disk:

```
# mount /dev/sdc1 /IBM/opt/db2inst1
# cp /home/db2inst1 /IBM/opt/db2inst1
# ln -s /IBM/opt/db2inst1/db2inst1 /home/db2inst1
# chown db2inst1.db2grp1 /home/db2inst1
# umount /IBM/opt/db2inst1
```

With lpar1 working, we now create a similar soft link in lpar3:

```
# mv /home/db2inst1 /home/db2inst1.orig
# mkdir /IBM/data/db2inst1
# mount /dev/sdc1 /IBM/data/db2inst1
# ln -s /IBM/data/db2inst1 /home/db2inst1
# chown db2inst1.db2grp1 /home/db2inst1
```

## 8.1.4 Heartbeat clustering software installation

With DB2 installed, we now install and configure the heartbeat clustering solution to manage our storage, application, and IP address. We install the following packages into our two systems (lpar1 and lpar3). The packages are available in the SLES 8 CD:

```
# rpm -ivh heartbeat-ldirectord-1.0.4-0.rpm
# rpm -ivh heartbeat-1.0.4-0.rpm
# rpm -ivh heartbeat-stonith-1.0.4-0.rpm
```

In the heartbeat clustering, there are three major configuration files:

- /etc/ha.d/authkeys
- /etc/ha.d/ha.cf
- /etc/ha.d/haresources

In the following sections, we describe each file.

### /etc/ha.d/authkeys

The authkeys configuration file specifies the secret authentication keys that must be identical for both nodes in the cluster. There are several different authentication encryptions you can choose. In our setup, we use md5. The authkeys is shown in Example 8-5.

*Example 8-5 /etc/ha.d/authkeys*

```
lpar1:/etc/ha.d # cat authkeys
# key for the cluster is linuxforp
auth 3
3 md5 linuxforp
```

The authkeys must be set to read only by the root user; otherwise, the heartbeat software will fail right away:

```
# chmod 600 /etc/ha.d/authkeys
```

### /etc/ha.d/ha.cf

The ha.cf file is the core configuration file that defines the nodes which are part of the clustering. In this file, we also define which link we use for the heartbeat, and the sequence of the heartbeat. Example 8-6 shows the ha.cf configuration file that we use in our configuration.

*Example 8-6 /etc/ha.d/ha.cf*

```
# Logs definition
debugfile /var/log/ha-debug
logfile /var/log/ha-log
```

```
logfacility local10

# HeartBeat Packets Configuration
keepalive 2 # time between each heartbeat
deadtime 30 # how long to declare dead
bcast eth1 # heartbeat communication link

# Resource Configuration
nice_failback on # this will turn on the feature cascading without fall-back

# Node Definition
node lpar1
node lpar3
```

### /etc/ha.d/haresources

The haresources file manages the resources that you want to be part of the cluster. The heartbeat looks into the /etc/ha.d/resource.d file and the /etc/init.d/ directory for scripts to start your application that you specify in the haresources file. In our setup, we required a cluster IP, the external storage to be automatically mounted, and then DB2 to be started.

*Example 8-7 /etc/ha.d/haresources*

```
lpar1   192.168.100.85 Filesystem::/dev/sdc1::/data/IBM/db2inst1::reiserfs
db2::db2inst1
```

Based on the logic of how the application is started, we create the haresources configuration file. Example 8-7 shows the contents of the file we have. It tells heartbeat to make lpar1 be the primary node with cluster IP address 192.168.100.85, and then mount the file system to the mountpoint /data/IBM/db2inst1, and then start DB2 with the instance ID db2inst1.

After this is done, we customize the DB2 script located inside /etc/ha.d/resource.d/ for DB2 8 ESE. Because DB2 is parallel database-capable, we added these lines into the script, as shown in Example 8-8.

*Example 8-8 Extract of /etc/ha.d/resource.d/db2*

```
:
:
:
db2_start() {

#### included for DB2 8.1 EEE
NODENAME=`hostname`

cp /home/db2inst1/sqllib/db2nodes.cfg.$NODENAME
/home/db2inst1/sqllib/db2nodes.cfg

#### included for DB2 8.1 EEE

  if
    output=`runasdb2 $db2adm/db2start`
  then
    : Hurray! DB2 started OK
    ha_log "info: DB2 UDB instance $1 started: $output"
  else
    case $output in
      SQL1026N*|*"is already active"*)
                ha_log "info: DB2 UDB instance $1 already running: $output";;

      *)        ha_log "ERROR: $output"; return 1;;
    esac
  fi
:
:
:
```

We need to create the necessary db2nodes.cfg for lpar1 and lpar3 inside /home/db2inst1/sqllib/:

- /home/db2inst1/sqllib/db2nodes.cfg.lpar1 contains:
  - 0 lpar1 0
- /home/db2inst1/sqllib/db2nodes.cfg lpar3 contains:
  - 0 lpar3 0

Now that we have all the configuration files ready, we copy the configuration to lpar3:

```
# cd /etc/ha.d
# scp haauthkeys ha.cf haresources resource.d/db2 root@lpar3:/etc/ha.d/
```

Next, we do a basic check on the cluster setup by using the command **BasicSanityCheck**. This command is found in the directory /usr/lib/heartbeat.

This command performs basic checks and outputs the errors (if any) into the /tmp/linux-ha.testlog file.

Example 8-9 on page 376 shows the BasicSanityCheck command that we ran during the creation of the configuration file for our cluster.

*Example 8-9 BasicSanityCheck on the heartbeat cluster*

```
leecy@lpar1:/usr/lib/heartbeat # ./BasicSanityCheck
Starting heartbeat
Starting High-Availability services
done

Reloading heartbeat

Reloading heartbeat
Stopping heartbeat
Stopping High-Availability services
done
Checking STONITH basic sanity.
Performing apphbd success case tests
Performing apphbd failure case tests
Starting IPC tests
1 errors. Log file is stored in /tmp/linux-ha.testlog
```

## 8.1.5 Testing the cluster

Once done, we test the cluster. We start the cluster by using the command **/etc/init.d/heartbeat start** as shown in Example 8-10.

*Example 8-10 Starting the heartbeat cluster*

```
lpar1:~ # /etc/init.d/heartbeat start
Starting High-Availability services
done
lpar1:~ #

```

We started the cluster on both nodes and then noticed that the cluster IP address automatically gets created in the primary node. At the same time, the file system gets mounted and the application started.

During this process, the /var/log/ha-log shows details of what is happening in the background. This log is also very useful for debugging if the resource fails to start. Example 8-11 shows the ha-log of our cluster with the resource successfully started.

*Example 8-11 /var/log/ha-log output when cluster is up*

```
# tail -f /var/log/ha-log
heartbeat: 2003/11/10_17:53:34 info: **************************
heartbeat: 2003/11/10_17:53:34 info: Configuration validated. Starting
heartbeat 1.0.4
heartbeat: 2003/11/10_17:53:34 info: nice_failback is in effect.
```

```
heartbeat: 2003/11/10_17:53:34 info: heartbeat: version 1.0.4
heartbeat: 2003/11/10_17:53:34 info: Heartbeat generation: 30heartbeat:
2003/11/10_17:53:34 info: UDP Broadcast heartbeat started on port 694 (694)
interface eth1
heartbeat: 2003/11/10_17:53:34 info: pid 2795 locked in memory.
heartbeat: 2003/11/10_17:53:35 info: pid 2797 locked in memory.
heartbeat: 2003/11/10_17:53:35 info: pid 2799 locked in memory.
heartbeat: 2003/11/10_17:53:35 info: pid 2798 locked in memory.
heartbeat: 2003/11/10_17:53:35 info: Local status now set to: 'up'
heartbeat: 2003/11/10_17:53:36 info: Link lpar1:eth1 up.
heartbeat: 2003/11/10_17:54:05 WARN: node lpar3: is dead
heartbeat: 2003/11/10_17:54:05 WARN: No STONITH device configured.
heartbeat: 2003/11/10_17:54:05 WARN: Shared resources (storage!) are not
protected!
heartbeat: 2003/11/10_17:54:05 info: Resources being acquired from lpar3.
heartbeat: 2003/11/10_17:54:05 info: Local status now set to: 'active'
heartbeat: 2003/11/10_17:54:05 info: Running /etc/ha.d/rc.d/status status
heartbeat: 2003/11/10_17:54:05 info: /usr/lib/heartbeat/mach_down:
nice_failback: acquiring foreign resources
heartbeat: 2003/11/10_17:54:05 info: mach_down takeover complete.
heartbeat: 2003/11/10_17:54:05 info: mach_down takeover complete for node
lpar3.
heartbeat: 2003/11/10_17:54:05 info: Resource acquisition completed.
heartbeat: 2003/11/10_17:54:05 info: Running /etc/ha.d/rc.d/ip-request-resp
ip-request-resp
heartbeat: 2003/11/10_17:54:05 received ip-request-resp 192.168.100.85 OK yes
heartbeat: 2003/11/10_17:54:05 info: Acquiring resource group: lpar1
192.168.100.85 Filesystem::/dev/sdb1::/data/IBM/db2inst1::reiserfs
db2::db2inst1
heartbeat: 2003/11/10_17:54:05 info: Running /etc/ha.d/resource.d/IPaddr
192.168.100.85 start
heartbeat: 2003/11/10_17:54:05 info: /sbin/ifconfig eth0:0 192.168.100.85
netmask 255.255.255.0 broadcast 192.168.100.255
heartbeat: 2003/11/10_17:54:05 info: Sending Gratuitous Arp for 192.168.100.85
on eth0:0 [eth0]
heartbeat: 2003/11/10_17:54:05 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A068C 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_17:54:05 info: Running /etc/ha.d/resource.d/Filesystem
/dev/sdb1 /data/IBM/db2inst1 reiserfs start
heartbeat: 2003/11/10_17:54:06 info: Running /etc/ha.d/resource.d/db2 db2inst1
start
heartbeat: 2003/11/10_17:54:07 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A068C 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_17:54:08 info: DB2 UDB instance db2inst1 started:
11-10-2003 17:54:08     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
heartbeat: 2003/11/10_17:54:09 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A068C 192.168.100.85 ffffffffffff
```

```
heartbeat: 2003/11/10_17:54:11 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A068C 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_17:54:13 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A068C 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_17:54:17 info: Local Resource acquisition completed.
(none)
heartbeat: 2003/11/10_17:54:17 info: local resource transition completed.
heartbeat: 2003/11/10_17:54:28 info: Link lpar3:eth1 up.
heartbeat: 2003/11/10_17:54:28 info: Status update for node lpar3: status up
heartbeat: 2003/11/10_17:54:28 info: Running /etc/ha.d/rc.d/status status
heartbeat: 2003/11/10_17:54:29 info: Status update for node lpar3: status
active
heartbeat: 2003/11/10_17:54:29 info: Running /etc/ha.d/rc.d/status status
```

Next, we power off lpar1. The resource instantly fails over to lpar3, with the cluster IP addresses created as well, as in Figure 8-7.

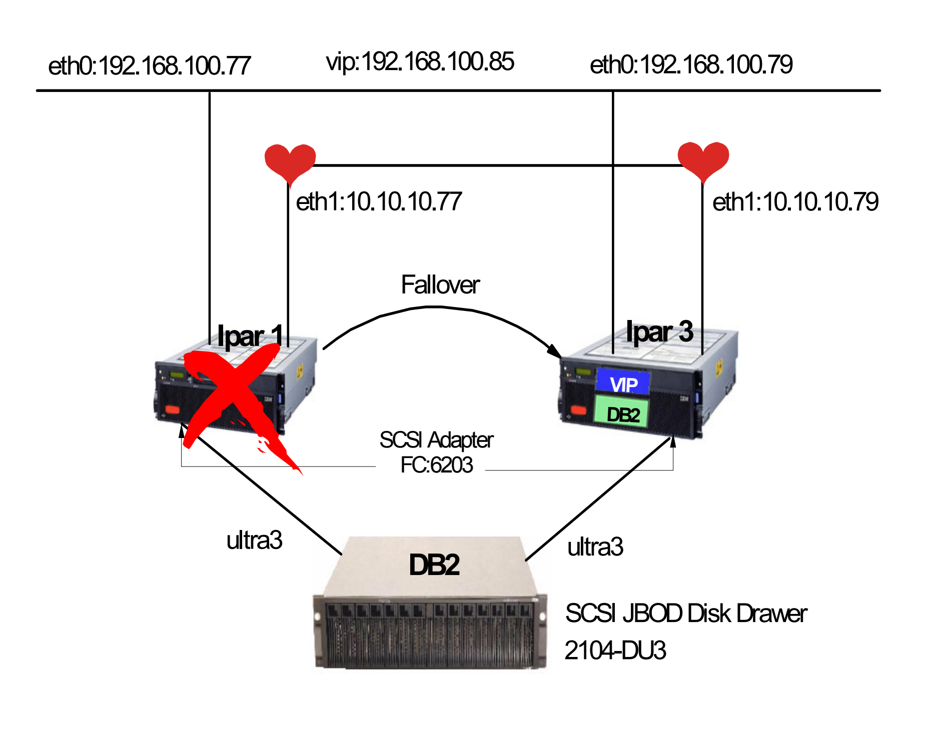


*Figure 8-7 DB2 failover test*

The entries in /var/log/ha-log show that lpar1 has failed and the resources fails over to lpar3. Example 8-12 on page 379 shows /var/log/ha-log during our test.

*Example 8-12 Fail-over log*

```
heartbeat: 2003/11/10_19:32:16 WARN: node lpar1: is dead
heartbeat: 2003/11/10_19:32:16 WARN: No STONITH device configured.
heartbeat: 2003/11/10_19:32:16 WARN: Shared resources (storage!) are not
protected!
heartbeat: 2003/11/10_19:32:16 info: Resources being acquired from lpar1.
heartbeat: 2003/11/10_19:32:16 info: Link lpar1:eth1 dead.
heartbeat: 2003/11/10_19:32:16 info: Running /etc/ha.d/rc.d/status status
heartbeat: 2003/11/10_19:32:16 info: No local resources
[/usr/lib/heartbeat/ResourceManager listkeys lpar3]
heartbeat: 2003/11/10_19:32:16 info: Resource acquisition completed.
heartbeat: 2003/11/10_19:32:16 info: Taking over resource group 192.168.100.85
heartbeat: 2003/11/10_19:32:16 info: Acquiring resource group: lpar1
192.168.100.85 Filesystem::/dev/sdb1::/data/IBM/db2inst1::reiserfs
db2::db2inst1
heartbeat: 2003/11/10_19:32:16 info: Running /etc/ha.d/resource.d/IPaddr
192.168.100.85 start
heartbeat: 2003/11/10_19:32:17 info: /sbin/ifconfig eth0:0 192.168.100.85
netmask 255.255.255.0 broadcast 192.168.100.255
heartbeat: 2003/11/10_19:32:17 info: Sending Gratuitous Arp for 192.168.100.85
on eth0:0 [eth0]
heartbeat: 2003/11/10_19:32:17 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A0619 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_19:32:17 info: Running /etc/ha.d/resource.d/Filesystem
/dev/sdb1 /data/IBM/db2inst1 reiserfs start
heartbeat: 2003/11/10_19:32:18 info: Running /etc/ha.d/resource.d/db2 db2inst1
start
heartbeat: 2003/11/10_19:32:19 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A0619 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_19:32:21 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A0619 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_19:32:21 info: DB2 UDB instance db2inst1 started:
11-10-2003 19:32:21     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
heartbeat: 2003/11/10_19:32:23 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A0619 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_19:32:25 /usr/lib/heartbeat/send_arp eth0 192.168.100.85
0002553A0619 192.168.100.85 ffffffffffff
heartbeat: 2003/11/10_19:32:27 info: mach_down takeover complete for node
lpar1.
```

Now we are confident that our cluster is working as designed, so we add the heartbeat to the startup script for both nodes with the following command:

```
# chkconfig heartbeat 3
```

## 8.1.6 Extending the cluster with Apache and PHP for front-end

Now that we have the back-end DB2 cluster running, we set up the front-end servers to connect to the database server. We create two servers, lpar7 and lpar8, for Web servers, using Preprocessor Hypertext Protocol (PHP) software. PHP is an open source application which you can install from the SLES8 CD, or download from:

http://www.php.net/

We downloaded the latest Apache and PHP software products from the Web. Apache can be obtained from:

http://www.apache.org/homepage

### Configuring the Web server

We recompile the Apache software we downloaded from the Web to enable dynamic module support. We also put Apache into the /usr/local/apache directory:

```
# tar -zxvf <path>/apache_1.3.29.tar.gz
# ./configure --prefix=/usr/local/apache --enable-shared=max \
--enable-module=so
# make ; make install
```

After the Web server is compiled, we need to install DB2 into the server. We recommend that you install the typical server installation and leave the directory and user IDs as defaults.

After completed, we install the PHP script to interface with the DB2 database:

```
# tar -zxvf php-4.3.3.tar.gz
#./configure --with-prefix=/usr/local/php \
--with-apxs=/usr/local/apache/bin/apxs --without-mysql \
--with-ibm-db2=/home/db2inst1/sqllib
# make ; make install
```

With the Apache Web server and PHP installed, we need to update the configuration files to recognize the PHP extensions. We add the following lines to the /usr/local/apache/conf/httpd.conf file:

```
 AddType application/x-httpd-php     .php
 AddType application/x-httpd-php-source      .phps
```

Next, we create a sample PHP script with the text shown in Example 8-13 on page 381.

