



SGI® Management Center for SGI® ICE™

007-5718-005

COPYRIGHT

© 2011, 2012, SGI. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of SGI.

The SGI Tempo systems management software stack, part of the SGI Management Center product, depends on several open source packages which require attribution. They are as follows:

c3:

C3 version 3.1.2: Cluster Command & Control Suite Oak Ridge National Laboratory, Oak Ridge, TN, Authors: M.Brim, R.Flanery, G.A.Geist, B.Luethke, S.L.Scott (C) 2001 All Rights Reserved NOTICE Permission to use, copy, modify, and distribute this software and # its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation. Neither the Oak Ridge National Laboratory nor the Authors make any # representations about the suitability of this software for any purpose. This software is provided "as is" without express or implied warranty. The C3 tools were funded by the U.S. Department of Energy.

conserver:

Copyright (c) 2000, conserver.com All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. - Neither the name of conserver.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1998, GNAC, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. - Neither the name of GNAC, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 1992 Purdue Research Foundation, West Lafayette, Indiana 47907. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or the Regents of the University of California. Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following

restrictions: 1. Neither the authors nor Purdue University are responsible for any consequences of the use of this software. 2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Credit to the authors and Purdue University must appear in documentation and sources. 3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. 4. This notice may not be removed or altered.

Copyright (c) 1990 The Ohio State University. All rights reserved. Redistribution and use in source and binary forms are permitted provided that: (1) source distributions retain this entire copyright notice and comment, and (2) distributions including binaries display the following acknowledgment: "This product includes software developed by The Ohio State University and its contributors" in the documentation or other materials provided with the distribution and in all advertising materials mentioning features or use of this software. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

pysqlite:

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

LIMITED RIGHTS LEGEND

The software described in this document is "commercial computer software" provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

TRADEMARKS AND ATTRIBUTIONS

Altix, ICE X, Performance Co-Pilot, SGI, the SGI logo, and Supportfolio are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries.

Altair is a registered trademark and PBS Professional is a trademark of Altair Engineering, Inc. Intel, Xeon, and Itanium are trademarks or registered trademarks of Intel Corporation. InfiniBand is a trademark of the InfiniBand Trade Association. Linux is a registered trademark of Linus Torvalds. LSI Logic is a registered trademark of the LSI Logic Corporation. InfiniScale is a registered trademark of Mellanox Technologies. Novell is a registered trademark and SUSE is a trademark of Novell, Inc., in the United States and other countries. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks mentioned herein are the property of their respective owners.

New Features in This Manual

This rewrite of the *SGI Management Center for SGI ICE* supports the SGI Management Center 1.5 release.

Major Documentation Changes

Performed the following:

- Added "Cluster Management Commands" on page 14.
- Added "cmcdetectd Daemon" on page 79.
- Updated "blademon Command For Automatic Blade Discovery" on page 89.
- Updated "MySQL Database Server Replication" on page 102.
- Added "Configure Maximum IRUs Supported by a RLC" on page 104.
- Added "Monitoring System Events Using the Embedded Support Partner" on page 319.

Record of Revision

Version	Description
001	February 2011 Original publication.
002	July 2011 Supports the SGI Management Center 1.4 release.
003	August 2011 Supports additional changes for the SGI Management Center 1.4 release.
004	November 2011 Supports changes for the SGI Management Center 1.5 release.
005	March 2012 Supports additional changes for the SGI Management Center 1.5 release.

Contents

About This Guide	xxxi
Related Publications	xxxi
Obtaining Publications	xxxii
Conventions	xxxiii
Reader Comments	xxxiii
1. SGI ICE System Overview	1
Hardware Overview	1
Basic System Building Blocks	1
InfiniBand Fabric	4
Gigabit Ethernet Network	5
Individual Rack Unit	5
Power Supply	6
Four-tier, Hierarchical Framework	6
Chassis Manager	8
System Nodes	9
System Admin Controller	10
Rack Leader Controller	10
Chassis Management Control (CMC) Blade	11
Compute Node	11
Individual Rack Unit	12
Login Service Node	12
Batch Service Node	13
Gateway Service Node	13

Storage Service Node	13
Cluster Management Commands	14
Software Management Overview	15
crepo Command	16
cinstallman Command	16
cimage Command	17
Networks	18
Networks Overview	18
Gigabit Ethernet (GigE) and 10/100 Ethernet Connections	20
VLANs	22
InfiniBand Fabric	26
Network Interface Naming Conventions	27
System Component Names	28
VLAN_Head Network Connections	28
VLAN_GBE Network Connections	29
VLAN_BMC Network Connections	30
VLAN_1588 Network Connections	31
Non-resolvable Names	31
Hostnames	32
InfiniBand Network	33
2. System Discovery, Installation, and Configuration	35
Configuring Factory-installed SGI ICE System	36
Accessing SGI Management Center	39
Install SMC During a Fresh Install	40
Install SMC on a System Running SMC for SGI ICE 1.5	40
Overview of Installing Software and Configuring Your SGI ICE System	41
Installing SMC for SGI ICE Admin Node Software	42

Installing SLES11 on the Admin Node	42
Installing RHEL 6 on the Admin Node	52
Initial Configuration of a RHEL 6 Admin Node	54
configure-cluster Command Cluster Configuration Tool	56
Set Network Time Configuration for RHEL6 Admin Node	78
cmcdetectd Daemon	79
discover Command	80
Installing Software on the Rack Leader Controllers and Service Nodes	86
blademon Command For Automatic Blade Discovery	89
Discovering Compute Nodes	91
Service Node Discovery, Installation, and Configuration	92
Configure the Backup DNS Server	94
InfiniBand Configuration	96
Redundant Management Network	98
Switch Management Network	100
MySQL Database Server Replication	102
Configure Maximum IRUs Supported by a RLC	104
Configuring the Service Node	105
Service Node Configuration for NAT	106
Troubleshooting Service Node Configuration for NAT	107
Using External DNS for Compute Node Name Resolution	108
Service Node Configuration for DNS	109
Service Node Configuration for NFS	109
Service Node Configuration for NIS for the House Network	109
Setting Up an NFS Home Server on a Service Node for Your SGI ICE System	111
Partitioning, Creating, and Mounting Filesystems	112
Home Directories on NAS	116