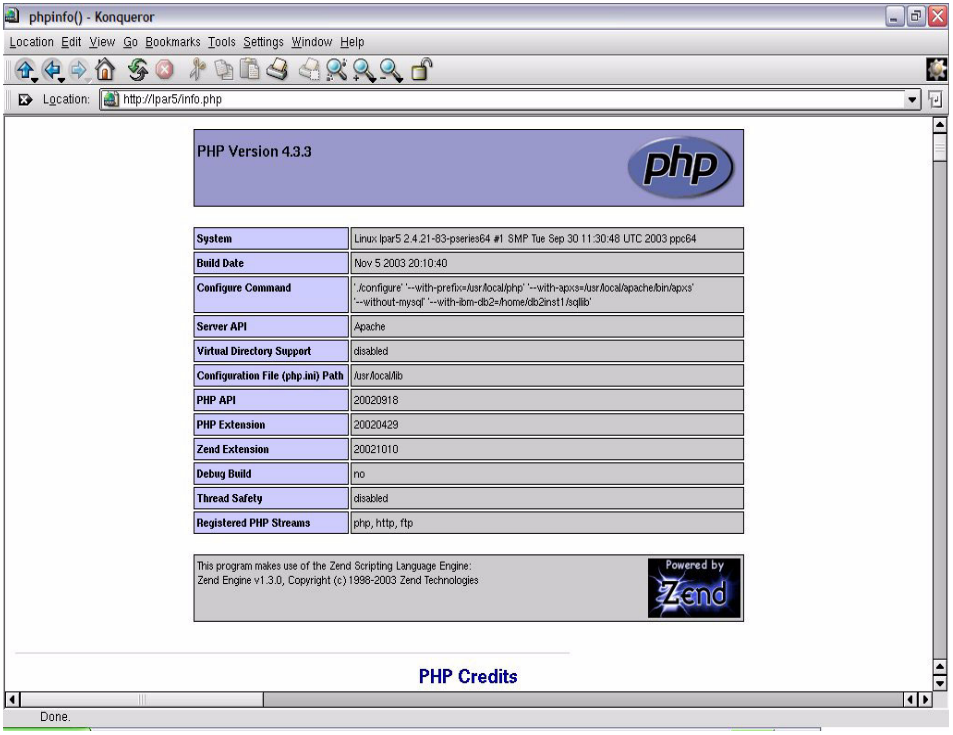
*Example 8-13 phpinfo test*

```
<?php
phpinfo()
?>
```

After compiling and successfully enabling PHP, we pointed our browser to:

```
http://localhost/phpinfo.php
```

The PHP test screen shown in Figure 8-8 appeared. We scrolled to the bottom and saw that the DB2 connection had been enabled using Open Database Connectivity (ODBC).


phpinfo() - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://lpar5/info.php
php
Powered by Zend
Done.

PHP Version 4.3.3

| | |
|---|---|
| **System** | Linux lpar5 2.4.21-83-pseries64 #1 SMP Tue Sep 30 11:30:48 UTC 2003 ppc64 |
| **Build Date** | Nov 5 2003 20:10:40 |
| **Configure Command** | './configure' '--with-prefix=/usr/local/php' '--with-apxs=/usr/local/apache/bin/apxs' '--without-mysql' '--with-ibm-db2=/home/db2inst1/sqllib' |
| **Server API** | Apache |
| **Virtual Directory Support** | disabled |
| **Configuration File (php.ini) Path** | /usr/local/lib |
| **PHP API** | 20020918 |
| **PHP Extension** | 20020429 |
| **Zend Extension** | 20021010 |
| **Debug Build** | no |
| **Thread Safety** | disabled |
| **Registered PHP Streams** | php, http, ftp |

This program makes use of the Zend Scripting Language Engine:
Zend Engine v1.3.0, Copyright (c) 1998-2003 Zend Technologies

**PHP Credits**

*Figure 8-8 phpinfo test page*

Now that we have PHP interfacing with DB2, we catalog the remote database that we created in lpar1 and lpar3 into lpar7.

We catalog the remote server using the cluster IP address (192.168.100.85) that we gave the DB2 heartbeat cluster:

```
# db2 catalog tcpip node lpar7 remote 192.168.100.85 server db2c_db2inst1
```

Next, we need to catalog the remote database; for simplicity, we call it *sample*.

```
# db2 catalog db sample as sample at node lpar7
```

After that, we test the database connection and do a db list directory, as shown in Example 8-14.

*Example 8-14   Testing the database connection*

```
db2inst1@lpar7:~> db2 connect to sample user db2inst1 using ibmdb2

   Database Connection Information

 Database server        = DB2/LINUXPPC 8.1.2
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE

db2inst1@lpar7:~> db2 "list db directory"

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = SAMPLE
 Database name                        = SAMPLE
 Node name                            = LPAR7
 Database release level               = a.00
 Comment                              =
 Directory entry type                 = Remote
 Catalog database partition number    = -1
```

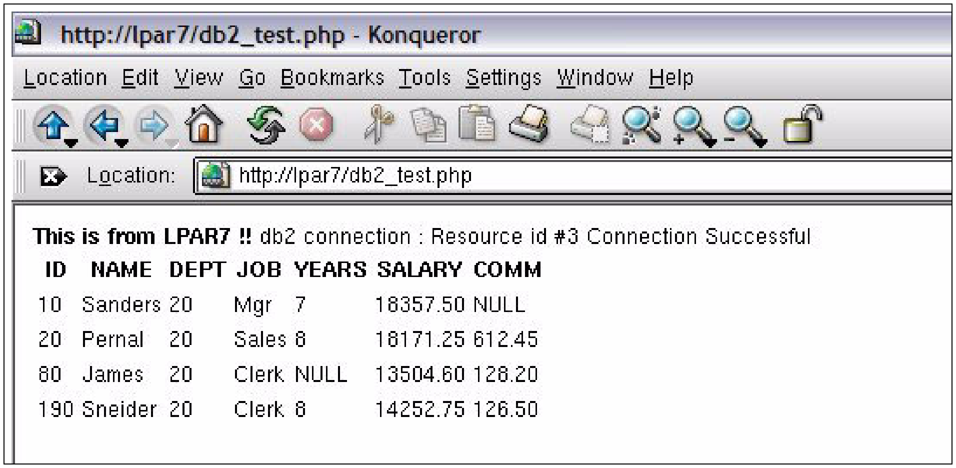
The test is successful, and we have connected to the remote database. Now, we create a simple PHP script to connect to the remote database. Example 8-15 shows the sample PHP script that is used to connect to the remote database and do a simple query of the database by looking for the staff in the department.

*Example 8-15   Sample PHP script to query remote DB2 database*

```
<?php
putenv("DB2INSTANCE=db2inst1");

$dbname = "sample";
$username = "db2inst1";
$password = "ibmdb2";

echo “This is from lpar7 !!!”;
```

```
$dbconn = odbc_pconnect($dbname, $username, $password);
echo "db2 connection : $dbconn \n";

if ($dbconn <= 0) {
echo "Error in connection";
exit;
}
else {
echo "Connection Successful \n";
};

$query = "SELECT * FROM staff WHERE dept=20";
$result = odbc_Exec($dbconn, $query );
odbc_result_all($result);
odbc_close($dbconn);

return($dbconn);

?>
```

Based on the script in Example 8-15, we point our browser to the server, and we get the database query displayed into a Web page. Figure 8-9 on page 383 shows the output of the script.


http://lpar7/db2_test.php - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location:
http://lpar7/db2_test.php

**This is from LPAR7 !!** db2 connection : Resource id #3 Connection Successful

| ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
|---|---|---|---|---|---|---|
| 10 | Sanders | 20 | Mgr | 7 | 18357.50 | NULL |
| 20 | Pernal | 20 | Sales | 8 | 18171.25 | 612.45 |
| 80 | James | 20 | Clerk | NULL | 13504.60 | 128.20 |
| 190 | Sneider | 20 | Clerk | 8 | 14252.75 | 126.50 |

*Figure 8-9 lpar7 db2_test.php page*

In your case, set up the same configuration for lpar8, and copy the php scripts to the node. Update the scripts to reflect lpar8 instead of lpar7, and test both Web servers. You should be able to connect to the database server from both lpar7 and lpar8 regardless of whether the database is on lpar1 or lpar3. This is transparent to the Web server, as it is talking to the DB2 cluster via the IP address 192.168.100.85.

### 8.1.7 Using LVS to load balance Web servers

We set up a load balancing cluster to create a single external IP for user connection. The load balancer distributes the requests into lpar7 and lpar8, and monitors the servers. In our test, we used IP Virtual Server (IVPS), which is the IP load balancing feature inside the Linux kernel.

IPVS examines every incoming packet and rewrites the packet to support load balancing. It also automatically creates IPVS rules hash tables for connections, and checks the connection table for established connections.

IPVS lets you choose how to set it up for load balancing, to manage the incoming packets and pass them to the back-end servers.There are basically three methods of configuring the IPVS:

- Via Network Address Translation (NAT)

  Network Address Translation translates incoming addresses to the back-end addresses.

- Via IP tunneling (TUN)

  With this method, packets are IPIP-encapsulated and then forwarded to the real servers.

- Via direct routing (DR)

  With this method, the MAC addresses in the packets are changed and the packets are forwarded to the real servers

Figure 8-10 on page 385 shows the differences between the different types of IPVS options available.

Some modifications are needed on the real server's ifconfig and routing table for LVS-DR and LVS-TUN forwarding, to enable the servers to talk directly to external clients. For LVS-NAT, the real servers only need a functioning TCP/IP stack.

In LVS setup, you can serve multiple service from a number of real servers, local or remote. If you are using LVS to load balance a group of Web servers (port 80), then all real servers must present identical contents, since the client could connect to any of them, over many connections/reconnections.

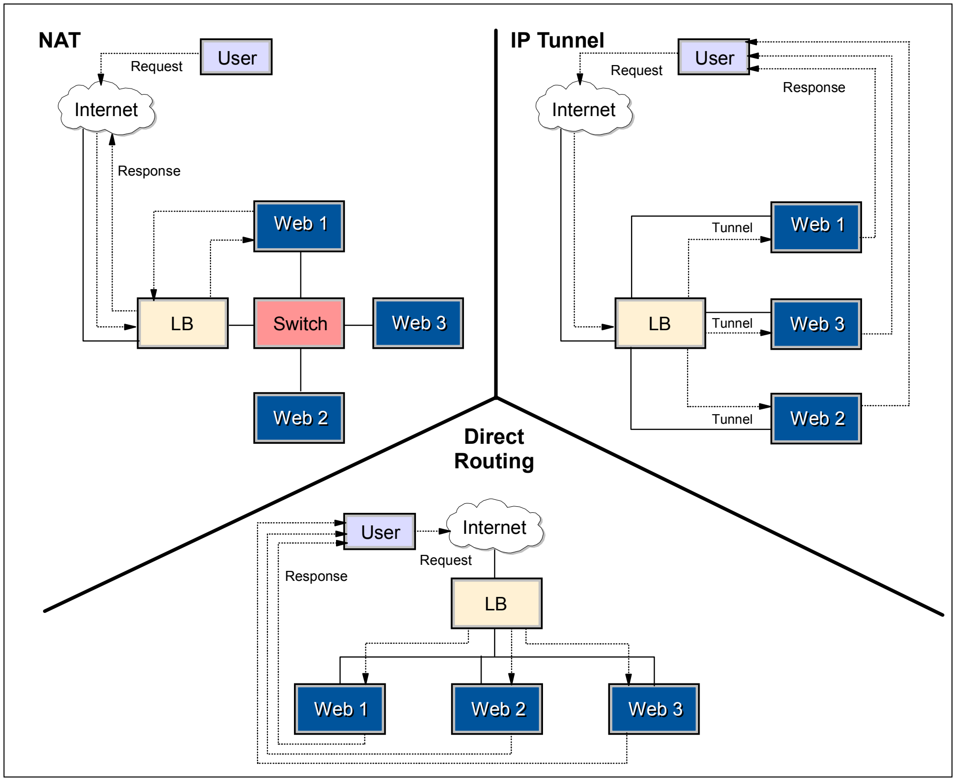


*Figure 8-10 Differences between the IPVS options*

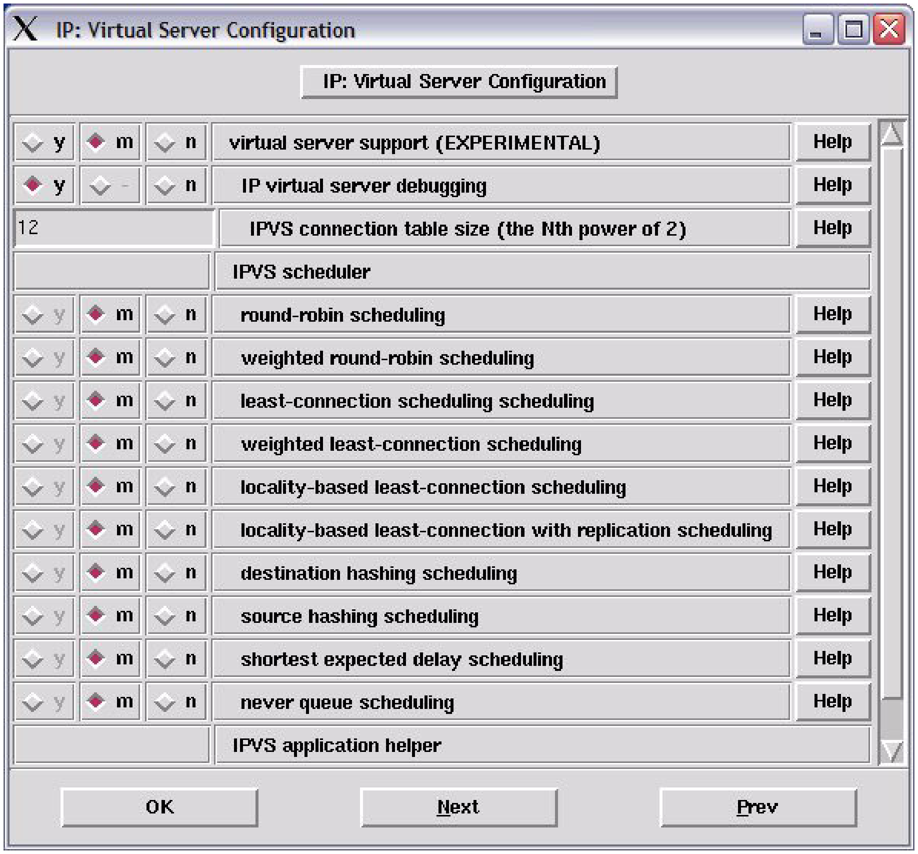
In our setup, we are using the direct routing option. To enable LVS in the kernel, we need to compile the kernel. First, we install the kernel source from the SLES 8 CD:

```
# rpm -ivh kernel-source-2.4.21-83.ppc.rpm
# cd /usr/src/linux
```

We now copy the existing config file of the current server setup and customize it to include IPVS:

```
# cp /boot/.config /usr/src/linux
# make xconfig
```

We select all the IPVS options to be compiled as modules. The IPVS options are inside the section Network Options under the tab IP: Virtual Server Configuration”. Figure 8-11 on page 386 shows the available options in the kernel configurator.


IP: Virtual Server Configuration
IP: Virtual Server Configuration
OK
Next
Prev

| y | m | n | Option | |
|---|---|---|---|---|
| ○ y | ● m | ○ n | virtual server support (EXPERIMENTAL) | Help |
| ● y | ○ - | ○ n | IP virtual server debugging | Help |
| 12 | | | IPVS connection table size (the Nth power of 2) | Help |
| | | | IPVS scheduler | |
| ○ y | ● m | ○ n | round-robin scheduling | Help |
| ○ y | ● m | ○ n | weighted round-robin scheduling | Help |
| ○ y | ● m | ○ n | least-connection scheduling scheduling | Help |
| ○ y | ● m | ○ n | weighted least-connection scheduling | Help |
| ○ y | ● m | ○ n | locality-based least-connection scheduling | Help |
| ○ y | ● m | ○ n | locality-based least-connection with replication scheduling | Help |
| ○ y | ● m | ○ n | destination hashing scheduling | Help |
| ○ y | ● m | ○ n | source hashing scheduling | Help |
| ○ y | ● m | ○ n | shortest expected delay scheduling | Help |
| ○ y | ● m | ○ n | never queue scheduling | Help |
| | | | IPVS application helper | |

*Figure 8-11 IP virtual server configuration*

After we select the IPVS options as modules, we recompile and install them. Note that we are not compiling the whole kernel; instead, we are just compiling additional modules to be part of the same kernel:

```
# make_ppc64.sh dep
# make_ppc64.sh modules
# mv /lib/modules/2.4.21-83-pseries64 /lib/modules/2.4.21-83-pseries.orig
# make_ppc64.sh modules_install
```

After we recompile the kernel, we set up heartbeat to manage the LVS. The hardware requirements for heartbeat are the same as for setting up the DB2 cluster. Make sure you have the necessary adapters and cables for heartbeat.

We install the heartbeat-l directord into the server:

```
# rpm -ivh heartbeat-1.0.4-10.ppc.rpm
# rpm -ivh heartbeat-ldirectord-1.0.4-10.ppc.rpm
```

We configure the authkeys and ha.cf files to define the nodes we have in the cluster. Refer to Example 8-5 on page 373 for details about how to set up the authkeys file. You can give a new key for this pair of nodes.

Example 8-16 displays lpar5 and lpar6 as our load balancer cluster. In this heartbeat setup, we are using a single adapter for both public and heartbeat. In a production environment, however, it is recommended that you have dedicated adapters for heartbeat.

*Example 8-16 /etc/ha.d/ha.cf file for load balancing*

```
# Logs definition
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local10

# HeartBeat Packets Configuration
keepalive 2 # time between each heartbeat
deadtime 30 # how long to declare dead
bcast eth0 # heartbeat communication link

# Resource Configuration
nice_failback on # this will turn on the feature cascading without fall-back

# Node Definition
node lpar5
node lpar6
```

The major configuration file of the ldirectord cluster is the ldirector.cf file. This file is placed into /etc/ha.d. Inside the file, you need to define the virtual IP addresses, the real IP addresses of the server, and the port to use.

Figure 8-17 shows the /etc/ha.d/ldirectord.cf file that we use.

*Example 8-17 /etc/ha.d/ldirectord.cf*

```
# Global Directives
checktimeout=10
checkinterval=2
fallback=127.0.0.1:80
autoreload=yes
logfile="/var/log/ldirectord.log"
logfile="local0"
quiescent=no

virtual=192.168.100.87:80
        real=192.168.100.83:80 gate
        real=192.168.100.84:80 gate
        service=http
        checkport=80
        request="/index.html"
        receive="Test Page"
        scheduler=rr
        protocol=tcp
```

As shown in Example 8-17, we specified 192.168.100.87 as the Web cluster IP address for external users. 192.168.100.83 and 192.168.100.84 are the IP addresses of lpar7 and lpar8, where we run Apache and PHP. We also specify the file used to test whether the server is up or down. For this setup, we are using the round robin scheduler. You can also manage the setup ldirectord to distribute packets based on the workload of the servers.

We now make ldirectord the resource that heartbeat will be managing. As shown in Example 8-18, we have specified that lpar5 is the primary node with the IP address 192.168.100.87. This corresponds to the address that we specified in the /etc/ha.d/ldirectord.cf file.

*Example 8-18 /etc/ha.d/haresources*

```
lpar5 192.168.100.87 ldirectord::/etc/ha.d/ldirectord.cf
```

Once we load the IPVS modules into the kernel, we start the heartbeat services:

**# insmod ip_vs**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs.o

**# insmod ip_vs_rr**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_rr.o

**# insmod ip_vs_sh**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_sh.o

**# insmod ip_vs_lc**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_lc.o

**# insmod ip_vs_lblcr**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_lblcr.o

**# insmod ip_vs_wlc**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_wlc.o

**# insmod ip_vs_sed**

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_sed.o

```
# insmod ip_vs_nq

Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_nq.o
# insmod ip_vs_ftp
Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_ftp.o
# insmod ip_vs_lblc
Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_lblc.o
# insmod ip_vs_wrr
Using /lib/modules/2.4.21-83-pseries64/kernel/net/ipv4/ipvs/ip_vs_wrr.o
```

After successfully loading the modules, we check the lsmod table; the modules should be listed in the table. We update our IPVS clients (lpar7 and lpar8) to be IPVS-ready. We create an IP address of the cluster at loopback and hide the adapter from broadcasting any ARP. Example 8-19 shows what we are running in lpar7 and lpar8.

*Example 8-19 Updating client to be IPVS-ready*

```
[root@lpar7 root]# echo "0" >/proc/sys/net/ipv4/ip_forward
[root@lpar7 root]# sbin/ifconfig lo:0 192.168.100.87 broadcast 192.168.100.87
netmask 0xffffffff up

[root@lpar7 root]# /sbin/route add -host 192.168.100.87 dev lo:0
[root@lpar7 root]# echo "1" >/proc/sys/net/ipv4/conf/all/hidden
[root@lpar7 root]# echo "1" >/proc/sys/net/ipv4/conf/lo/hidden
```

Now we can start the heartbeat software using the command /etc/init.d/**heartbeat start**. This automatically creates the cluster IP addresses and starts the ldirectord to act as our load balancers.

Next, let us examine the IP virtual server table in Example 8-20. We notice that the incoming packets to 192.168.100.87 automatically get distributed to lpar7 and lpar8.

*Example 8-20 IP virtual server routing table*

```
IP Virtual Server version 1.0.10 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.100.87:http rr
  -> lpar7:http                   Route   1      0          0
  -> lpar8:http                   Route   1      0          0
```

### Testing the load balancer cluster

We run a command to loop the request for display to the db2_php.test script that queries the DB2 cluster directly. The request does a round robin, the first request to lpar7, then lpar8, and so on.

We also test that if lpar7 is switched off or taken down for management, all traffic goes to lpar8. Example 8-21 shows the test of querying lpar7 and lpar8, which creates DB2 connections to the DB2 cluster we have in lpar1 and lpar3.

*Example 8-21 Testing IPVS cluster*

```
# while :; do lynx -dump webip/db2_test.php; sleep 1; done

   This is from LPAR7 !! db2 connection : Resource id #3 Connection
   Successful

   ID   NAME   DEPT  JOB  YEARS  SALARY   COMM
   10  Sanders 20   Mgr   7     18357.50 NULL
   20  Pernal  20   Sales 8     18171.25 612.45
   80  James   20   Clerk NULL  13504.60 128.20
   190 Sneider 20   Clerk 8     14252.75 126.50

   This is from LPAR8 !! db2 connection : Resource id #3 Connection
   Successful

   ID   NAME   DEPT  JOB  YEARS  SALARY   COMM
   10  Sanders 20   Mgr   7     18357.50 NULL
   20  Pernal  20   Sales 8     18171.25 612.45
   80  James   20   Clerk NULL  13504.60 128.20
   190 Sneider 20   Clerk 8     14252.75 126.50
```

With this setup, lpar7 and lpar8 are fully redundant. We can also add new Web servers into the configuration easily by modifying the ldirectord.cf file and reloading it. Figure 8-12 on page 391 shows our current landscape with the option to set up additional Web servers when required.

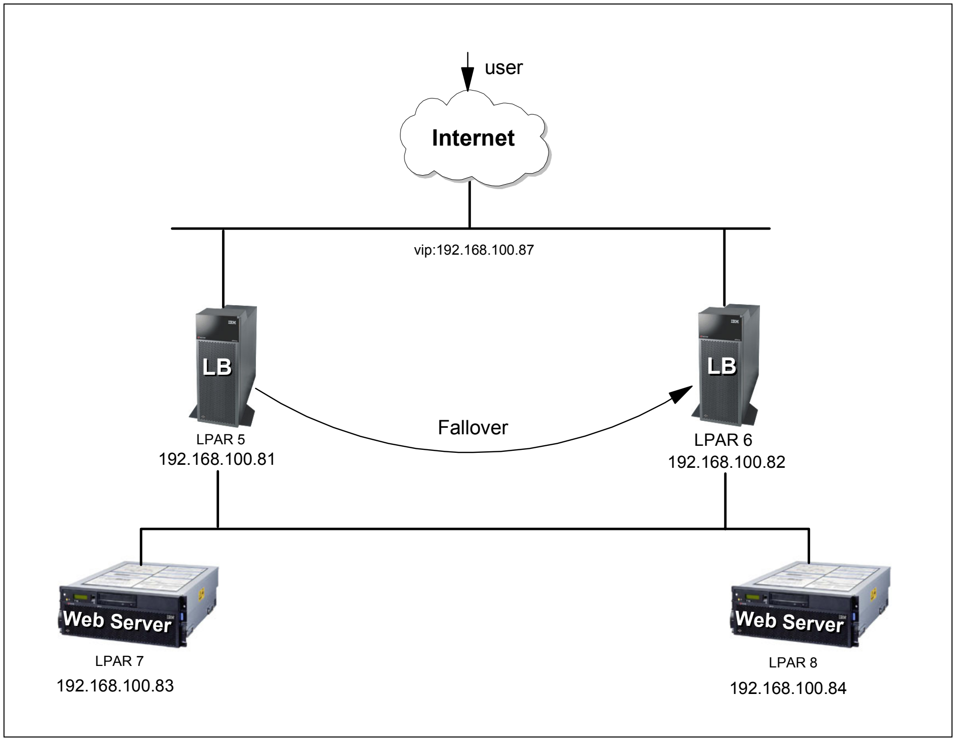


*Figure 8-12 Load Balancers (LB) with two Web servers*

> **Tip:** Suppose you run the command **ipvsadm -L** and see the following messages:
>
> ```
> Can't initialize ipvs: Protocol not available.
> Are you sure that IP Virtual Server is built in the kernel or as module?
> ```
>
> These messages mean that the kernel does not have the IPVS modules made available to it.

## 8.1.8 Final landscape of the setup

In the final setup, we have two servers running ldirectord acting as load balancing two Web servers and two clustered database servers. This is a fully redundant solution where both the Web servers are actively talking to the database servers. The final setup looks like the diagram in Figure 8-13 on page 392.

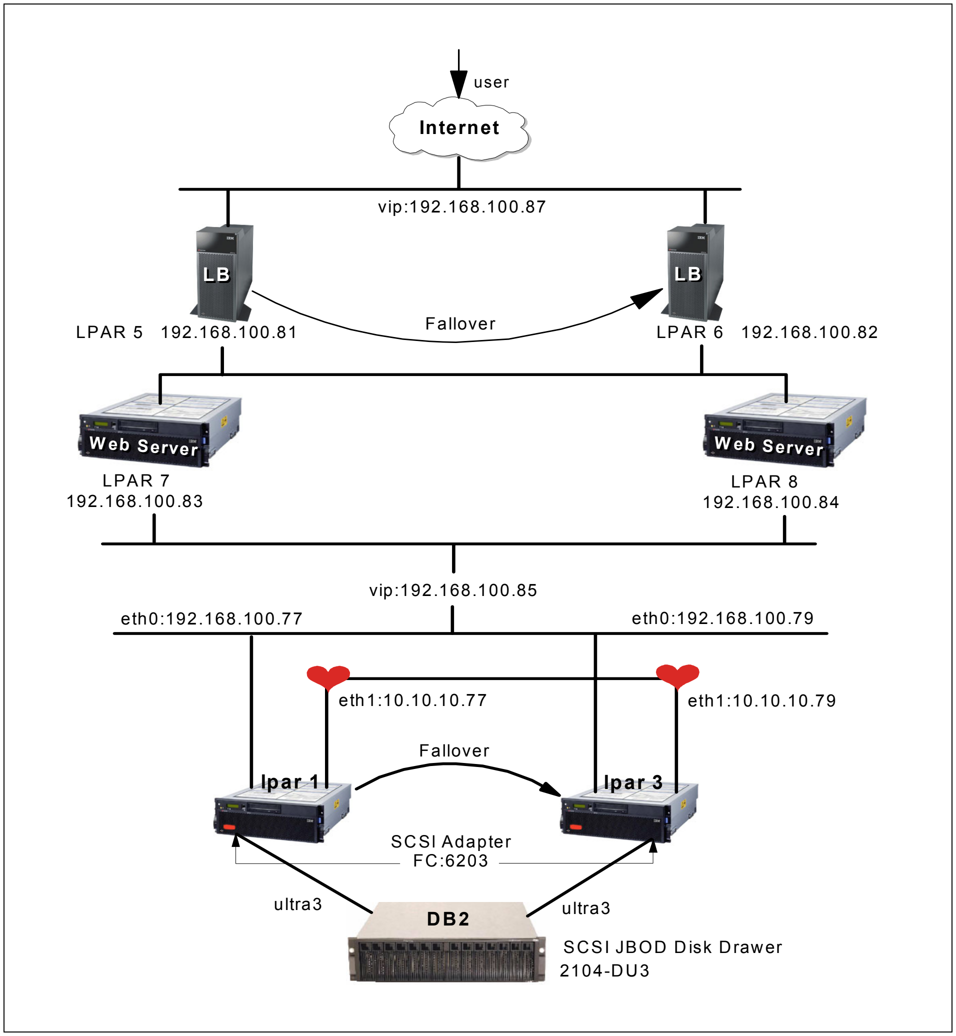


*Figure 8-13 Final cluster diagram*

## 8.1.9 Alternate solutions to improve the cluster

In this section, we discuss alternate solutions to improve the cluster.

### Other disk solution

One critical improvement that we would like to see in the cluster is the use of Fibre Channel disks instead of SCSI disks. The IBM Fibre Channel adapter (F/C 6228) is compatible with the lpfcdd driver.

We load it by running the command:

```
# modprobe lpfcdd
```

After it is loaded, we can create the disk partitions and file systems using steps that are similar to those used in the SCSI solution; refer to “Disk setup” on page 364.

### Other software stack

Besides using Apache and PHP, we can also use WebSphere as the middleware. WebSphere is a J2EE application server supported on Linux for pSeries, and it offers a whole suite of applications and portlets.

### Other clustering solutions

- Distributed Replicated Block Device (DRBD)

  DRBD is designed to mirror a whole block device via a network. It basically takes the data and ships it across to the other node through a network. On the remote node, the data will be written to the disk. This essentially is Network RAID-1 for storage. This is ideal for Web servers, ftp servers and database servers for data reliability. Using DRDB with a clustering solution allows you to have the most recent and up-to-date data through real time mirroring.

- General Parallel File system (GPFS)

  GPFS is a shared-disk file system that provides data access for GPFS clients. The data can be spanned across multiple GPFS nodes and the files can be accessed concurrently from multiple nodes. GPFS provides great performance and availability through logging and replication. It also offers options which can be configured for failover from both disk and server malfunctions.

- Other clustering solution

  While heartbeat is good for a small cluster, it lacks many of the enterprise features to handle disk quorum, different types of clusters, and so on. For enterprise customers, you can also try Tivoli System Automation.

  Heartbeat is the cluster solution bundled with SLES 8. For any other open source clustering solutions, you might need to compile the software products manually to run and work in SLES 8.

## 8.2 High Availability with Red Hat ClusterSuite

Red Hat ClusterSuite is an optional product that you can purchase for your Red Hat Advanced Server (RHAS) 3. ClusterSuite provides an application failover infrastructure with up to eight nodes.

## 8.2.1 Red Hat ClusterSuite

Red Hat Advanced Server 3.0 offers a high availability clustering solution using its own ClusterSuite package. ClusterSuite is a set of RPM packages that are easy to set up and easy to use for managing HA clusters. We tested ClusterSuite software by setting up an highly available Apache server running in a simple cluster environment. Our goal was mainly to test the basic clustering environment with one simple application, one service IP, and one shared file system with defined http DocumentRoot.

Red Hat ClusterSuite offers features that are not available on SuSE Linux, such as disk-based heartbeat and an easy-to-use graphical interface.

### Installing ClusterSuite

The ClusterSuite package consists of the following RPM packages:

- clumanager-1.1.73-1.ppc.rpm
- ipvsadm-1.21-8.ppc.rpm
- piranha-0.7.4.1.ppc.rpm
- rh-cs-en.rpm
- redhat-config-cluster-0.1.81-1.noarch.rpm

If you have ClusterSuite on a CD-ROM and are using a graphical desktop, the CD is launched automatically when inserted, using the Package Management Tool. Install the software using either the Package Management Tool or the command line.

### Setting up a shared disk environment

Before you start the cluster configuration, the shared disk environment must be set up and tested to ensure that the shared disk can be seen from both nodes of the cluster, and that the shared file system can be mounted from both nodes.

ClusterSuite uses two raw partitions around 10 MB in size to run heartbeats and store cluster state information. Our test environment, and the steps needed to create and prepare the shared disk, are described here.

We used two logical partitions, lpar6 and lpar8, in two different pSeries servers running Red Hat Enterprise Advanced Server 3.0.

Two lpars were connected over a 10/100 ethernet network and connected to an external scsi disk array with two 36 GB disks. We configured everything on lpar6 and replicated it to lpar8.