RHEL Service Node House Network Configuration	116
Setting Up a NIS Server for Your SGI ICE System	117
Setting Up a NIS Server Overview	118
Setting Up a SLES Service Node as a NIS Master	118
Setting Up a SLES Service Node as a NIS Client	120
Setting up a SLES Rack Leader Controller as a NIS Slave Server and Client	121
Setting up the SLES Compute Nodes to be NIS Clients	122
NAS Configuration for Multiple IB Interfaces	122
Creating User Accounts	125
Tasks You Should Perform After Changing a Rack Leader Controller	125
Installing SMC for SGI ICE Patches and Updating SGI ICE Systems	125
Overview of Installing SMC for SGI ICE Patches	126
Update the Local Package Repositories on the Admin Node	127
Mirroring Distrubtion Updates	127
Update the SGI Package Repositories on the Admin Node	127
SLES Admin Nodes: Update the SLES Package Repository	128
Register with Novell	128
Configuring the SMT Using YaST	129
Setting up SMT to Mirror Updates	130
Downloading the Updates from Novell and SGI	131
RHEL Admin Nodes: Update the RHEL Package Repository	131
Update Distros That Do Not Match the Admin Node	132
SLES	132
RHEL	132
Installing Updates on Running Admin, Leader, and Service Nodes	132
Bittorrent Provisioning	133
Updating Packages Within Systemimager Images	135

Additional Steps for Compute Image Kernel Updates	136
Cascading Dual-Boot	136
Partition Layout for Admin, Leader, and Service Nodes with Multiroot	137
Partition Layout for a Single Root	138
Admin Node Installation Choices Related to Cascading Dual-Boot	138
Leader and Service Node Installation	140
Choosing a Slot to Boot the Admin Node	140
How to Handle Resets, Power Cycles, and BMC dhcp Leases When Changing Slots	141
Leader and Service Node Booting	141
Leader and Service Node Booting on a System Configured with One Root Slot	142
Leader and Service Node Booting on a System Configured with Multiple Roots Slots	142
Slot Cloning	143
Admin Node: Managing Which Slot Boots by Default	143
Admin Node: Managing Grub Labels	143
Admin Node: Which root slot is in use?	144
Installing MPI on a Running SGI ICE System	144
System Admin Controller High Availability Solution	147
Overview	147
Hardware Requirements	148
Physical Connections	149
Software Requirements	149
Physical Node BMC Configuration	150
External Storage Configuration	150
Base SLES 11 SP1 Installation Considerations	150
Network Configuration	151
High Availability Extension Documentation	151
HAE, Virtualization and Updates installation	151

Install SGI Helper Scripts (<code>sac-ha</code> Package)	152
Configure the SGI Helper Scripts	152
Accessing <code>crm_gui</code> and the <code>hacluster</code> Account	154
Virtual Machine Installation	154
Allocate Physical Node IP Addresses with SGI Management Center	154
High Availability Customization	155
SMC Licenses	155
Virtual Admin Guest Service	155
Virtual Admin Node Resource	155
Special Virtual Machine Guest Configuration: SGI Emulator Wrapper Script	156
Migrating a Virtualized Admin Node Live	156
Rack Leader Controller High Availability Solution	158
Hardware Requirements	158
Software Requirements	158
SLES11 SP1 High Availability Extension (HAE) Documentation	159
Compute Node <code>tmpfs</code> Boot Requirement	159
Physical Connections	159
High Availability Extension	162
HA-RLC Naming	162
HA-RLC Networking Changes	163
HAE with HA-RLC	164
Installing the ICE Admin Node for a HA-RLC Cluster	165
Create SLES11 SP1 and HAE Repositories	165
Update Repositories Example	165
Initial Cluster Configuration with HA-RLC	168
Discover HA—RLC Racks	169
Discover HA-RLC Rack Overview	169

Discover Both HA-RLCs in a Rack	170
Monitor HA-RLC Installation	171
Confirm HAE After HA-RLC Boot	171
Monitor Compute Node Set up	172
Confirm Boot of Compute Nodes	173
Confirm Initial HA-RLC Setup	173
Sequential Discovery of HA-RLC	174
Monitor Rack HAE Status	175
crm_mon Command Line Utility	175
crm_gui Utility	176
Hawk Browser Interface	177
Administrative Resource Migration	179
crm - Resource Migration	179
crm_gui - Resource Migrate	179
Hawk Browser Interface — Resource Migration	182
Replace a HA-RLC	185
Remove an HA-RLC from a Cluster	185
Discover Replacement HA-RLC	186
3. System Operation	187
Software Image Management	187
Finding Which Distributions (distros) Are Supported	188
Operating Systems Supported per Node Type	189
System Admin Controller	189
Rack Leader Controller	189
Service Nodes	189
Compute Nodes	190
Compute Node Services Turned Off by Default	190

crepo Command	191
cinstallman Command	195
Customizing Software On Your SGI ICE System	198
Creating Compute Node Custom Images	198
Modify Compute Image Kernel Boot Options	201
Compute Node Per-Host Customization for Additional Network Interfaces	201
Customizing Software Images	203
cimage Command	206
Using cinstallman to Install Packages into Software Images	210
Using yum to Install Packages on Running Service or Leader Nodes	211
Creating Compute and Service Node Images Using the cinstallman Command	211
Installing a Service Node with a Non-default Image	213
Retrieving a Service Node Image from a Running Service Node	214
Using a Custom Repository for Site Packages	215
SGI ICE System Configuration Framework	216
Cluster Configuration Repository: Updates on Demand	219
cnodes Command	219
Power Management Commands	220
cpower Command	220
Operations on Nodes	222
IPMI-style Commands	222
IRU, Rack, and System Domains	223
Shutting Down and Booting	224
C3 Commands	227
pdsh and pdcp Utilities	231
cadmin: SMC for SGI ICE Administrative Interface	232
Console Management	238

Keeping System Time Synchronized	240
System Admin Controller NTP	240
Rack Leader Controller NTP	241
Managed Service, Compute, and Leader BMC Setup with NTP	241
Service Node NTP	241
Compute Node NTP	241
NTP Work Arounds	241
Changing the Size of /tmp on Compute Nodes	242
Enabling or Disabling the Compute Node iSCSI Swap Device	244
Changing the Size of Per-node Swap Space	245
Switching Compute Nodes to a tmpfs Root	247
Setting up Local Storage Space for Swap and Scratch Disk Space	248
Viewing the Compute Node Read-Write Quotas	252
RAID Utility	254
Restoring the grub Boot-loader on a Node	257
Backing up and Restoring the System Database	257
Enabling EDNS	259
Firmware Management	259
License Requirement	259
Terminology	260
Firmware Update High Level Example	261
Firmware Manager Command Line Interface (fwmgr)	261
Firmware Manager Daemon (fwmgrd)	262
Notes specific to Management Center 1.3	262
4. System Fabric Management	265
InfiniBand Fabric Management	265