1. We changed the SCSI ID of the disk by editing the file /etc/modules.conf with the syntax in Example 8-22 on page 395.

*Example 8-22 Modules.conf*

```
[leecy@lpar6 etc]# cat modules.conf
alias eth0 e100
alias scsi_hostadapter sym53c8xx
options sym53c8xx       sym53c8xx=hostid:5 ## <-- Replace 5 with your SCSI ID
```

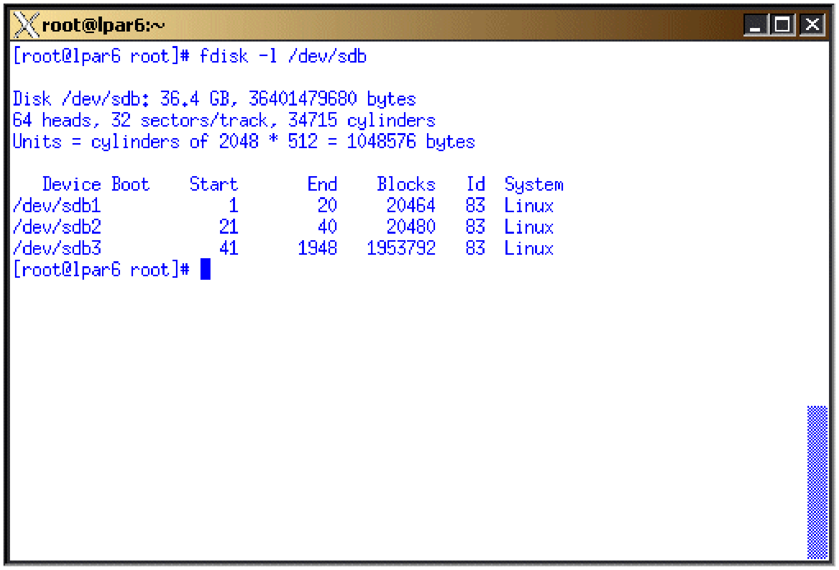
Next, we recreated an initrd and appended it into /etc/yaboot.conf. Example 8-23 shows the commands **`mkinitrd`** and **`yaboot.conf`**. Then we ran the command **`ybin`**.

*Example 8-23 Recreating a new initrd for yaboot to use*

```
[root@lpar6 etc]# mkinitrd /boot/initrd-13112003 2.4.21-4.EL
[root@lpar6 etc]# cat /etc/yaboot.conf
# yaboot.conf generated by anaconda

boot=/dev/sda1
init-message=Welcome to Red Hat Enterprise Linux AS\!
Hit <TAB> for boot options

partition=2
timeout=30
install=/usr/lib/yaboot/yaboot
delay=10
nonvram

image=/vmlinux-2.4.21-4.EL
        label=linux
        read-only
        initrd=/initrd-13112003
        append="console=hvc0 root=LABEL=/"
[root@lpar6 etc]#
```

2. We created two primary raw partitions of 20 MB each, using fdisk on the shared disk device /dev/sdb.
3. We created one extended shared partition of 2 GB, using fdisk on the shared disk device.

   The fdisk -l /dev/sdb output is shown in Figure 8-14 on page 396.



```
[root@lpar6 root]# fdisk -l /dev/sdb

Disk /dev/sdb: 36.4 GB, 36401479680 bytes
64 heads, 32 sectors/track, 34715 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1        20     20464   83  Linux
/dev/sdb2            21        40     20480   83  Linux
/dev/sdb3            41      1948   1953792   83  Linux
[root@lpar6 root]# 
```

*Figure 8-14 fdisk -l /dev/sdb*

4. We created an ext2 shared file system on the extended partition:

   ```
   #mke2fs -j -b 4096 /dev/sdb3
   # mount /dev/sdb3 /itso_share
   ```

5. We modified the /etc/sysconfig/rawdevices file with the new device information. Example 8-24 shows the contents.

*Example 8-24 /etc/sysconfig/rawdevices*

```
/dev/raw/raw1 /dev/sdb1
/dev/raw/raw2 /dev/sdb2
```

6. We copied the file to lpar8:

   ```
   #scp /etc/sysconfig/rawdevices root@lpar8:/etc/sysconfig
   ```

7. We started the rawdevices on both lpars:

   ```
   #service rawdevices restart
   ```

8. We queried the devices by using the **`raw -aq`** command. Example 8-25 shows the output.

*Example 8-25 raw -aq*

```
/dev/raw/raw1 bound to major8, minor 17
/dev/raw/raw2 bound to major8, minor 18
```

9. We unmounted the mounted shared file system on lpar6, and mounted it on lpar8 to test whether it was accessible. There is no need to add this to /etc/fstab, as this is controlled by the cluster manager.

> **Note:** It is mandatory to complete shared disk configuration *before* you configure the cluster, because cluster configuration requires the raw device definitions.

For detailed information on connecting shared disk systems, single and multi-channel SCSI intiators, and planning and configuring shared disks, refer to the guide *Red Hat Cluster Suite: Configuring and Manging a Cluster*.

## Configuring the cluster

A Red Hat cluster is configured using the **`redhat-config-cluster`** command, which starts a graphical interface. Our simple, two-node test cluster is shown in tFigure 8-15.

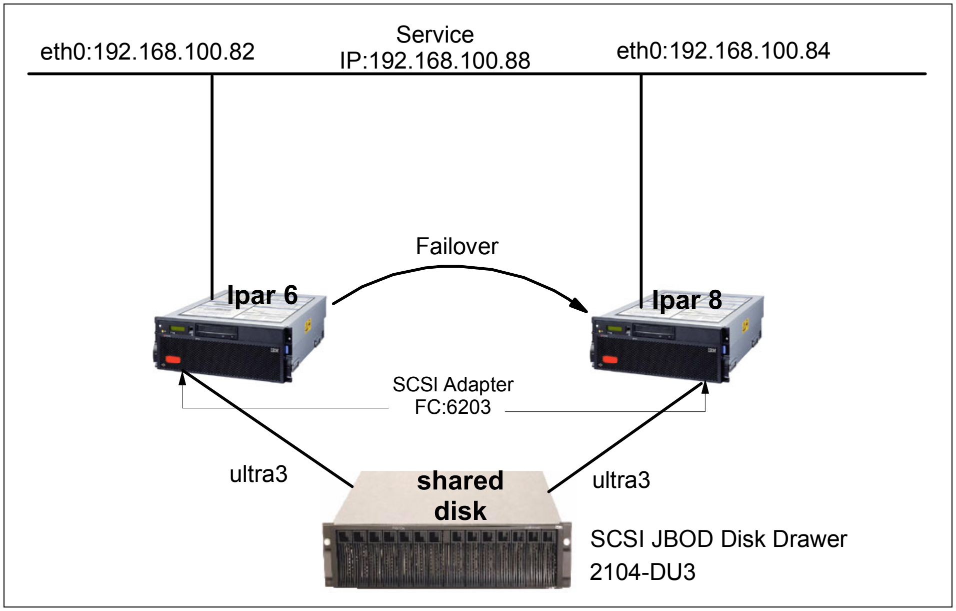


*Figure 8-15 Simple test cluster*

You can set up this test cluster as follows:

1. Update /etc/hosts or the DNS server with hostnames of cluster member nodes and any other servers that communicate with cluster members.

2. Start the graphical interface with redhat-cluster-config. Figure 8-16 shows the graphical screen opened.

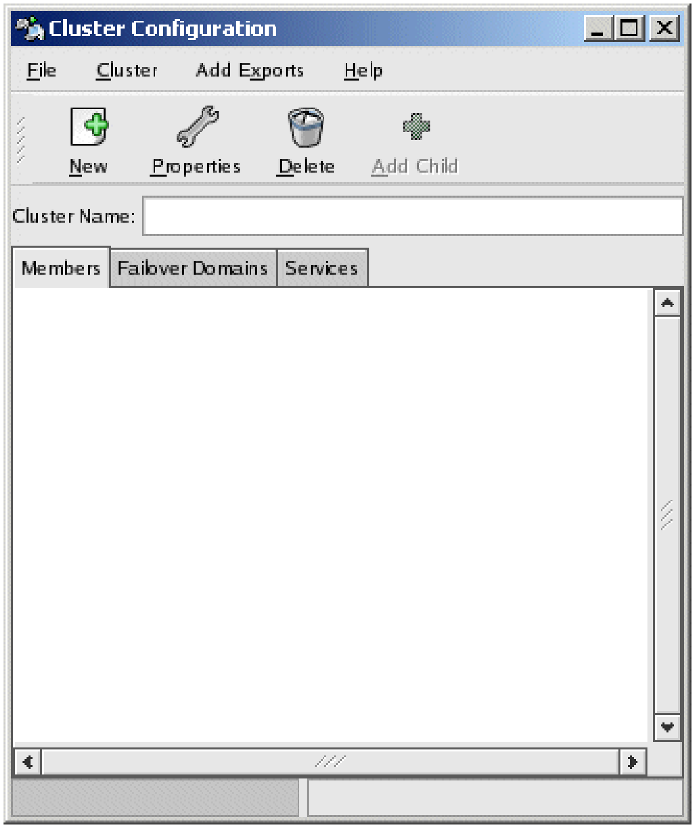


*Figure 8-16 redhat-config-cluster screen*

3. Enter a clustername, such as: cluster-itso.
4. Click New while highlighting the “Members” tab on the window to add a new member. Figure 8-17 shows the member definition window. Type in: `lpar6` and repeat the same to add lpar8.

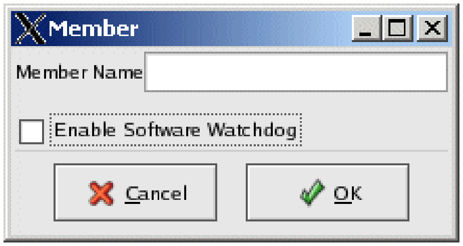


*Figure 8-17 Add member window*

5. Switch to the Failover domains tab and click New to add a new failover domain. Select Domain name as “http-domain”. Click Add Members and select members lpar6 and lpar8. Select check boxes to restrict failover to only these members and ordered failover.

This controls failover with lpar6 as the primary node in this domain and lpar8 is failover node. If lpar6 fails, service is failed over to lpar8. When lpar6 comes back up, service is moved back to lpar6. Figure 8-18 shows the window to add a new failover domain.

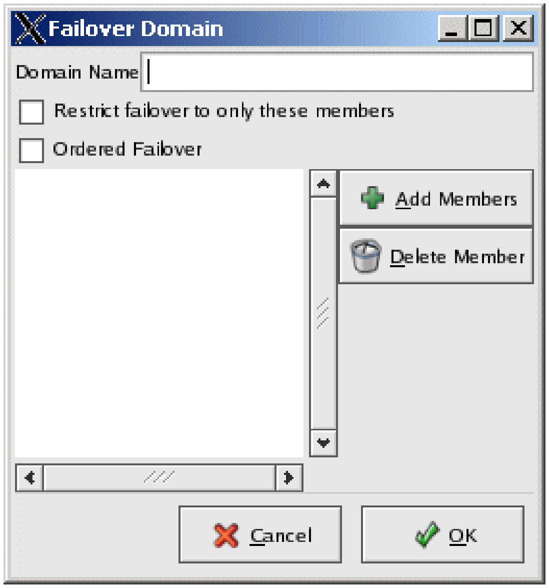
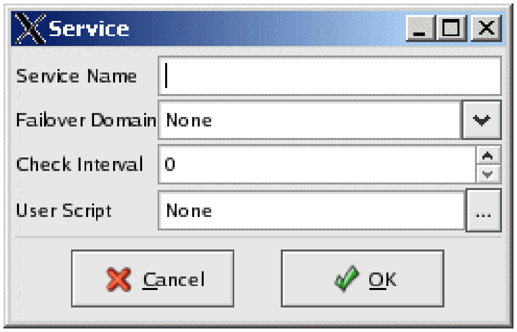


*Figure 8-18 Add failover domain window*

6. Switch to theServices tab and click New to define a new service named http. Define the service name as “http”, select the Failover Domain as

"http-domain", select the Check interval as 10 Seconds, and the User script as "/etc/rc.d/init.d/http". This adds "http" service as a shared service.

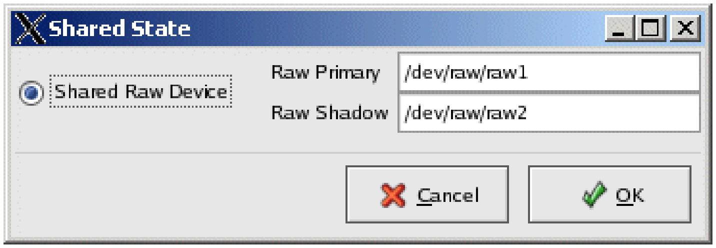
Figure 8-19 on page 400 shows the window to add a new Service.



Service Name:
Failover Domain: None
Check Interval: 0
User Script: None

*Figure 8-19 Add http service*

7. Confirm the default shared state devices by clicking the Cluster-Shared State Menu. Accept the default raw devices identified, or change them as necessary. /etc/sysconfig/rawdevices must be updated prior to this step and rawdevices services must be started. Figure 8-20 shows the window to confirm the raw devices.

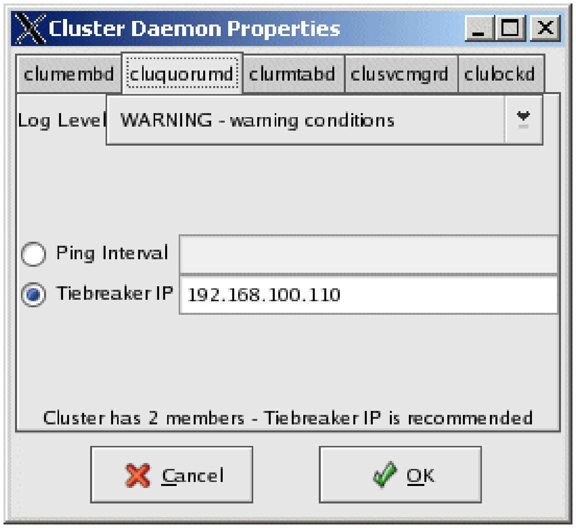


● Shared Raw Device
Raw Primary: /dev/raw/raw1
Raw Shadow: /dev/raw/raw2

*Figure 8-20 Shared State raw devices*

8. In the case of a two-node cluster, an IP such as a router or gateway IP on the same subnet is required to confirm network availability during a problem; this is known as a tiebreaker IP.

   To define a tiebreaker IP, open the cluster daemons window by using the Cluster-Cluster Daemon Properties Menu; see Figure 8-21 on page 401. We used IP 192.168.100.110 for our tiebreaker IP.



Log Level: WARNING - warning conditions

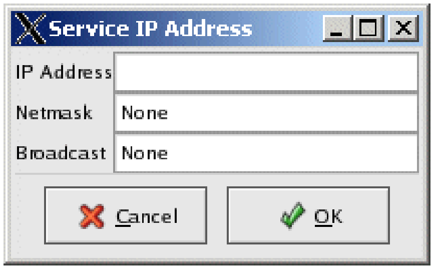
○ Ping Interval

● Tiebreaker IP: 192.168.100.110

Cluster has 2 members - Tiebreaker IP is recommended

*Figure 8-21 Tiebreaker IP window*

9. Add a service IP to the defined “http” service in the failover domain “http-domain” by highlighting the http service on the main screen and selecting **Add Child**. Define the Service IP and netmask in the window displayed. This IP is failed over with http service incase of a problem. Figure 8-22 shows the window.



| | |
|---|---|
| IP Address | |
| Netmask | None |
| Broadcast | None |

*Figure 8-22 Adding Service IP*

10. Add the shared file system to http service by highlighting http service and selecting **Add Child-Add device**. Define /dev/sdb3 as a special device file, and /itso_sharing as Mount point ext2 as FS type. Check Force Unmount; this forcefully releases the file system before failing over, instead of waiting for the release; see Figure 8-23 on page 402.

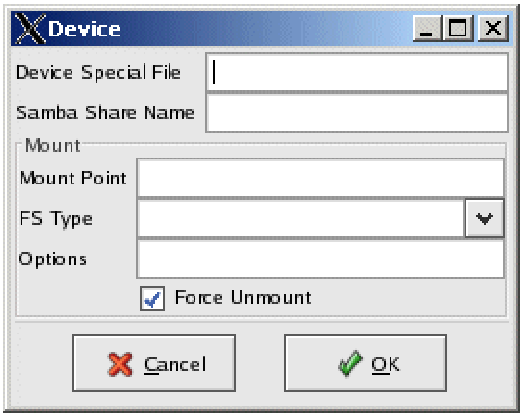


*Figure 8-23 Add shared device window*

11. Select **File-Save** to save the config. The config is written to /etc/cluster.xml file. Copy the config file to lpar8:

```
# scp /etc/cluster.xml root@lpar8:/etc
```

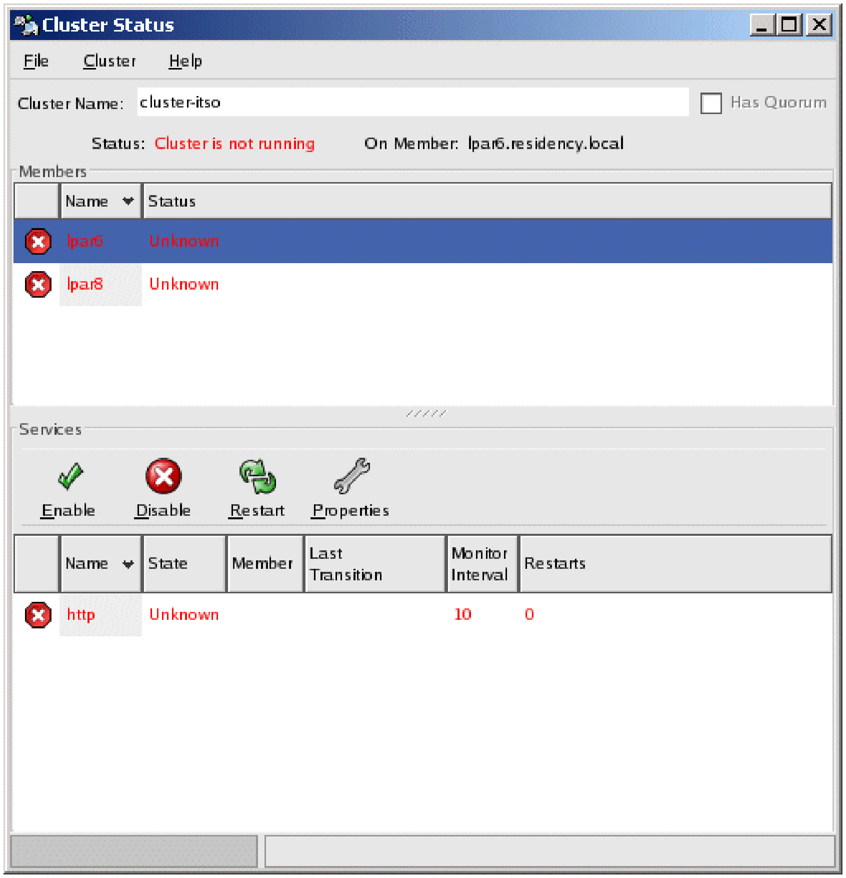
12. Enable cluster logging by modifying the /etc/syslog.conf file:

```
local4.* /var/log/cluster.log
```

Refresh the syslog daemon:

```
#service syslog restart
```

13. Close the cluster config window and then reopen it again by typing: `redhat-cluster-config`. This time the cluster window displays the cluster status screen instead of configuration; see Figure 8-24 on page 403.



Cluster Name: cluster-itso ☐ Has Quorum

Status: Cluster is not running On Member: lpar6.residency.local

Members

| | Name | Status |
|---|---|---|
| ☒ | lpar6 | Unknown |
| ☒ | lpar8 | Unknown |

Services

| | Name | State | Member | Last Transition | Monitor Interval | Restarts |
|---|---|---|---|---|---|---|
| ☒ | http | Unknown | | | 10 | 0 |

*Figure 8-24 Cluster status window before starting cluster services*

14. To modify any cluster configuration settings, select the Cluster-Configure menu.

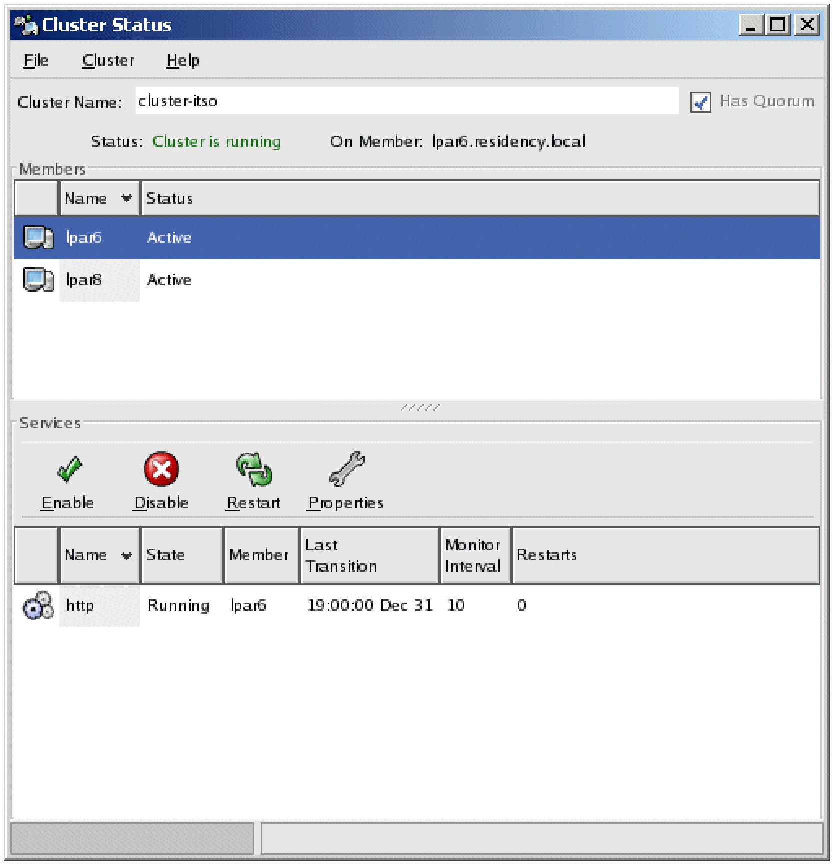
This completes the simple cluster configuration. We noticed that the cluster configuration became automatically synchronized from lpar6 to lpar8 after creating the resources on one side.

## Testing the cluster

You can test the configured cluster for simple failovers, as follows:

1. Start the cluster by selecting **Cluster-Start Local Cluster Daemons** first on lpar6, and then on lpar8, by using the cluster status graphical window. Once the cluster is active, it starts the cluster daemons, starts the http service,

mounts the shared file system and aliases the service IP on lpar6. Figure 8-25 shows the status window when the cluster is active.



Cluster Name: cluster-itso ☑ Has Quorum

Status: Cluster is running On Member: lpar6.residency.local

Members

| Name | Status |
|---|---|
| lpar6 | Active |
| lpar8 | Active |

Services

| Name | State | Member | Last Transition | Monitor Interval | Restarts |
|---|---|---|---|---|---|
| http | Running | lpar6 | 19:00:00 Dec 31 | 10 | 0 |

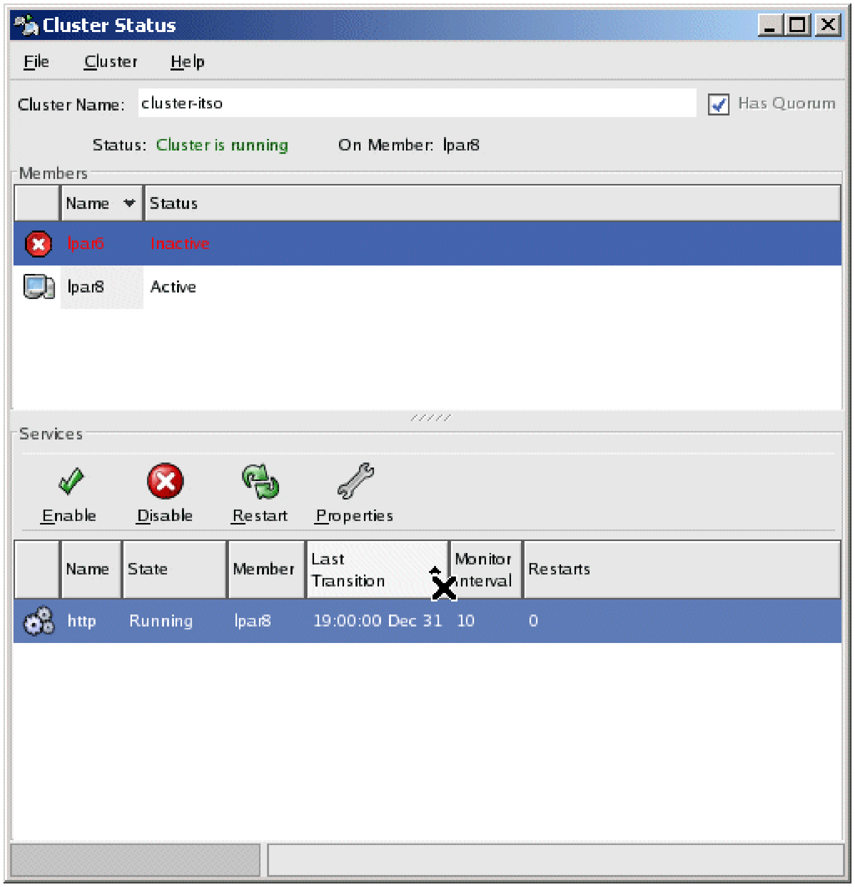
*Figure 8-25 Active Cluster status*

If you prefer to use the command line, you can run the command **clustat**. This will show a quick status of your cluster; see Example 8-26.

*Example 8-26 Using the command line to get cluster status*

```
[root@lpar8 root]# clustat
Cluster Status - cluster-itso                                          13:50:57
Quorum: Yes, view 3
Shared State: Shared Raw Device Driver v1.0 [Min. Size=1176064]

  Member             Status
  ------------------ ----------
  lpar6              Active
  lpar8                Active     <-- You are here

  Service        Status   Owner (Last)     Last Transition Chk Restarts
  -------------- -------- ---------------- --------------- --- --------
  http           started  lpar6            19:00:00 Dec 31  60        0
[root@lpar8 root]#
```

2. Test the node failover by stopping lpar6 either by init 6, or resetting it using the Hardware Management Console (HMC). All services fail over to the active node, which is lpar8. Figure 8-26 shows the services active on lpar8 when lpar6 is shut down.



Cluster Name: cluster-itso ☑ Has Quorum

Status: Cluster is running On Member: lpar8

Members

| Name | Status |
|---|---|
| lpar6 | Inactive |
| lpar8 | Active |

Services

| Name | State | Member | Last Transition | Monitor Interval | Restarts |
|---|---|---|---|---|---|
| http | Running | lpar8 | 19:00:00 Dec 31 | 10 | 0 |

*Figure 8-26 Failover cluster status*

3. Power on lpar6 and start cluster services. When the lpar6 cluster manager is up, lpar8 releases the shared services and lpar6 takes over the resources.

As mentioned, ClusterSuite can help you to set up and bring up a highly available cluster quickly. Virtual load balancing clusters can also be set up with Red Hat ClusterSuite software. This is called Linux Virtual Servers (LVS) and is used primarily to load balance requests coming from the Internet to a pool of back-end real servers used extensively in high traffic e-business environments.

For more detailed information on ClusterSuite, supporting hardware/software, detailed install/configuration instructions and setting up sample HA applications, refer to *Red Hat Cluster Suite: Configuring and Managing a cluster* from Red Hat.

# 8.3 Tivoli Storage Manager (TSM)

IBM Tivoli Storage Manager (TSM) is a centralized, comprehensive backup and recovery solution that employs smart data move and smart data store technology, which makes backups and restores fast and flexible. The IBM Tivoli suite of storage products supports more than a dozen operating system platforms, a variety of network connectors, and more than 500 offline storage devices. For more information, refer to :

http://www-3.ibm.com/software/tivoli/solutions/storage/backup/

In this section, we describe the basic TSM installation process on SLES 8. For more information, refer to the IBM Redbook IBM Tivoli Storage Manager Implementation Guide, SG24-5416.

You can download a 60-day free trial copy of IBM Tivoli Storage Resource Manager from:

http://www.ibm.com/software/tivoli/resource-center/storage/code-srm.jsp

## 8.3.1 TSM server installation

This section describes the necessary steps to complete TSM server installation.

### Prerequisites

At the time of writing, only SuSE SLES 8 for pSeries is supported.

### Package installation

For the server installation, we had following packages in the ../TSMServer.V52/ppc64 directory:

TIVsm-license-5.2.0-0.ppc64.rpm

TIVsm-server-5.2.0-0.ppc64.rpm

TIVsm-license_keys-5.2.0-0.ppc64.rpm

TIVsm-tsmscsi-5.2.0-0.ppc64.rpm

We had the following packages in the ../TSMServer.V52/noarch:

TIVsm-webadmin-5.2.0-0.noarch.rpm

TIVsm-webhelp.de_DE-5.2.0-0.noarch.rpm

TIVsm-webhelp.en_US-5.2.0-0.noarch.rpm

TIVsm-webhelp.es_ES-5.2.0-0.noarch.rpm

... other languages

First, we use rpm to install server and license rpms, as shown in Example 8-27.

*Example 8-27   Installing TSM server-rpms*

```
# rpm -ivh TIVsm-license-5.2.0-0.ppc64.rp
m TIVsm-license_keys-5.2.0-0.ppc64.rpm TIVsm-server-5.2.0-0.ppc64.rpm
TIVsm-tsmscsi-5.2.0-0.ppc64.rpm
TIVsm-license               ##################################################
TIVsm-license_keys          ##################################################
TIVsm-server                ##################################################
Allocated space for db.dsm: 17825792 bytes
Allocated space for log.dsm: 9437184 bytes

Tivoli Storage Manager for Linux/ppc64
Version 5, Release 2, Level 0.0

Licensed Materials - Property of IBM

(C) Copyright IBM Corporation 1990, 2003. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 10:28:25 on Jun 13 2003.
ANR7801I Subsystem process ID is 32092.
ANR0905W Options file dsmserv.opt not found.
ANR0300I Recovery log format started; assigned capacity 8 megabytes.
ANR0301I Recovery log format in progress; 4 megabytes of 8.
ANR0301I Recovery log format in progress; 8 megabytes of 8.
ANR0302I Recovery log formatting took 19 milliseconds.
ANR0303I Format rate: 107789.5 pages/second.
ANR0304I Page service time:     0.0 ms.
ANR0305I Recovery log format complete.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
```

```
ANR0352I Transaction recovery complete.
ANR0992I Server installation complete.
Allocated space for backup.dsm: 10485760 bytes
Allocated space for archive.dsm: 5242880 bytes

Tivoli Storage Manager for Linux/ppc64
Version 5, Release 2, Level 0.0

Licensed Materials - Property of IBM

(C) Copyright IBM Corporation 1990, 2003. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 10:28:25 on Jun 13 2003.
ANR7801I Subsystem process ID is 32112.
ANR0900I Processing options file dsmserv.opt.
ANR0990I Server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 16 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR1636W The server machine GUID changed: old value (), new value (00.00.00.00
.0f.e0.11.d8.b5.7d.00.02.55.3a.06.2c).
ANR2100I Activity log process has started.
ANR4726I The NAS-NDMP support module has been loaded.
ANR2803I License manager started.
ANR2560I Schedule manager started.
ANR0984I Process 1 for AUDIT LICENSE started in the BACKGROUND at 05:35:18 PM.
ANR2820I Automatic license audit started as process 1.
ANR2200I Storage pool BACKUPPOOL defined (device class DISK).
ANR2200I Storage pool ARCHIVEPOOL defined (device class DISK).
ANR2200I Storage pool SPACEMGPOOL defined (device class DISK).
ANR0993I Server initialization complete.
ANR0916I TIVOLI STORAGE MANAGER distributed by Tivoli is now ready for use.
ANR2825I License audit process 1 completed successfully - 0 nodes audited.
ANR0985I Process 1 for AUDIT LICENSE running in the BACKGROUND completed with
completion state SUCCESS at 05:35:18 PM.
ANR2060I Node CLIENT registered in policy domain STANDARD.
ANR2099I Administrative userid CLIENT defined for OWNER access to node CLIENT.
ANR2068I Administrator ADMIN registered.
ANR2076I System privilege granted to administrator ADMIN.
ANR2206I Volume /opt/tivoli/tsm/server/bin/backup.dsm defined in storage pool
BACKUPPOOL (device class DISK).
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/backup.dsm varied online.
ANR2206I Volume /opt/tivoli/tsm/server/bin/archive.dsm defined in storage pool
```

```
ARCHIVEPOOL (device class DISK).
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/archive.dsm varied online.

************************************************************
IMPORTANT: Read the contents of file /README
           for extensions and corrections to printed
           product documentation.
************************************************************
TIVsm-tsmscsi                ##################################################
```

Now, we can also install TIVsm-webadmin-5.2.0-0.noarch.rpm, located in the noarch directory as shown in Example 8-28.

*Example 8-28 Installing webadmin*

```
# rpm -Uvh TIVsm-webadmin-5.2.0-0.noarch.rpm
TIVsm-webadmin               ##################################################

Tivoli Storage Manager for Linux/ppc64
Version 5, Release 2, Level 0.0

Licensed Materials - Property of IBM

(C) Copyright IBM Corporation 1990, 2003. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 10:28:25 on Jun 13 2003.
ANR7801I Subsystem process ID is 184.
ANR0900I Processing options file dsmserv.opt.
ANR0990I Server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 16 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR1635I The server machine GUID, 00.00.00.00.0f.e0.11.d8.b5.7d.00.02.55.3a.06
.2c, has initialized.
ANR2100I Activity log process has started.
ANR4726I The NAS-NDMP support module has been loaded.
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/backup.dsm varied online.
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/archive.dsm varied online.
ANR2803I License manager started.
ANR2560I Schedule manager started.
ANR0993I Server initialization complete.
```

```
ANR0916I TIVOLI STORAGE MANAGER distributed by Tivoli is now ready for use.
ANR4693I Interface Driver information will be loaded in quiet mode: Only
warning and error messages will be displayed.
ANR4980I Auditing Interface Driver definitions.
ANR4983I Auditing Interface Driver Groups.
ANR4985I Auditing Interface Driver Group Members.
ANR4986I Auditing Interface Driver Classes.
ANR4988I Auditing Interface Driver Complex Class containers.
ANR4991I Auditing Interface Driver Tasks.
ANR4992I Auditing Interface Driver Task Members.
ANR4989I Auditing Interface Driver Operations.
ANR4990I Auditing Interface Driver Operation Parameters.
ANR4982I Interface Driver audit completed - definitions are consistent.
```

We edit the opt/tivoli/tsm/server/bin/dsmserv.opt file and add the line “HTTPPORT 1580”, as shown in Example 8-29.