InfiniBand Fabric Overview	265
InfiniBand Management Tool Graphical User Interface	266
Fabric Component <code>sgifmcli</code> Command	270
<code>sgifmcli</code> SGI Fabric Component Command	271
<code>sgifmdb</code> Fabric Management Database Command	273
InfiniBand Fabric Management Configuration and Operation Overview	274
Network Topology	276
Configuring the InfiniBand Fabric	276
InfiniBand Fabric Failover Mechanism	279
Configuring the InfiniBand Fat-tree Network Topology	282
Configuring the Lightweight Fabric	283
Verifying the InfiniBand Network	284
Useful Utilities and Diagnostics	285
<code>ibstat</code> and <code>ibstatus</code> Commands	285
<code>perfquery</code> Command	287
<code>ibnetdiscover</code> Command	288
<code>ibdiagnet</code> Command	289
5. System Maintenance, Monitoring, and Debugging	295
Maintenance Procedures	295
Temporarily Take a Node Offline for Maintenance	295
Permanently Replace a Failed Blade	296
Permanently Remove a Blade	297
Add a New Blade	298
Node Replacement Procedure for a Cold Spare Admin, Leader, and Service Nodes	298
Cold Spare Admin or Leader Node Availability	299
Shelf Spare Hardware Limitations	300
Tools Required	300

Identify the Failed Unit and Unplug all Cables	300
Transfer Disks from Existing Server to the Cold Spare	304
Migrating to a Cold Spare: Importing the Disk Volumes	304
Migrating to a Cold Spare: Booting for the First Time on the Migrated Node	306
Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode	310
Overview	310
Enable or Disable Auto Recovery Mode	310
IP Addresses Reserved for Auto Recovery Mode	310
DHCP Set Up for Auto Recovery Mode	311
Auto Recovery and the <code>discover</code> Command	311
How To Avoid Out of Memory Occurrences on SLES11 and PBS Pro	311
Inventory Verification Tool	314
System Monitoring Overview	315
System Monitoring Operation	317
Accessing the Ganglia System Monitor	317
Monitoring System Metrics	318
SEL/Hardware Event Monitoring	318
Node Availability Monitoring	319
Monitoring System Events Using the Embedded Support Partner	319
Overview	320
Embedded Support Partner (ESP)	320
System Group Manager (SGM)	320
ESP Support for SGI ICE Systems	321
Setting up Automatic Call Logging	322
Set up the Customer Profile	322
Sending Event Data to the ESP Database at SGI	324
Additional Useful ESP Utilities	325

ESP Graphical User Interface	331
Monitoring System Metrics with Performance Co-Pilot	334
Configuring Compute Blade Metrics	335
Monitoring SDR Metrics	337
Turning Off the <code>temperature.pmie</code> Feature	338
Adjusting <code>temperature.pmie</code> Values	338
Cluster Performance Monitor	339
Troubleshooting	340
dbdump Command	340
smc-info-gather Command	342
cminfo Command	343
kdump Utility	344
System Firmware	345
BIOS Version Interrogation	345
BMC Revision Interrogation	345
CMC Version Interrogation	346
InfiniBand Version Interrogation	346
Getting Firmware Information for All System Nodes	346
Appendix A. Out of Memory Adjustment	349
Index	367

Figures

Figure 1-1	Basic System Building Blocks for SGI ICE 8200	2
Figure 1-2	Basic System Building Blocks for SGI ICE 8400	3
Figure 1-3	Chassis Manager Cabling	9
Figure 1-4	Service Nodes	14
Figure 1-5	Network Connections In a System With Two IRUs	19
Figure 1-6	Chassis Manager	20
Figure 1-7	VLAN_GBE and VLAN_BMC Network Connections - IRU View	23
Figure 1-8	VLAN_GBE and VLAN_BMC Network Connections – Rack View	24
Figure 1-9	VLAN_HEAD Network Connections	25
Figure 1-10	Two InfiniBand Fabrics in a System with Two IRUs	27
Figure 2-1	System Admin Controller Power On Button and DVD Drive	43
Figure 2-2	YaST2 - firstboot@Linux Welcome Screen	45
Figure 2-3	Hostname and Domain Name Screen	46
Figure 2-4	Network Configuration II Screen	47
Figure 2-5	Network Settings Screen	48
Figure 2-6	Network Card Setup Screen	49
Figure 2-7	Network Settings Screen	50
Figure 2-8	Network Settings Routing Screen	51
Figure 2-9	Cluster Configuration Tool: Main Menu Screen	56
Figure 2-10	Cluster Configuration Tool: Initial Configuration Check Screen	58
Figure 2-11	Cluster Configuration Tool: Initial Cluster Setup Screen	59
Figure 2-12	Initial Cluster Setup Tasks Screen	60
Figure 2-13	Cluster Configuration Tool: Repo Manager Screen One	61

Figure 2-14	Cluster Configuration Tool: Repo Manager Screen Two	62
Figure 2-15	Cluster Configuration Tool: Repo Manager Screen Three	63
Figure 2-16	Cluster Configuration Tool: Repo Manager Screen Four	64
Figure 2-17	Cluster Configuration Tool: Repo Manager: Add Media Screen Four	65
Figure 2-18	Cluster Network Setup Screen	66
Figure 2-19	Update Subnet Address Warning Screen	67
Figure 2-20	Update Subnet Addresses Screen	68
Figure 2-21	Update Cluster Domain Name Screen	69
Figure 2-22	NTP Time Server/Client Setup Screen	70
Figure 2-23	Advanced NTP Configuration Screen	71
Figure 2-24	New Synchronization Screen	72
Figure 2-25	NTP Server Screen	73
Figure 2-26	NTP Time Server/ Client Setup Screen Three	73
Figure 2-27	Cluster Configuration Tool: NTP Time Server / Client Setup Screen (RHEL)	74
Figure 2-28	Admin Infrastructure One Time Setup Screen One	75
Figure 2-29	Configure House DNS Resolvers Screen	76
Figure 2-30	Setting DNS Forwarding Screen	77
Figure 2-31	Cluster Configuration Tool: Admin Infrastructure One Time Setup Screen	78
Figure 2-32	Configure blademond rescan interval Option Screen	90
Figure 2-33	Configure blademond Rescan Interval Screen	91
Figure 2-34	Configure Backup DNS Server Option Screen	94
Figure 2-35	Enter Service Node to Use for DNS Backup Popup Menu	95
Figure 2-36	Configure InfiniBand Fabric from Cluster Configuration Tool	96
Figure 2-37	InfiniBand Management Tool Screen	97
Figure 2-38	Administer InfiniBand GUI	98
Figure 2-39	Configure Redundant Management Network Option Screen	99
Figure 2-40	Redundant Management Network? Screen	99