*Example 8-29 Editing /opt/tivoli/tsm/server/bin/dsmserv.opt*

```
*** IBM TSM Server options file
*** Refer to dsmserv.opt.smp for other options
COMMMETHOD TCPIP
TCPPORT 1500
HTTPPORT 1580
DEVCONFIG devcnfg.out
```

We can change to /opt/tivoli/tsm/server/bin, start the Tivoli server for the first time, and run it in the foreground first as shown in Example 8-30.

*Example 8-30 Starting the TSM server*

```
lpar5:/opt/tivoli/tsm/server/bin # ./dsmserv

Tivoli Storage Manager for Linux/ppc64
Version 5, Release 2, Level 0.0

Licensed Materials - Property of IBM

(C) Copyright IBM Corporation 1990, 2003. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 10:28:25 on Jun 13 2003.
ANR7801I Subsystem process ID is 7558.
ANR0900I Processing options file dsmserv.opt.
ANR0990I Server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 16 megabytes.
```

```
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR1635I The server machine GUID, 00.00.00.00.0f.e0.11.d8.b5.7d.00.02.55.3a.06
.2c, has initialized.
ANR2100I Activity log process has started.
ANR4726I The NAS-NDMP support module has been loaded.
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/backup.dsm varied online.
ANR1305I Disk volume /opt/tivoli/tsm/server/bin/archive.dsm varied online.
ANR0984I Process 1 for EXPIRATION started in the BACKGROUND at 05:52:16 PM.
ANR0811I Inventory client file expiration started as process 1.
ANR2803I License manager started.
ANR8200I TCP/IP driver ready for connection with clients on port 1500.
ANR2560I Schedule manager started.
ANR0812I Inventory file expiration process 1 completed: examined 0 objects,
deleting 0 backup objects, 0 archive objects, 0 DB backup volumes, and 0
recovery plan files. 0 errors were encountered.
ANR0985I Process 1 for EXPIRATION running in the BACKGROUND completed with
completion state SUCCESS at 05:52:16 PM.
ANR0993I Server initialization complete.
ANR0916I TIVOLI STORAGE MANAGER distributed by Tivoli is now ready for use.
TSM:SERVER1>
```

**Tip:** If we start the server in the foreground, we can administer it in the terminal window we started.

If the server runs in the background, and we have already installed the TSM client rpms as described in “Client installation” on page 413, we can run the administration console by issuing this command:

```
lpar5:/opt/tivoli/tsm/client/admin/bin # ./dsmadmc
```

The default user name and password is: admin admin.

Now, we can set the administration address, as shown in Example 8-31 on page 411.

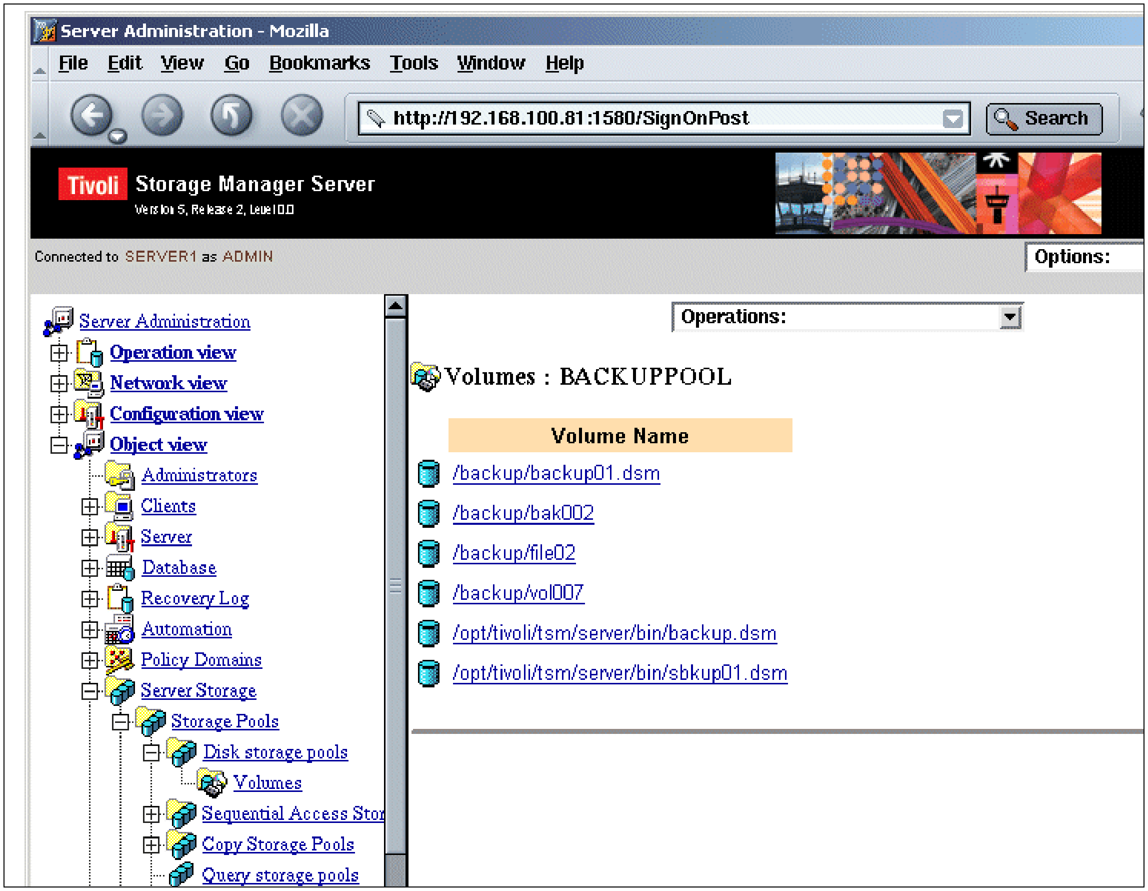
*Example 8-31 Setting the Web administration address*

```
TSM:SERVER1>
set serverlladdress 1500
ANR2017I Administrator SERVER_CONSOLE issued command: SET SERVERLLADDRESS 1500
ANR2133I Server lladdress set to 1500.
TSM:SERVER1>
set serverhladdress 192.168.100.81
ANR2017I Administrator SERVER_CONSOLE issued command: SET SERVERHLADDRESS
```

```
192.168.100.81
ANR2132I Server hladdress set to 192.168.100.81.
TSM:SERVER1>
```

An alternative to this command line administration is using the Web administration interface. It gives a good overall picture of the configuration options. You can also increase the database size and add storage volumes through this interface.

Start your browser in the local network, go to the server-address:1580, and login with the default user admin and password admin. The Web administration interface is shown in Figure 8-27.



Volumes : BACKUPPOOL

| Volume Name |
|---|
| /backup/backup01.dsm |
| /backup/bak002 |
| /backup/file02 |
| /backup/vol007 |
| /opt/tivoli/tsm/server/bin/backup.dsm |
| /opt/tivoli/tsm/server/bin/sbkup01.dsm |

*Figure 8-27 TSM Web-based administration tool*

We define a volume in the backuppool storage pool, as shown in Example 8-32.

*Example 8-32 Defining new backup volume on disk*

```
TSM:SERVER1>
define volume BACKUPPOOL /backup/file02 formatsize=4000
ANR2491I Volume Creation Process starting for /backup/file02, Process Id 14
```

We define one client, lpar3, with password lpar3 as shown in Example 8-33.

*Example 8-33 Defining the client*

```
TSM:SERVER1>
register node lpar3 lpar3
ANR2017I Administrator SERVER_CONSOLE issued command: REGISTER NODE lpar3 ?***?
ANR2060I Node LPAR3 registered in policy domain STANDARD.
ANR2099I Administrative userid LPAR3 defined for OWNER access to node LPAR3.
TSM:SERVER1>
```

## Client installation

The IBM Tivoli Storage Manager backup-archive client requires the following software in order to run:

- Linux kernel 2.4.x (for x=2 or higher)
- glibc 2.2.x (for x=2 or higher)
- libstdc++ 2.9.x (for x=6 or higher)
- X Windows System X11R6 (for end user GUI only)
- RPM 3.0.x or 4.0 (for x=0 or higher)

For ACL support you also need the acl-2.0.19-17 package.

> **Note:** X Windows System X11R6 is a requirement to install the client. If it is not installed and you do not plan to use the end-user GUI, you have to add the nodeps option of rpm to disable the check for requirements. Use one of the following Window Manager Systems: Gnome, KDE 2, Exceed LDAP, or LAMP.

Enter the command shown in Example 8-34 to install the backup-archive client (command-line and API), the administrative client (command-line), and the Web client.

*Example 8-34 TSM client installation*

```
lpar3:/mnt/TSMClient.V52/tsmcli/linuxppc # rpm -ivh TIVsm-API.ppc64.rpm
TIVsm-API                   ##################################################
Postinstall of the API
```

```
TSM Linux API installation complete.

Be sure to set up the configuration files!

lpar3:/mnt/TSMClient.V52/tsmcli/linuxppc # rpm -ivh TIVsm-BA.ppc64.rpm
TIVsm-BA                  ##################################################
Postinstall of the Backup Archive client

TSM Linux client installation complete.

Be sure to set up the system configuration file
before starting the client!
lpar3:/mnt/TSMClient.V52/tsmcli/linuxppc # rpm -ivh TIVsm-BA.E2ACL.ppc64.rpm
TIVsm-BA.E2ACL            ##################################################
Postinstall of the EXT2 ACL support for Backup Archive client

The EXT2 ACL support for ITSM Linux client installation complete.
```

If the node has not been set up to use TSM before, copy the /opt/tivoli/tsm/client/ba/bin/dsm.sys.smp to dsm.sys. It is assumed that the dsm.sys file is controlled by the system administrator. Edit this file to include the TSM server that you wish to connect to, as shown in Example 8-35.

*Example 8-35 TSM file*

```
SErvername  lpar5
   COMMmethod         TCPip
   TCPPort            1500
   TCPServeraddress   lpar5.residency.local
```

Copy /opt/tivoli/tsm/client/ba/bin/dsm.opt.smp to dsm.opt. Edit this file to use the ITSM server listed in the previous step.

In order to use client we need to set two variables:

```
# export DSM_CONFIG=/opt/tivoli/tsm/client/ba/bin/dsm.opt
# export DSM_DIR=/opt/tivoli/tsm/client/ba/bin
```

## Starting the client

There are at least three different clients available for pSeries Linux in the directory /opt/tivoli/tsm/client/ba/bin:

- **dsmc** is the command line client.
- **dsm** is an old-fashioned X Windows client.
- **dsmj** is a Java-based graphical client.

After logging in with the previously defined node name lpar3 and password lpar3, we see the screen shown in Figure 8-28.

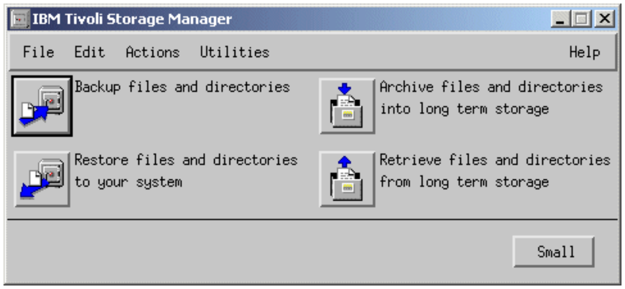
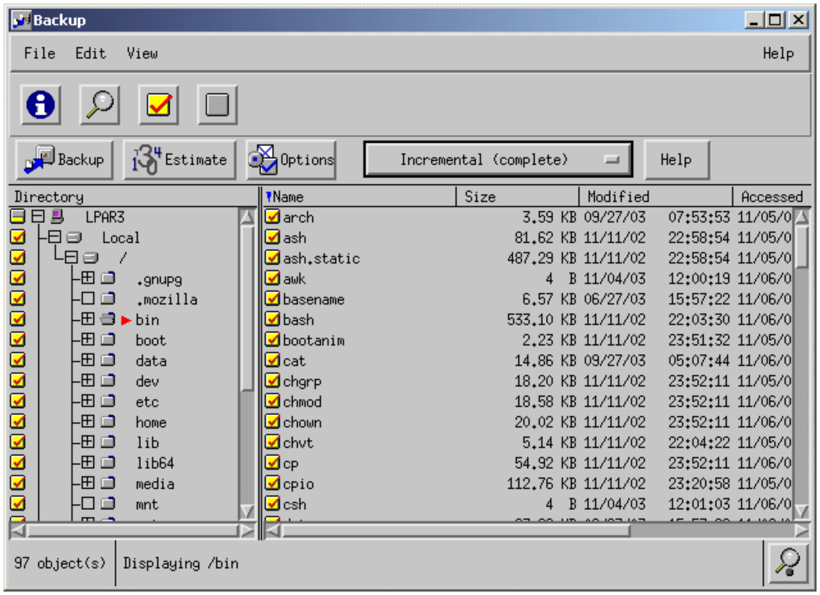

IBM Tivoli Storage Manager
File
Edit
Actions
Utilities
Help
Backup files and directories
Archive files and directories into long term storage
Restore files and directories to your system
Retrieve files and directories from long term storage
Small

*Figure 8-28 dsm screen*

We choose Backup files and directories and select Local to back up all local files, as shown in Figure 8-29.

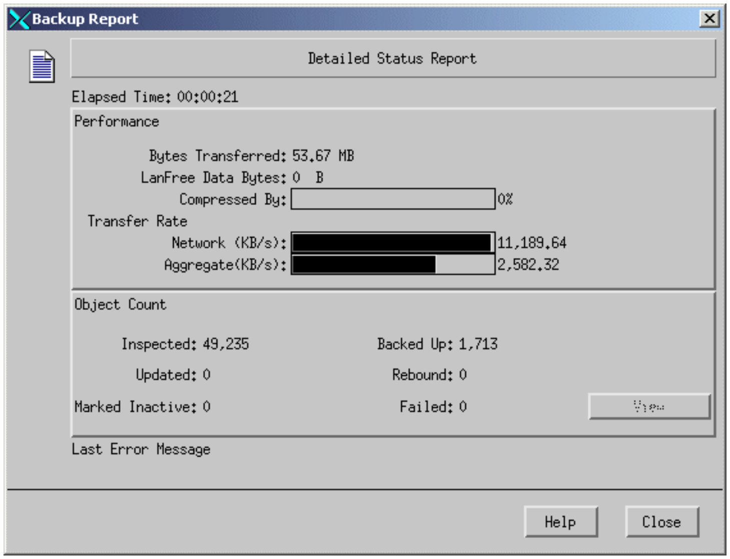


| Name | Size | Modified | Accessed |
|---|---|---|---|
| arch | 3.59 KB | 09/27/03 07:53:53 | 11/05/0 |
| ash | 81.62 KB | 11/11/02 22:58:54 | 11/05/0 |
| ash.static | 487.29 KB | 11/11/02 22:58:54 | 11/05/0 |
| awk | 4 B | 11/04/03 12:00:19 | 11/06/0 |
| basename | 6.57 KB | 06/27/03 15:57:22 | 11/06/0 |
| bash | 533.10 KB | 11/11/02 22:03:30 | 11/06/0 |
| bootanim | 2.23 KB | 11/11/02 23:51:32 | 11/05/0 |
| cat | 14.86 KB | 09/27/03 05:07:44 | 11/06/0 |
| chgrp | 18.20 KB | 11/11/02 23:52:11 | 11/05/0 |
| chmod | 18.58 KB | 11/11/02 23:52:11 | 11/06/0 |
| chown | 20.02 KB | 11/11/02 23:52:11 | 11/06/0 |
| chvt | 5.14 KB | 11/11/02 22:04:22 | 11/05/0 |
| cp | 54.92 KB | 11/11/02 23:52:11 | 11/06/0 |
| cpio | 112.76 KB | 11/11/02 23:20:58 | 11/05/0 |
| csh | 4 B | 11/04/03 12:01:03 | 11/06/0 |

97 object(s) Displaying /bin

*Figure 8-29 Selecting files and directories for backup*

Pressing the Backup button backs up lpar3 over the network to the disk device on lpar5, as shown in Figure 8-30 on page 416.



Elapsed Time: 00:00:21

Performance

Bytes Transferred: 53.67 MB
LanFree Data Bytes: 0  B
Compressed By: 0%
Transfer Rate
Network (KB/s): 11,189.64
Aggregate(KB/s): 2,582.32

Object Count

Inspected: 49,235 Backed Up: 1,713
Updated: 0 Rebound: 0
Marked Inactive: 0 Failed: 0

Last Error Message

*Figure 8-30 Backing up with dsm*

If we choose to use dsmj, it looks slightly different in design, but it has the same functionality and menu structure.

A

# Linux for AIX system administrators

In this appendix, we introduce Linux to AIX system administrators who co-manage AIX and Linux systems in their environments. This is a high level comparison of common administrative tasks and locations of system-specific files that are somewhat common to both Linux and AIX.

Note that this appendix does not provide an in-depth Linux comparison for AIX system administrators; its purpose is to help simplify the life of UNIX administrators managing servers in mixed environments. You can also use ppc utilities compiled by IBM for Linux to run some AIX commands on Linux.

In this appendix we compare:

- “Major features” on page 418
- “Common system files” on page 419
- “Task-specific command comparison” on page 420

## Major features

Table A-1 lists side-by-side comparisons of major features of AIX and Linux.

*Table A-1 Major features*

| Feature | AIX | Linux |
|---|---|---|
| Logical Volume Manager | LVM (can only increase file system size on the fly) | LVM (can increase and also decrease file systems on the fly) |
| JFS | JFS, JFS2 | JFS, ext2 and ext3 (Red Hat), Reiserfs (SuSE) |
| Admin interface | Smit, Smitty, WebSM | Webmin (Openware)<br>redhat-config (Red Hat)<br>YasT2 (SuSE) |
| Boot options | SMS menus (Press 1 or F1), Service Processor menus (Press 1 @beep) | SMS menus (Press 1 or F1), Open Firmware menus (Press 8 or F8) |
| Hardware monitoring | Inventory Scout, diagnostics, error logger | /var/adm/messages, Custom scripts |
| Clustering | HACMP | Cluster Suite (Red Hat)<br>Heartbeat (SuSE) |
| Recovering root access | boot from CD/network to maintenance mode | Boot from CD/network to “Rescue” mode |
| Network install | Network Install Manager (NIM) | Cluster Systems Management (CSM)<br>Kickstart (Red Hat)<br>AutoYaST (SuSE) |
| Image backups | mksysb,sysback | Storix (third party) |
| File system backups | smitty lvm, TSM, sysback | |
| Default shell | ksh | bash |
| Run levels | 0-1 (Reserved)<br>2 (Multiuser-default)<br>3-9 (User preferences)<br>S,s,m,M (Maintenance mode) | 0 (Halt)<br>1 (Single user mode)<br>2 (Multiuser-no NFS)<br>3 (Multiuser-with NFS-default)<br>4 Unused)<br>5 (default with X11, xdm)<br>6 (Reboot) |

| Feature | AIX | Linux |
|---|---|---|
| Default window manager | CDE | GNOME,KDE |

# Common system files

Table A-2 lists common system files and the locations of those files in AIX and Linux.

*Table A-2 Common system files*

| File | AIX | Linux |
|---|---|---|
| Password file | /etc/passwd | /etc/passwd |
| Encrypted password file | /etc/security/passwd | /etc/shadow |
| Error logs | /var/adm/ras/errpt<br>/var/adm/messages | /var/log/messages |
| Group files | /etc/group<br>/etc/security/group | /etc/group<br>/etc/gshadow |
| Allow/Deny/remote login | /etc/security/user | /etc/securetty |
| DNS | /etc/resolv.conf<br>/etc/netsvc.conf | /etc/resolv.conf<br>/etc/nsswitch.conf |
| Host Name definitions | /etc/hosts | /etc/hosts |
| Services | /etc/services | /etc/services |
| Kernel | /usr/lib/boot/unix_64 | /boot/vmlinuz |
| Device files | ODM database at /etc/objrepos | /dev |
| Inittab | /etc/inittab | /etc/inittab |
| inetd.conf | /etc/inetd.conf | /etc/inetd.conf |
| File systems | /etc/filesystems | /etc/fstab |
| Network definitions | ODM database at /etc/objrepos | /etc/sysconfig/network-scripts (Red Hat)<br>/etc/sysconfig/network (SuSE) |
| NFS exports | /etc/exports | /etc/exports |
| System environment | /etc/environment | /etc/profile<br>/etc/bash.bashrc |

| File | AIX | Linux |
|---|---|---|
| Common User-related | /etc/security/user | /etc/default/useradd |
| Profile scripts with new id | /etc/.profile | /etc/skel/*.* |

## Task-specific command comparison

Table A-3 compares task-specific commands for AIX and Linux.

*Table A-3 Command comparison*

| Task | AIX | Linux |
|---|---|---|
| Listing physical volumes | lspv | pvdisplay |
| List partitions in a disk | lspv -l <disk> | fdisk -l <disk> |
| List volume groups | lsvg | vgdisplay |
| Create volume group | mkvg | vgcreate |
| Remove volume group | rmvg | vgremove |
| Add physical volume to volume groups | extendvg | vgextend |
| Remove VG definition | exportvg | vgexport |
| Remove physical volume from volume group | reducevg | vgreduce |
| Import volume groups | importvg | vgimport |
| Activate volume group | varyonvg | vgchange |
| List logical volume | lslv | lvdisplay |
| Create logical volumes | mklv | lvcreate |
| Grow file systems with LV | chfs | resize_reiserfs, resize2fs |
| Shrink file system | - | resize_reiserfs, resize2fs |
| Paging/Swap space | lsps -a | procinfo<br>cat /proc/swaps |
| OS level | oslevel | uname -a |

| Task | AIX | Linux |
|---|---|---|
| Run level | who -r | runlevel |
| Uptime | uptime | uptime |
| Performance monitoring | vmstat, ps, sar | vmstat, ps, sar |
| List installed filesets | lslpp -l | rpm -qa |
| Which fileset a file is in | which_fileset <file_name> | rpm -qf <file_name> |
| Verify installed filesets | lppchk -v | rpm -V <packagename> |
| List files in a fileset/package | - | rpm -ql <package_name> |
| List running kernel modules | genkex | lsmod |
| Insert module | N/A (dynamic) | insmod, modprobe |
| Unload modules | N/A (dynamic) | rmmod, modprobe |
| List memory installed | bootinfo -r | free, procinfo, cat /proc/meminfo |
| Create users | mkuser | useradd |
| Change user details | chuser | usermod, chage |
| Delete users | rmuser | userdel |
| Create group | mkgrp | groupadd |
| Change group details | chgrp | groupmod |
| Delete group | rmgrp | groupdel |
| Install software | installp, smitty installp, rpm, geninstall | rpm -iv, yast -i, yast2 |
| Update software | smitty update_all, installp, rpm, | rpm -Uv, yast2 |
| Remove software | smitty installp, rpm | rpm -e, yast2 |
| IP configuration | smitty tcpip | config-network (Red Hat)<br>Yast2 network (SuSE) |
| IP alias | ifconfig en0 alias <IP> | ifconfig eth0:1<IP> |
| Network interfaces | netstat -ni | ifconfig |

| Task | AIX | Linux |
|---|---|---|
| Network routes | netstat -nr | netstat -nr |
| staticroutes | ODM, /etc/staticroutes&/etc/rc.net | /etc/sysconfig/network (Red Hat)<br>/etc/sysconfig/routes (SuSE) |
| Network options | no -a | sysctl -a |
| Error logs | errpt, alog | syslog, evlog, tail /var/log/messages, dmesg |
| Start daemons | startsrc -s (SRC-controlled subsystems) | rc.svc_name start, chkconfig, /etc/init.d/<svc_name> start |
| Stop daemons | stopsrc -s | rc.svc_name stop, chkconfig, /etc/init.d/svc_name stop |
| Refresh daemons | refresh -p | rc.svc_name restart, /etc/init.d/svc_name restart |
| Shutdown halt | shutdown -h | shutdown -h |
| Fast reboot | shutdown -Fr | shutdown -r now |
| System dump | sysdumpdev -l | N/A |
| Kernel tuning | vmo (virtual memory - was vmtune)<br>schedo (scheduler tuning - was schedtune)<br>no (network options) | sysctl (for all) |
| Change kernel | Change symlink in /usr/lib/boot/unix_mp | Change in /etc/yaboot.conf |
| PAM authentication | /etc/pam.conf | /etc/pam.d/* |
| Boot image | bosboot | ybin (Red Hat)<br>lilo (SuSE) |
| Change bootlist | bootlist | N/A |
| System boot messages | alog | dmesg |

B

# Feature comparison

In this appendix, we discuss major features of pSeries hardware while running SuSE Linux Enterprise server Version 8 and the IBM AIX Operating System. We present a general feature comparison of some of the major strengths of pSeries hardware currently available at the time of writing.

This changes upon further developments and announcements of Linux and AIX on pSeries hardware in the future.

Table B-1 shows a matrix of the major features, such as pSeries hardware options, storage options, operating system-related features and High Performance Computing options.

*Table B-1 Feature matrix*

| Feature | Running AIX | Running Linux |
|---|---|---|
| Logical Partitions (LPAR) support | Yes (AIX 5.X) | Yes |
| Dynamic Logical Partitioning (DLPAR) support | Yes (AIX 5.2) | No |
| Hot swappable disks | Yes | No |
| Hot swappable I/O adapters | Yes | No |
| Hot swappable memory cards | No | No |
| HMC for hardware control | Yes | Yes |

| Feature | Running AIX | Running Linux |
|---|---|---|
| System maintenance (SMS) options | Yes | Yes |
| Service Processor (SP) options | Yes | Yes (open boot firmware) |
| Hardware diagnostics | Yes | Yes |
| Firmware update | Yes | Yes |
| Other Reliability, Availability and Scalability (RAS) features | Yes | Yes |
| Serial Storage Architecture (SSA) | Yes | No |
| FastT storage | Yes | Yes |
| Shark storage support | Yes | Yes |
| External SCSI storage | Yes | Yes |
| SP Switch1/2 | Yes (using PSSP) | No |
| New HPS Switch | Yes (CSM) | No |
| Operating system dumps | Yes | No (could be hardware independent) |
| Workload Manager (WLM) | Yes (AIX-related) | No (OS-related) |
| LVM/JFS | Yes | Yes |
| LVM Raid support | Mirroring | Raid 0, 1, 4, 5 |
| File system types | JFS, JFS2 | ext2,ext3,JFS,ReiserFS, XFS etc. |
| Resizing logical volumes | Increasing only (mounted, on the fly) | Increasing (mounted, on the fly), decreasing unmounted |
| Cluster Systems Management (CSM) | Yes | Yes |
| GPFS support | Yes | Yes |
| Large page support | Yes | No |
| SMP support | Yes | Yes |
| Activec/VMX-HPC | No (planned) | Yes |

| Feature | Running AIX | Running Linux |
|---|---|---|
| DRM/DRMGR support | Yes | No |