Figure 2-41	Configure Switch Management Network Option Screen	101
Figure 2-42	Default Switch Management Network Setting Screen	101
Figure 2-43	Configure MySQL Replication (optional) Option	102
Figure 2-44	Configure MySQL Replication Screen	103
Figure 2-45	Configure Default Max Rack IRU Setting Option Screen	104
Figure 2-46	Configure Max Rack IRUs Screen	105
Figure 2-47	Admin Node High Availability Solution Functional Diagram	148
Figure 2-48	HAE Network Ethernet Connections	149
Figure 2-49	CMC Network Cabling Diagram for HA-RLC	161
Figure 2-50	CRM Cluster Status GUI	177
Figure 2-51	Hawk Interface with Resource Control Pop-up Menu	178
Figure 2-52	<code>crm_gui</code> Migrate alias-group Resource Example	180
Figure 2-53	<code>crm_gui</code> Migrate alias-group Resource Example	181
Figure 2-54	CRM GUI Configuration Constraints Selection Example	182
Figure 2-55	Using Hawk Browser Interface to Migrate Resources (Method 1 of 2)	183
Figure 2-56	Using Hawk Browser Interface to Migrate Resources (Method 2 of 2)	184
Figure 4-1	InfiniBand Management Tool Screen	267
Figure 4-2	Configure Topology Screen	268
Figure 4-3	Administer InfiniBand Tool Screen	269
Figure 4-4	Administer InfiniBand Status Option	269
Figure 4-5	Two InfiniBand Fabrics in a System with Two IRUs	275
Figure 4-6	<code>opensm</code> Software Failover	280
Figure 5-1	Admin/RLC Server Front Panel Controls and Indicator LEDs	301
Figure 5-2	Simple CMC LAN (VLAN) Cable Examples	303
Figure 5-3	Admin/Leader Server Front Features and Rear Connector Locations	304
Figure 5-4	Ganglia System Monitor	316
Figure 5-5	Ganglia System Monitoring Node View	317

Figure 5-6 pmice- Cluster Performance Monitor 339

Examples

Example 2-1	<code>discover</code> Command Examples	84
Example 2-2	Turning On the Redundant Management Network On Leader Node	100
Example 2-3	<code>tcpdump</code> Command Examples	107
Example 3-1	<code>cimage</code> Command Examples	207
Example 3-2	<code>cnodes</code> Example	220
Example 3-3	<code>cpower</code> Command Examples	226
Example 3-4	C3 Command General Examples	228
Example 3-5	C3 Command Specific Use Examples	231
Example 3-6	SMC for SGI ICE Administrative Interface (<code>cadmin</code>) Command	235
Example 3-7	Using the <code>lsiutil</code> Utility	254
Example 4-1	Getting <code>sgifmdb(8)</code> Command Help	274
Example 5-1	<code>dbdump</code> Command Examples	341
Example 5-2	<code>cminfo</code> Command Examples	344
Example A-1	<code>oom_adj.user.pl.txt</code> : OOM Adjustment Script	349
Example A-2	<code>cronentry</code> : Sample cron Entry for <code>oom_adj</code> Script	350
Example A-3	<code>prologue</code> : Sample prologue Script	350
Example A-4	<code>epilogue</code> : Sample epilogue Script	353
Example A-5	<code>chk_node.pl.txt</code> : Script epilogue and prologue Use.	357

Procedures

Procedure 2-1	Configuring Factory-installed SGI ICE System (SLES)	36
Procedure 2-2	Configuring Factory-installed SGI ICE System (RHEL)	38
Procedure 2-3	Install SGI Management Center	40
Procedure 2-4	Overview of Installing Software and Configuring Your SGI ICE System	41
Procedure 2-5	Installing SLES 11 Software on the System Admin Controller	43
Procedure 2-6	Installing RHEL 6 Software on the System Admin Controller	52
Procedure 2-7	Initial Configuration of a RHEL 6 Admin Node	54
Procedure 2-8	Using the Cluster Configuration Tool to Configure Your Admin Node	57
Procedure 2-9	Installing Software on the Rack Leader Controllers and Service Nodes	86
Procedure 2-10	Discovering Compute Nodes	92
Procedure 2-11	Service Node Configuration for NAT	106
Procedure 2-12	Service Node Configuration for NFS	109
Procedure 2-13	NIS with Compute Nodes Directly Accessing the House NIS Infrastructure	110
Procedure 2-14	NIS with a Service Node as a NIS Slave Server to the House NIS Master	110
Procedure 2-15	Partitioning and Creating Filesystems for an NFS Home Server on a Service Node	112
Procedure 2-16	Setting Up a SLES Service Node as a NIS master	118
Procedure 2-17	Setting Up a SLES Service Node as a NIS Client	120
Procedure 2-18	Setting up a Rack Leader Controller as a NIS Slave Server and Client	121
Procedure 2-19	Setting up the Compute Nodes to be NIS Clients	122
Procedure 2-20	Creating User Accounts on a NIS Server	125
Procedure 2-21	Configuring SMT Using YaST	129
Procedure 2-22	Setting up SMT to Mirror Updates	130

Procedure 2-23	Migrating a Virtualized Admin Node Live	157
Procedure 2-24	Create Software Repositories for HA-RLC	166
Procedure 2-25	Initial Cluster Configuration with HA-RLC	168
Procedure 3-1	Creating a Simple Compute Node Image Clone	203
Procedure 3-2	Manually Adding a Package to a Compute Node Image	204
Procedure 3-3	Manually Adding a Package to the Service Node Image	205
Procedure 3-4	Using the <code>cinstallman</code> Command to Create a Service Node Image:	212
Procedure 3-5	Use the <code>cinstallman</code> Command to Create a Compute Node Image	212
Procedure 3-6	Setting Up a Custom Repository for Site Packages	215
Procedure 3-7	Using <code>conserver</code> Console Manager	239
Procedure 3-8	Increasing the <code>/tmp</code> Size	242
Procedure 3-9	Enabling the iSCSI Swap Device	245
Procedure 3-10	Disabling the iSCSI Swap Device	245
Procedure 3-11	Increasing Per-node Swap Space	245
Procedure 3-12	Switching Compute Nodes to a <code>tmpfs</code> Root	247
Procedure 3-13	Viewing the Compute Node Read-Write Quotas	252
Procedure 3-14	Backing up and Restoring the System Database	258
Procedure 3-15	Enabling EDNS	259
Procedure 4-1	Configure the Master Subnet Manager	276
Procedure 4-2	Enabling the InfiniBand Failover Mechanism	280
Procedure 4-3	Configuring InfiniBand Fat-tree Network Topology	282
Procedure 4-4	Configuring the Lightweight Fabric	284
Procedure 4-5	Verifying the InfiniBand Network	284
Procedure 5-1	Temporarily Take a Node Offline for Maintenance	296
Procedure 5-2	Permanently Replace a Failed Blade	296
Procedure 5-3	Permanently Remove a Blade	297

Procedure 5-4	Add a New Blade	298
Procedure 5-5	Replacing a Node with a Cold Spare: Installing the Hardware	301
Procedure 5-6	Migrating to a Shelf Spare: Importing the Disk Volumes	305
Procedure 5-7	Migrating to a Cold Spare in a Non-cascading Dual Boot Cluster Node	307
Procedure 5-8	Migrating to a Cold Spare: Service or Leader Using Cascading Dual Boot	309
Procedure 5-9	Setting up the ESP GUI to Access ESP Data	332
Procedure 5-10	Turning Off the <code>temperature.pmie</code> Feature	338
Procedure 5-11	Adjusting <code>temperature.pmie</code> Values	338

About This Guide

This guide is a reference document for people who manage the operation of SGI® ICE™ systems. It describes how to use SGI® Management Center (SMC) software for SGI ICE systems to perform general system discovery, installation, configuration, and operations.

This manual contains the following chapters:

- Chapter 1, "SGI ICE System Overview" on page 1
- Chapter 2, "System Discovery, Installation, and Configuration" on page 35
- Chapter 3, "System Operation" on page 187
- Chapter 4, "System Fabric Management" on page 265
- Chapter 5, "System Maintenance, Monitoring, and Debugging" on page 295

Note: This manual documents SMC for SGI ICE 8200 and SGI ICE 8400 systems. For information about SMC for SGI® ICE™ X, see *SGI Management Center for SGI ICE X* at <http://docs.sgi.com>.

Related Publications

This section describes documentation you may find useful, as follows:

- *SGI Management Center Installation and Configuration*

This manual is intended for system administrators. It describes how to install and configure the SGI Management Center. A companion manual, *SGI Management Center System Administrator Guide*, describes general cluster administration.

- *SGI Management Center System Administrator's Guide*

This manual describes how you can monitor and control a cluster using the SGI Management Center. A companion manual, *SGI Management Center Installation and Configuration Guide*, describes installing and configuring the SGI Management Center.

- *SGI Altix ICE 8200 System Hardware User's Guide*

This is the hardware user's guide for the SGI ICE 8200 series systems. It describes the features of the SGI ICE 8200 series system, as well as, troubleshooting, upgrading, and repairing.

- *SGI Altix ICE 8400 Series System Hardware User's Guide*

This is the hardware user's guide for the SGI ICE 8400 series systems. It describes the features of the SGI ICE 8400 series system, as well as, troubleshooting, upgrading, and repairing.

- *SGI ICE X System Hardware User Guide*

This is the hardware user's guide for the SGI ICE X systems. It describes the features of the SGI ICE X system, as well as, troubleshooting, upgrading, and repairing.

- *SGI Management Center for SGI ICE X*

This manual describes how to use SGI Management Center (SMC) software for SGI ICE X systems to perform general system discovery, installation, configuration, and operations.

For a list of current SGI software and hardware manuals, see the *SGI Performance Suite 1.3 Start Here*:

- SGI documentation supporting SGI ICE systems
- Novell documentation for SUSE Linux Enterprise Server 11 (SLES11)
- Red Hat documentation for Red Hat Linux Enterprise Server 6 (RHEL 6)
- Intel Compiler Documentation
- Intel documentation about Xeon architecture

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at: <http://docs.sgi.com>. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.

- Online versions of the *SGI Performance Suite 1.3 Start Here*, release notes, which contain the latest information about software and documentation for each SGI Performance Suite product, the list of RPMs distributed with each product can be found in the `/docs` directory on each SGI Performance Suite product media.
- You can view man pages by typing `man title` on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
user input	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

techpubs@sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.
- Send mail to the following address:

SGI
Technical Publications
46600 Landing Parkway
Fremont, CA 94538

SGI values your comments and will respond to them promptly.

SGI ICE System Overview

The SGI Integrated Compute Environment (ICE) systems are an integrated blade environment that can scale to thousands of nodes. The SGI Management Center for SGI® ICE™ software enables you to provision, install, configure, and manage your system. This chapter provides an overview of the SGI ICE system and covers the following topics:

- "Hardware Overview" on page 1
- "Cluster Management Commands" on page 14
- "Software Management Overview" on page 15
- "Networks" on page 18
- "Network Interface Naming Conventions" on page 27

Hardware Overview

This section provides a brief overview of the SGI ICE system hardware and covers the following topics:

- "Basic System Building Blocks" on page 1
- "System Nodes" on page 9

For detailed hardware descriptions, see the *SGI Altix ICE 8200 Series System Hardware User's Guide* or the *SGI Altix ICE 8400 Series System Hardware User's Guide*.

Basic System Building Blocks

The SGI ICE system is a blade-based, scalable, high density compute system. The basic building block is the individual rack unit (IRU). The IRU provides power, cooling, system control, and the network fabric for 16 compute blades. Figure 1-1 on page 2 shows an SGI ICE 8200 series system. Four IRUs can reside in a custom designed 42U high rack.