**Note:** This feature matrix is mainly considered for pSeries hardware p615, p630, p650, p655, p690 and JS20.

C

# 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/SG247014
```

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, SG247014. The additional materials contain a README file, the ganglia binary RPMs and the source RPMs for SLES8 and RHAS3.

# Abbreviations and acronyms

| | |
|---|---|
| **ACL** | Access Control List |
| **ARP** | Address Resolution Protocol |
| **ATLAS** | Automatically Tuned Linear Algebra Software |
| **BIOS** | Basic Input Output System |
| **BIST** | Built In Self Test |
| **BLAS** | Basic Linear Algebra Subprograms |
| **CEC** | Central Electronic Complex |
| **CFM** | Cluster File Manager |
| **CHRP** | Common Hardware Reference Platform |
| **CSM** | Cluster Systems Management |
| **DCEM** | Distributed Command Execution Manager |
| **DHCP** | Dynamic Host Control Protocol |
| **DNS** | Domain Name Service |
| **dsh** | Distributed shell |
| **DVI** | Digital Video Interface |
| **ESSL** | Engineering and Scientific Subroutine Library |
| **EVLOG** | Enterprise Event log |
| **FTP** | File Transfer Protocol |
| **GCC** | GNU Compilers Collection |
| **GFLOPS** | Billion Floating Point Operations Per Second |
| **GPFS** | General Parallel File System |
| **HA** | High Availability |
| **HBA** | Host Based Authentication |
| **HMC** | Hardware Management Console |
| **HPC** | High Performance Computing |
| **HPL** | High Performance Linpack |
| **HTTP** | HyperText Transfer Protocol |
| **IBM** | International Business Machines Corporation |
| **IPL** | initial program load |
| **ITSO** | International Technical Support Organization |
| **JBOD** | Just a Bunch Of Disks |
| **JDK** | Java Development Kit |
| **LAM** | Local Area Multicomputer |
| **LAN** | local area network |
| **LAPACK** | Linear Algebra PACKage |
| **LELA** | Linux Error Log Analysis |
| **Linpack** | Linear Algebra PACKages |
| **LPAR** | Logical Partition |
| **LVM** | Logical Volume Manager |
| **LVS** | Linux Virtual Servers |
| **Mbps** | Mega bits per second |
| **MFLOPS** | Million Floating Point Operations Per Second |
| **MPI** | Message Passing Interface |
| **NAT** | Network Address Translation |
| **NetPIPE** | Network Protocol Independent Performance Evaluator |
| **NFS** | Network File System |
| **NNTP** | Network News Protocol |
| **NSD** | Network Shared Disk |
| **NVRAM** | Non-Volatile RAM |
| **ODBC** | Open DataBase Connectivity |
| **OMP** | OpenMP |

| | |
|---|---|
| **PCI** | Peripheric Component Interconnect |
| **PESSL** | Parallel Engineering and Scientific Subroutine Library |
| **PHP** | Preprocessor Hypertext Protocol |
| **POST** | Power-on Self Test |
| **ppc** | Power PC |
| **PReP** | PowerPC Reference Platform |
| **PSSP** | Parallel Systems Support Program |
| **PVFS** | Parallel Virtual File Systems |
| **PVM** | Parallel Virtual Machine |
| **RAID** | Redundant Array of Independent Disks |
| **RHAS** | RedHat Advanced Server |
| **RM** | Resource Manager |
| **RMC** | Resource Monitoring and Control |
| **RPM** | RedHat Package Manager |
| **RS6000** | IBM Risk System 6000 |
| **RSCT** | Reliable Scalable Cluster Technology |
| **RTAS** | Run Time Abstraction Service |
| **SAN** | Storage Area Network |
| **SCSI** | Small Computer System Interface |
| **SLES** | SuSE Linux Enterprise Server |
| **SMP** | Symmetrical Multiple Processor |
| **SMS** | Systems Management Services |
| **SMS** | Software Maintenance System |
| **SMTP** | Simple Mail Transfer Protocol |
| **SP2** | System Power Parallel |
| **SPOF** | Single Point of Failure |
| **SRC** | System Resource Controller |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **TCP** | Transmission Control Protocol |
| **TFTP** | Trivial File Transfer Protocol |
| **TLES** | TurboLinux Enterprise Server |
| **TSM** | Tivoli Storage Manager |
| **VACPP** | VisualAge C++ |
| **VFS** | Virtual File System |
| **VLAN** | Virtual Local Area Network |
| **VNC** | Virtual Network Computing |
| **VSIPL** | Vector Signal Image Processing Package Library |
| **WCOLL** | Working Collective |
| **XML** | Extensible Markup Language |
| **XML** | eXtended Markup Language |
| **YABOOT** | Yet Another BOOT loader |
| **YAST** | Yet Another Setup Tool |

# 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 435. Note that some of the documents referenced here may be available in softcopy only.

- *CMS Guide for the PSSP System Administrator*, SG24-6953
- *An Introduction to CSM 1.3 for AIX 5L,* SG24-6859
- *Effective System Management Using the IBM Hardware Management Console for pSeries,* SG24-7038
- *The Complete Partitioning Guide for IBM eServer pSeries Servers*, SG24-7039
- *Configuring p690 in an IBM eServer Cluster 1600*, REDP-0187
- *IBM eServer Certification Study Guide - p690 Technical Support*, SG24-7195

## Other publications

These publications are also relevant as further information sources:

- *CSM for Linux: Software Planning and Installation Guide*, SA22-7853
- *CSM for Linux: Administration Guide,* SA22-7873
- *CSM for Linux: Hardware Control Guide,* SA22-7856
- *RSCT for Linux: Guide and Reference,* SA22-7892
- *2104-DU3 Installation Guide,* GA33-3311
- *General Parallel File System for Clusters: Concepts, Planning, and Installation Guide,* GA22-7968
- *General Parallel File System for Clusters: Administration and Programming Reference*, SA22-7967
- *General Parallel File System for Clusters: Problem Determination Guide*, GA22-7969

- *General Parallel File System: Data Management API Guide*, GA22-7966
- *IBM WebSphere Edge Server User Guide*, GC09-4567

## Online resources

These Web sites and URLs are also relevant as further information sources:

- Cluster Systems Management download page

  `http://techsupport.services.ibm.com/server/cluster/fixes/csmfixhome.html`
- SuSE Fix page

  `http://support.suse.de/psdb`
- AutoUpdate software download

  `http://freshmeat.net/projects/autoupdate`
- openCIMOM freeware download package

  `http://www.ibm.com/servers/aix/products/aixos/linux/download.html`
- IBM PowerPC processors

  `http://www.ibm.com/chips/products/powerpc/`
- Main mailing lists for Linux on PowerPC

  `http://lists.linuxppc.org`
- Linux on pSeries portal

  `http://www.ibm.com/servers/eserver/pseries/linux/`
- IBM technical support

  `http://techsupport.services.ibm.com/server/Linux_on_pSeries`
  `https://techsupport.services.ibm.com/server/mdownload`
- IBM DB2

  `http://www.ibm.com/db2/linux/validate`
- IBM Tivoli

  `http://www.ibm.com/software/tivoli/resource-center/storage/code-srm.jsp`
- Preprocessor Hypertext Protocol (PHP)

  `http://www.php.net/`
- Apache

  `http://www.apache.org/`
- Ganglia Sourceforge

  `http://ganglia.sourceforge.net/`

- RRD Tool

  `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/`
- RPM

  `http://www.rpm.org/max-rpm/`
- The Linux documentation project

  `http://www.tldp.org/`
- Linux Standard Base

  `http://www.linuxbase.org/`
- File system Hierarchy Standard (FHS)

  `http://www.pathname.com/fhs/`
- Daniel Robbins article on file systems

  `http://www-106.ibm.com/developerworks/linux/library/l-fs.html`
- Steve Best article on journaling file systems

  `http://www.linux-mag.com/2002-10/jfs_01.html`
- Benchmarks on file system performance

  `http://oregonstate.edu/~kveton/fs/`
- United Linux technical white paper

  `http://www.unitedlinux.com/pdfs/whitepaper4.pd`
- Red Hat white paper on ext3 file system

  `http://www.redhat.com/support/wpapers/redhat/ext3/index.html`
- Reiserfs home page

  `http://www.namesys.com`
- Interview with file system developers

  `http://www.osnews.com/story.php?news_id=69`
- DeveloperWorks tutorial on RPM

  `http://www-106.ibm.com/developerworks/linux/library/l-rpm1`
- Red Hat Network Update site

  `http://www.redhat.com/software/rhn/update/`
- Rsync hompepage

  `http://samba.org/rsync`
- Amanda backup system home page

  `http://www.amanda.org`
- Storix home page

http://www.storix.com

- BIND9 Administrator manual

  http://www.bind9.net/Bv9ARM.html

- Netfilter / iptables project

  http://www.netfilter.org
- LDAP browser

  http://www.iit.edu/~gawojar/ldap/
- OpenLDAP project

  http://www.openldap.org
- Webmin project

  http://www.webmin.com
- Rdist information

  http://www.magnicomp.com/rdist/
- Rdist introduction

  http://www.benedikt-stockebrand.de/rdist-intro.html
- sudo project

  http://www.courtesan.com/sudo
- Software raid howto

  http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html

# How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

**ibm.com**/redbooks

# Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

# Index

## Symbols
/dev/md0 124
/etc/init.d/dhcpd start 146
/usr/lpp/mmfs/src 293

## Numerics
3151 15
32-bit 305
64-bit 305, 314

## A
ACL 221
AIX 273, 423
AM/MPI 317
Amanda 143
Anaconda 41
architecture 212
ash 97
ATLAS 323
authentication 220
autoyast2 6, 67, 74
  control file 68, 75

## B
backup 120, 143
bandwidth 306
bash 97
BIOS (Basic Input Output System) 6
BLACS 317
BLAS 317, 323
BLAST 312
Boot loader corruption 184
boot partition 31
BOOTP 59
BT 349
Building for RPM RHAS 3 195
Building ganglia for SLES 8 197
Built-In Self Test (BIST) 3, 164

## C
cache 317
CBLAS 317
Central Electronic Complex (CEC) 164
CFM 219
cfmupdatenode 257
CG 348
ch_gm 335
ch_p4 335
ch_shmem 335
chnode 240, 243
CHRP 6, 79–80
clustat 404
Cluster 1350 274
Cluster Systems Management (CSM) 217
Clustering 212
comm 335
command 146
  /etc/init.d/syslog 177
  /sbin/iptables 150–151
  /sbin/slog_fwd 180
  /sbin/SuSEconfig 156
  /usr/bin/sudo 109
  /usr/sbin/iptables 151
  arp 57
  BasicSanityCheck 375
  blockdev 115
  boot 84
  cat /proc/mdstat 130
  chkconfig 105
  chkconfig apache on 105
  chkconfig httpd on 105
  chkconfig --list apache 105
  csmconfig 221
  csmconfig -r 221
  db2sampl 371
  dev 82
  devalias 84
  diagela 172
  evlsend 182
  evlview 182
  fdisk 365
  fdisk /dev/sda 128
  fdisk -l 128
  fsck 132
  getadapters 241

heartbeat start 376, 389
hwinfo 88
ifconfig 66
init 1 105
iptables -L 151
ipvsadm -L 391
ldapadd 157
lilo 86, 365
linuxrc 61
logger 177
ls 81
lscfg 9, 172
lspci 88
lsrpdomain 287
lsrpnode 288–289
lssrc 288, 296
lvcreate 121, 131
lvcreate -L 3000M -n data1-lv datavg 116
lvextend 119
lvmdiskscan 115
lvreduce 119
mkraid 130
mkreiserfs 116
mkrpdomain 287
mkzimage_cmdline 75
mmconfig 295
mmcrcluster 294
mmcrfs 300
mmcrnsd 299
mmfscrfs 300
mmlscluster 294
mmlsconfig 295–296
mmlsdisk 300
mmlsnsd 299
mmshutdown 301
mmstartup 296–297, 302
mount 47
nvram 169
online_update 139–140
portmap 59
preprpnode 286
printenv 84
pvcreate /dev/sdb 115
pvmove 120
pwd 82
rcdhcpd 56
rcldap start 154
rdist 107–108
redhat-config-kickstart 77
resize_reiserfs 120
rhn_register 141
rmrpdomain 289
rpm -V package 137
rsync 107
snap 170
ssh-keygen 144
startrpdomain 287
stoprpnode 289
stty 21
sudo 106, 108
sudo -L 112
sudo -V 112
sum 170
tftp 58
tripwire 99
tune2fs 132
up2date 141
update_flash 170
uptime 166
useradd 105
vgcreate datavg /dev/sdb 116
vgcreate raidvg /dev/md0 131
vgextend datavg /dev/sda6 116
visudo 109
yast inst_source 139
yast2 26, 88
yast2 dns 99
yast2 -list 98
Common Hardware Reference Platform (CHRP) 80
condition 262
cross compilers 314
cross-compilers 314
CSM 145
csmconfig 233, 235
csmsetupyast 240, 243

D
daemon
dhcpd 145
evlogd 179
evlogrmtd 179
klogd 174
mmfsd 283
rtas_errd 173
syslogd 174
data replication 285

data=journal 133
data=ordered 133
data=writeback 133
D-card 306
Debian 10
definenode 239
device
  /dev/md0 123, 131
dgef 320
dges 320
DHCP 26, 145
diagela 172
direct-attached disks 280–281
directory
  ../ppc/update/SuSE-SLES/8/patches 140
  ../ppc/update/SuSE-SLES/8/rpm 140
  /boot 292
  /cfmroot 219
  /etc/init.d 59
  /etc/init.d/ 374
  /etc/init.d/rc3.d 104
  /etc/profile.d 69
  /etc/skel 106–108
  /etc/sysconfig 89, 156
  /install/sles 61
  /ppc/netboot/ 365
  /proc/device-tree 85
  /srv/var/www 205
  /tftpboot 57
  /usr/lib/heartbeat 375
  /usr/local/apache 380
  /usr/lpp/mmfs/bin 283, 293
  /usr/lpp/mms/src 291
  /usr/share/doc/howto 50
  /usr/src/mmfs/src/config 291
  /var/adm/backup/rpmdb 89
  /var/lib/ganglia/rrds 205
  /var/lib/rpm 137
  /var/www 205
DiskDruid 42
distributions 10
DLPAR 423
DNS 238
dsh 221, 250
dshbak 222
dual ported 306
Dumps 424
DVI 14
Dynamic LPAR 13
dynamic mapping 309

E
e32 274
e-business 215
E-card 306
Embarrassingly Parallel 312, 348
EP 348
ESSL 305, 320
ESSLSMP 320, 325
event 262
ext2 131–132
ext3 131–132

F
failure group 283
FASTA 312
FastT 424
fault tolerant 213
FFTW 317
file
  /boot/yaboot.chrp 31
  /dev/gpfs0 300
  /etc/bootptab 55
  /etc/dhcpd.conf 56, 145–146
  /etc/exports 59
  /etc/fstab 46
  /etc/gmetad.conf 207
  /etc/gmond.conf 207
  /etc/group 107
  /etc/gshadow 107
  /etc/ha.d/authkeys 373
  /etc/ha.d/ha.cf 373
  /etc/ha.d/haresources 373
  /etc/ha.d/ldirectord.cf 388
  /etc/inetd.conf 55
  /etc/init.d/boot 104
  /etc/init.d/boot.local 104
  /etc/init.d/skeleton 104
  /etc/inittab 104–105, 371
  /etc/lilo.conf 46, 86, 365
  /etc/login.defs 106–107
  /etc/named.conf 147
  /etc/nssswitch.conf 156
  /etc/openldap/ldap.conf 156
  /etc/openldap/slapd.conf 153
  /etc/passwd 107
  /etc/profile 97

/etc/resolv.conf 99
/etc/sasldb 155
/etc/security/pam_uni x2.conf 156
/etc/shadow 106–107
/etc/sudoers 112
/etc/syslog.conf 175
/etc/wgetrc 139
/etc/yaboot.conf 86–87
/linuxrc 51
/proc/ppc64/rtas/clock 167
/proc/ppc64/rtas/frequency 168
/proc/ppc64/rtas/poweron 168
/proc/ppc64/rtas/progress 165
/proc/ppc64/rtas/sensors 167
/proc/rtas/ 165
/proc/rtas/volume 168
/root/anaconda-ks.cfg 77
/tmp/linux-ha.testlog 375
/usr/local/apache/conf/httpd.conf 380
/var/log/messages 56, 147, 177
/var/named/residency.local 148
~/.bash_login 97
~/.bash_profile 97
~/.bashrc 97
~/.profile 97
100.168.192.rev 148
dhcpd.conf 147
id_rsa 144
id_rsa.pub 144
install 57
ks.cfg 76–77
netboot.img 57
nfsserver 59
opt/tivoli/tsm/server/bin/dsmserv.opt 410
residency.local 148
site.mcr 291
File system corruption 187
file systems 131
firmware 9, 224
First Failure Data Capture (FFDC) 3
FT 348

G
g++ 314
g77 314
gcc 314
General Parallel File System (GPFS) 4
getadapters 242
GFLOPS 321
Gigabit Ethernet 305, 312
GigE 312
GM 306, 324
GM mapper 308
GNU 314
GOTO 323
GPFS 279, 424
GPFS cluster 283, 294
GPFS kernel modules 298
GPFS nodeset 283, 295
GPL 97
Group Services 284, 287–288, 296
GXT135P 14

H
hardware control 219, 224
Hardware Management Console (HMC) 164
high availability 213, 306, 308
High Performance Computing (HPC) 4
High Performance Linpack 323
Highly Parallel Computing 323
HMC code 9
HPC 303, 323
HPL 323
HPL.dat 325
HPL_COLUMN_MAJOR 325
HPL_grid_init 325
HPL_ROW_MAJOR 325
hscroot 237
Hybrid 312
HyperTerminal 15

I
initrd 244
install 138
installms 230–231
installnode 245, 249, 257
IP forwarding 207
iptables 149
IS 348

J
JFS 131
journaling 132
journaling file system 132–133

**K**
kernel 225
kernel module
  mmfs 283
  mmfslinux 284
kernel modules 25
key 220
kickstart 6, 67, 76
  control file 76
ksh 97

**L**
L1 Data 317
L1 Instruction 317
L2 cache 317
LAM MPI 324
LAPAK 317
large page 321
latency 307
lc (loose clusters) 294
license 228, 233
lilo 262
Linear Algebra PACKage 323
Linpack 320, 322
Linpack DP 319, 323
Linpack TPP 322
Linux error log analysis (LELA) 164
load balancing 215
log
  /var/log/ha-log 376
  /var/log/messages 146
  /var/log/YaST2/y2log 99
  /var/mmfs/gen/mmfslog 297
logical drive 114
logical partition 127
logical volume 114, 116
Logical Volume Manager (LVM) 3, 24, 99, 113
logical volumes 113
LPAR 13, 223, 423
lshwinfo 238
lsnode 240, 250
lsnode -l 240, 245
lsrsrc 263–264
LU 349
lvcreate 230
lvdisplay 230
lvextend 230
LVM 32, 99, 113, 120, 123, 125
LVM volumes 126

**M**
makenode 246
man 237
man rpm 136
Managed 246
management domain 285
Mandelbrot 312
Manual mapping 309
message passing 312, 324
MFLOPS 320
MG 348
Minicom 15, 18
MinManage 246
mirrored device 125
mirroring 122
mkcondresp 265
mkresiserfs 230
monitorinstall 249
Monte Carlo 312
MOP/s 351
MPI 323
mpicc 325
MPICH 317, 324
MPICH_GM 335
mpif77 325
multi-threading 312
MX 306
Myrinet 306
Myrinet 2000 306

**N**
name server 147
netfilter 149
Network Shared Disks 299
NFS 58, 280
nodegrp 253
NPB 347
NSD 290
NSD (Network Shared Disk) 280

**O**
OBJECT_MODE 316, 325
Open Firmware 74, 78–80
Open Source 313
OpenMP 312

openSSH 220
openssh 144

P
p615 223
p630 223
p650 223
partition 113, 127
partition table 115
PATH 316
PCI 306
PCI-X 306
peak GFLOPS 321
peer domain 279, 285–286, 294
peer-to-peer 317
PESSL 305
PETSC 317
physical disks 113
physical extents 113, 120
physical partitions 113
physical volume 115
planning 222
portability layer 291
POWER4 3
Power4 305
Power4+ 305
power-on Reset (POR) 3
PowerPC 7, 79
ppc64 Linux run time abstraction service 165
PreManaged” 246
PReP 31, 80, 185
probemgr 265, 267
probes 220
PSSP 262, 274
pvcreate 230
PVFS 280

Q
-q32 316
-q64 316
-qarch 325
-qsmp 325
querying rpms 136

R
RAID 213
raid 123
raid level 124
raid partition 124
RAID0 123
RAID1 123
RAID4 123
raw -aq 396
rconsole 249
rconsole -r -t -n 245
read only volume 120
re-arm 262
Redbooks Web site 435
  Contact us xiv
redhat-config-cluster 397
reinstall 138
reiserfs 133
Reliability, Availability, and Serviceability (RAS) 2
Reliable Scalable Clustering Technology (RSCT) 218
resize_resiserfs 230
Resouce Manager
  IBM.AuditRm 219
Resource Manager
  IBM.DMSRM 219
  IBM.ERRM 219
  IBM.HWCtrlRM 219
response 262
RHAS 3 Rescue Mode 189
RM 212
RMC 212
rmnode 240
rpm 314
RPMS
  dhcp-server 56
  nfs-utils 59
RS232 227
RSCT 212
rtc 319
Run Time Abstraction Service 163
runtime function 319

S
SAN 280
sash 97
ScaLAPACK 317
script
  /etc/init.d/apache 105
  /linuxrc 51
second 319

Security 220, 227
sequence searching 312
Service Processor (SP) 3
shared memory 334
shared memory communication 334
Shark 424
Silicon-on-Insulator (SOI) 3
single point of failure (SPOF) 213
single ported 306
SLES8 Service Pack 3 290
SMP 305, 312, 320, 325, 334
SMS 220, 424
smsupdatenode 259, 262
snapshot 120
snapshot logical volume 120
software raid 123
SP 349, 424
SP2 274
SRC 284, 288
SSA 424
ssh key exchange 144
ssh-keygen 221
startcondresp 258
storage device 126
storix 143
sudo 109
SuperLU 317
swap device 127
Switch 424
Symmetrical Multiple Processor 305
sysctl 208
System Resource Controller 284
systemid 237
Systems Management Services (SMS) 25

## T
TCP 324, 331
tcsh 97
telnet 144
TFTP 59
threads 312
Topology Services 284, 287–288, 296
TPP 320
Trivial File Transfer Protocol (TFTP) 57

## U
UDP 324
unattended network installation 67
uninstallms 268
update 138
updatenode 250, 272
user roles 106

## V
VACPP 305, 315
VFS 279
vgcreate -v 230
vgdisplay 230
Virtual 212
Virtual Network Computing (VNC) 14
VNC 14–15, 26
volume group 116, 123
volume group (VG) 113
VSIPL 323

## W
watch 249
WCOLL 221
Webmin 100
WLM 424
WSMP 317

## X
xhpl 325
XLF 305, 315
xSeries 274

## Y
yaboot 79, 86
YaST 97

## Z
zone records 147
zsh 97

IBM

Redbooks

# Deploying Linux on IBM @server pSeries Clusters

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM®

# Deploying Linux on IBM *e*server pSeries Clusters

Redbooks

**System installation and administration**

**Application case studies**

**Cluster management**

For customers interested in Linux who are seeking powerful and reliable servers to support their applications, IBM eServer pSeries offers 64-bit POWER systems to match their requirements at lower costs. Unlike other servers supporting solely high performance Linux needs, IBM eServer pSeries provides leadership price/performance with high reliability and a strong roadmap to protect customers investments.

This IBM Redbook provides information and guidelines on how to deploy Linux on IBM eServer pSeries clusters. Topics covered include:
- System installation
- System administration
- Linux for pSeries RAS and problem determination
- Cluster Systems Management (CSM)
- Getting started with GPFS
- High Performance Computing case studies
- Commercial application case studies

This redbook also includes hints and tips that illustrate particular observations discovered during the residency. Additional reference materials and Web sites have been included as well, to provide the reader with other sources of information necessary to implement Linux on pSeries clusters.

**INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

**BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**
**ibm.com**/redbooks

SG24-7014-00 ISBN 0738498769