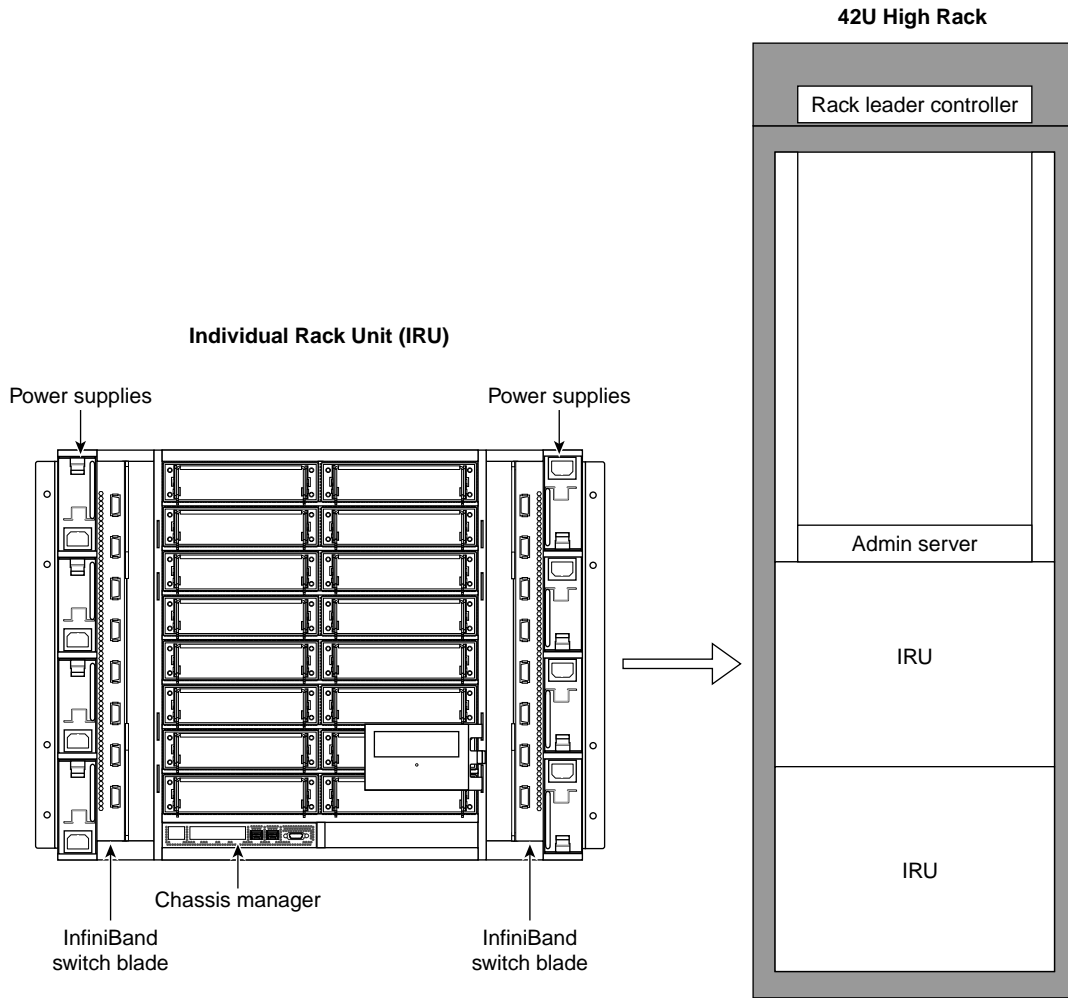


Figure 1-1 Basic System Building Blocks for SGI ICE 8200

The SGI ICE 8400 series of computer systems are also based on an InfiniBand I/O fabric and may be equipped with either of two different single-wide blade types and quad-data rate (QDR) InfiniBand switch blades as shown in Figure 1-2 on page 3.

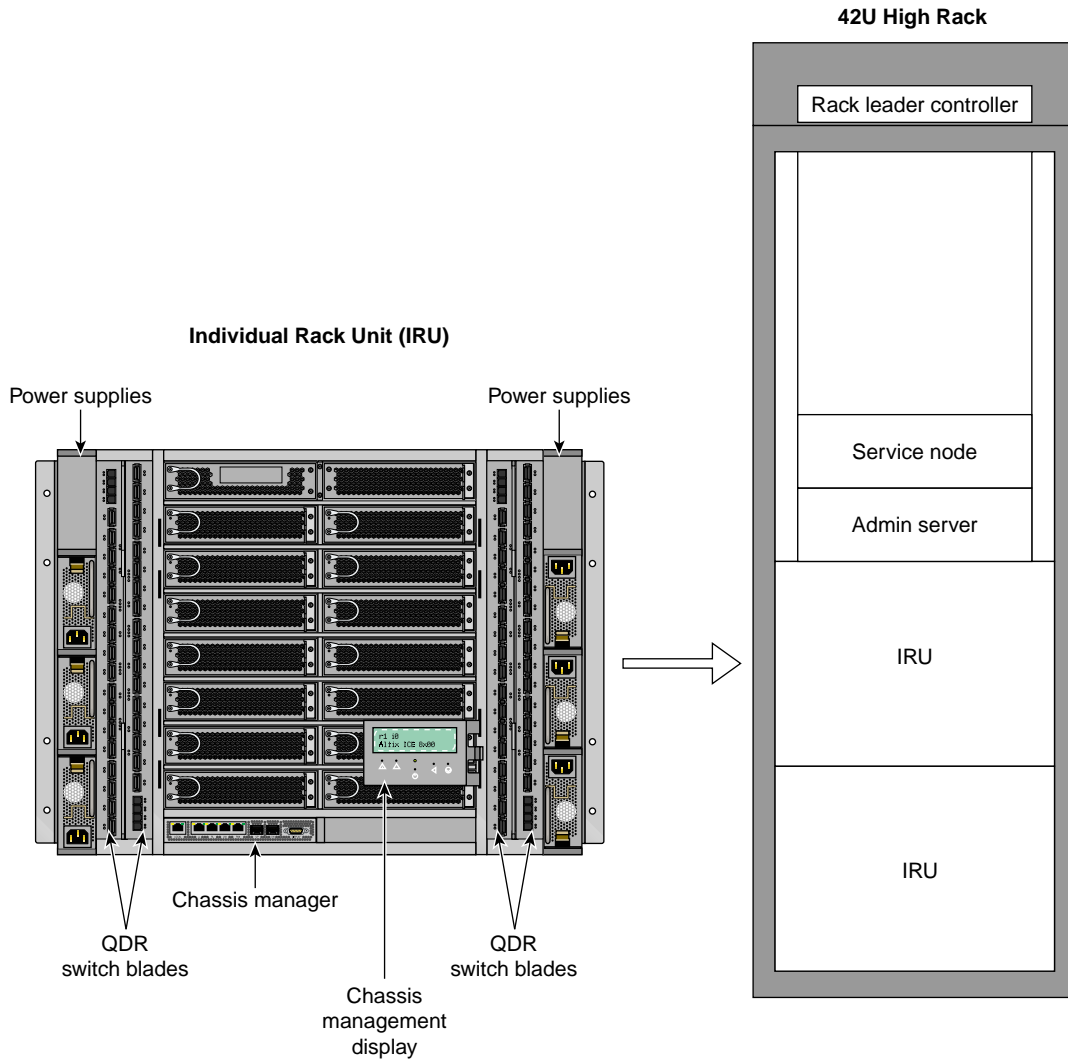


Figure 1-2 Basic System Building Blocks for SGI ICE 8400

For a detailed description of the SGI ICE 8400 series architecture, see the *SGI Altix ICE 8400 Series System Hardware User's Guide*.

This hardware overview section covers the following topics:

- "InfiniBand Fabric" on page 4
- "Gigabit Ethernet Network" on page 5
- "Individual Rack Unit" on page 5
- "Power Supply" on page 6
- "Four-tier, Hierarchical Framework" on page 6
- "Chassis Manager" on page 8

InfiniBand Fabric

The SGI ICE system uses an InfiniBand interconnect. Internal InfiniBand switch ASICs within the IRUs eliminate the need for external InfiniBand switches. InfiniBand backplanes built into the IRUs provide for fast communication between nodes and racks using relatively few InfiniBand cables.

The InfiniBand switch blade provides the interface between compute blades within the same chassis and also between compute blades in different IRUs. Fabric management software monitors and controls the InfiniBand fabric. SGI ICE systems are usually configured with two separate InfiniBand fabrics. These fabrics (also sometimes called InfiniBand subnets) are referred to as "ib0" and "ib1" within this document.

For information on MPI and MPT, see the *Message Passing Toolkit (MPT) User's Guide* available on the SGI Technical Publications Library at <http://docs.sgi.com>. The ib1 fabric is reserved for storage related traffic. The default configuration for MPI is to use only the ib0 fabric. For more information on the InfiniBand fabric, see Chapter 4, "System Fabric Management" on page 265.

Note: The "ib0 fabric" is a convenient shorthand for "the fabric which is connected to the ib0 interface on most of the nodes". Particularly in the case of storage service nodes, there may be several interfaces called ib0, ib1, and so on, all of which are connected to the same fabric (see "Storage Service Node" on page 13 and "NAS Configuration for Multiple IB Interfaces" on page 122). The LX series only has one ib fabric, "ib0". Any references to "ib1" in this manual do not apply to LX systems.

For performance reasons, it is often beneficial to use one fabric for message passing interface (MPI) traffic and the other for storage-related traffic. The default configuration for MPI is to use only the ib0 fabric and storage uses the ib1 fabric. For more information on the InfiniBand fabric, see Chapter 4, "System Fabric Management" on page 265.

Other configurations are possible, and may lead to better performance with specific workloads. For example, SGI's MPI library, the SGI Message Passing Toolkit (MPT), can be configured to use one, or both InfiniBand fabrics to optimize application performance. For information on MPI and MPT, see the *Message Passing Toolkit (MPT) User's Guide* available on the SGI Technical Publications Library at <http://docs.sgi.com>.

Gigabit Ethernet Network

A Gigabit Ethernet connection network built into the backplane of the IRUs provides a control network isolated from application data. Transverse cables provide connection between IRUs and between racks. For more information on how the Gigabit Ethernet connection fabric is used, see "VLANs" on page 22.

Individual Rack Unit

Each IRU has a one chassis management control (CMC) blade located directly below compute blade slot 0 as shown in Figure 1-1 on page 2. This is the chassis manager that performs environmental control and monitoring of the IRU. The CMC controls master power to the compute blades under direction of the rack leader controller (leader node). The leader node can also query the CMC for monitored environmental data (temperatures, fan speeds, and so on) for the IRU.

Power control for each blade is handled by its Baseboard Management Controller (BMC), also under direction of the rack leader controller. Once the leader node has asked the CMC to enable master power, the leader node can then command each BMC to power up its associated blade. The leader node can also query each BMC to obtain some environmental and error log information about each blade.

Note: Setting the circuit breakers on the power distribution units (PDUs) to the "On" position will apply power to the IRU and will start the chassis manager in each IRU. Note that the chassis manager in each IRU stays powered on as long as there is power coming into the unit. Turn off the PDU breaker switch that supplies voltage to the IRU if you want to remove all power from the unit. For detailed information about powering your system on or off, see the "Powering the System On and Off" section in chapter 1 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

The IRU provides data collected from compute nodes within the IRU to the leader node upon request.

Power Supply

The CMC and BMCs are powered by what is called "AUX POWER". This power supply is live any time the rack is plugged in and the main breakers are on. The CMC and BMCs are **not** able to be powered off under software control.

The compute blades have MAIN POWER which is controlled by the blade BMC. You can send a command to the BMC and have the main power to the associated blade turned on or off by that BMC.

The IRU has a MAIN POWER bus that feeds all of the blades. This main power bus can be turned on and off with a software command to the CMC. This "powering up of the IRU" turns on this main power, the fans in the IRU, and the power to the IB switches. The CMC, itself, is always powered on. This includes the Ethernet switch that is a part of the CMC.

Note: Setting the circuit breakers on the power distribution units (PDUs) to the "On" position will apply power to the IRU and will start the chassis manager in each IRU. Note that the chassis manager in each IRU stays powered on as long as there is power coming into the unit. Turn off the PDU breaker switch that supplies voltage to the IRU if you want to remove all power from the unit. For detailed information about powering your system on or off, see the "Powering the System On and Off" section in chapter 1 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

Four-tier, Hierarchical Framework

The SGI ICE system has a unique four-tier, hierarchical management framework as follows:

- System admin controller (admin node) – one per system
- Rack leader controller (leader node) – one per rack
- Chassis management controller (CMC) – one per IRU
- Baseboard Management Controller (BMC) – one per compute node, admin node, leader node, and managed service node

Unlike traditional, flat clusters, the SGI ICE system does **not** have a head node. The head node is replaced by a hierarchy of nodes that enables system resources to scale as you add processors. This hierarchy is, as follows:

- System admin controller (admin node)

- Rack leader controller (leader node)
- Service Nodes
 - Login
 - Batch
 - Gateway
 - Storage

The one system admin controller can provision and control multiple leader nodes in the cluster. It receives aggregated cluster management data from the rack leader controllers (leader nodes).

Each system rack has its own leader node. The leader node holds the boot images for the compute blades and aggregates cluster management data for the rack. The leader nodes are responsible for the compute nodes. They act as proxies for control and monitoring software (c3, conserver, pdsh, and so on). The leader nodes monitor the compute nodes' BMCs helping to keep monitoring loads lighter.

Diskless compute nodes in one rack do not compete for system bandwidth with compute nodes in other racks. Broadcast traffic is isolated. The system admin node does not get saturated with compute node traffic. The system admin node is only directly responsible for leader nodes and service nodes.

Ethernet traffic for managing the nodes in a rack is constrained within the rack by the leader node. Communication and control is distributed across the entire cluster, thereby preventing the admin node from becoming a communication bottleneck. Administrative tasks, such as booting the cluster, can be done in parallel rack-by-rack in a matter of seconds. For very large configurations, the access infrastructure can also be scaled by adding additional login and batch service nodes. It is the VLAN logical networks that help prevent network traffic bottlenecks.

Note: Understanding the VLAN logical networks is critical to administering an SGI ICE system. For more detailed information, see "VLANs" on page 22 and "Network Interface Naming Conventions" on page 27.

The rack leader controller (leader node) and system admin controller (admin node) are described in the section that follows ("System Nodes" on page 9).

Chassis Manager

Figure 1-3 on page 9 shows chassis manager cabling.

Note: All nodes reside in the SGI ICE custom designed rack. Figure 1-3 on page 9 and Figure 1-4 on page 14 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the SGI ICE hierarchical design. They are not meant as cabling diagrams.

The chassis manager in each rack connects to the leader node in its own rack and also the chassis manager in the adjacent rack. The system admin controller (admin node) connects to one CMC in the rack. The rack leader controller (leader node) accesses the BMC on each compute node in the rack via VLAN running over a Gigabit Ethernet (GigE) connection (see Figure 1-8 on page 24).

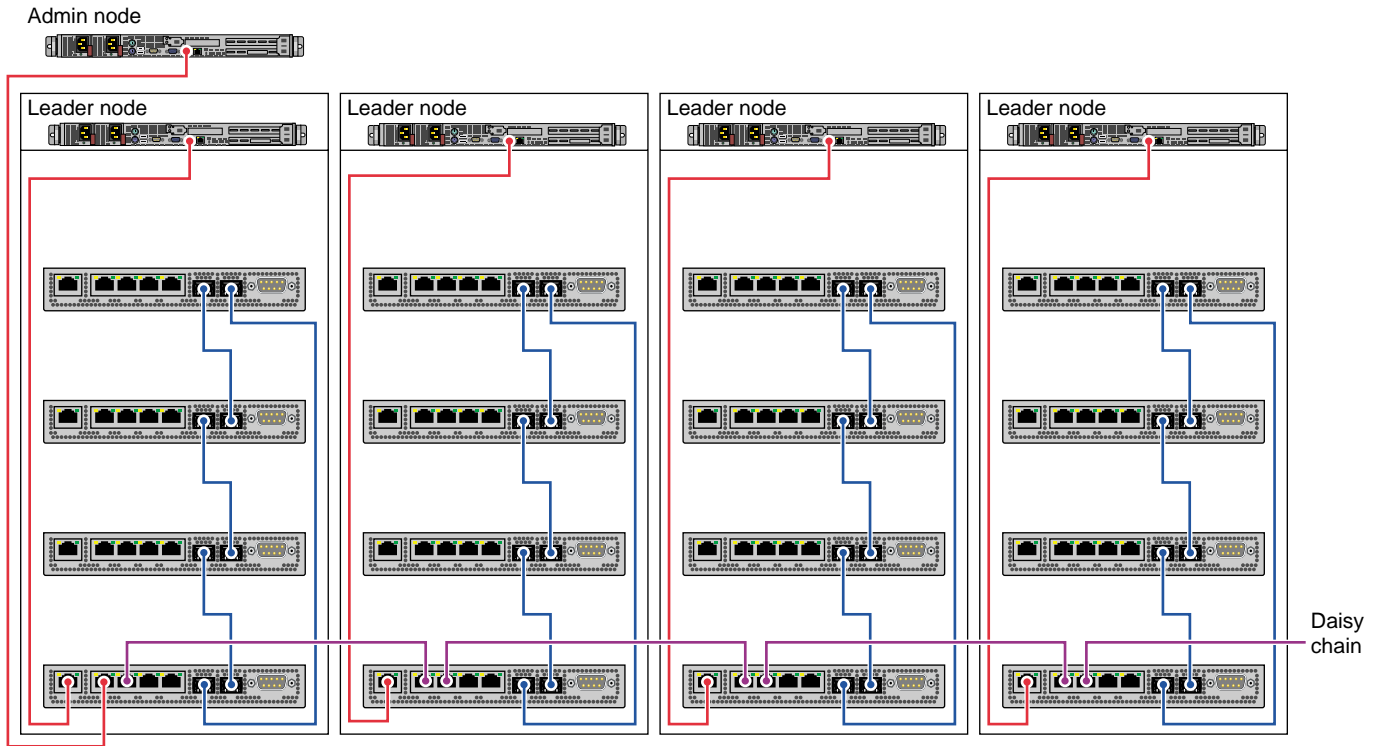


Figure 1-3 Chassis Manager Cabling

Figure 1-4 on page 14 shows cabling for a service node and storage service node (NAS cube).

System Nodes

This section describes the system nodes that are part of SGI ICE system and covers the following topics:

- "System Admin Controller" on page 10
- "Rack Leader Controller" on page 10
- "Chassis Management Control (CMC) Blade" on page 11

- "Compute Node" on page 11
- "Individual Rack Unit" on page 12
- "Login Service Node" on page 12
- "Batch Service Node" on page 13
- "Gateway Service Node " on page 13
- "Storage Service Node " on page 13

System Admin Controller

The system admin controller (admin node), is used by a system administrator to provision (install) and manage the SGI ICE system using SGI Management Center (SMC) software. There is only one system admin controller per SGI ICE system, as shown in Figure 1-3 on page 9 and it cannot be combined with any other nodes. A GigE connection provides the network connection between the admin node, leader nodes, and service nodes. Communication to and from the CMC and compute blades from the leader nodes is controlled by VLANs to reduce network traffic bottlenecks in the system. The system admin controller is used to provision and manage the leader nodes, compute nodes and service nodes. It receives and holds aggregated management data from the leaders node. The admin node is an appliance node. It always runs software specified by SGI.

Rack Leader Controller

The rack leader controller (leader node) is used to manage the nodes in a single rack. The rack leader controller is provisioned and functioned by the system admin controller (admin node). There is one leader node per rack, as shown in Figure 1-3 on page 9. A GigE connection provides the network connection to other leader nodes and to first IRU within its rack as shown in Figure 1-4 on page 14 and Figure 1-5 on page 19. An InfiniBand fabric connects it to the compute nodes within its rack and compute nodes in other racks. The leader node is an appliance node. It always runs software specified by SGI. The rack leader controller (leader node) does the following:

- Runs software to manage the InfiniBand fabric in your SGI ICE system
- Monitors and processes data from the IRUs within its rack
- Monitors and processes data from compute nodes within its rack

- Consolidates and forwards data from the IRUs and compute nodes within its rack to the admin node upon request

The leader node can contain multiple images for the compute nodes. "Customizing Software On Your SGI ICE System" on page 198 describes how you can clone and customize compute node images.

Chassis Management Control (CMC) Blade

Note: The following CMC description is the same as the information presented in "Basic System Building Blocks" on page 1.

Each IRU has one chassis management control (CMC) blade located directly below compute blade slot 0 as shown in Figure 1-1 on page 2. This is the chassis manager that performs environmental control and monitoring of the IRU. The CMC controls master power to the compute blades under direction of the rack leader controller (leader node).

Note: Setting the circuit breakers on the power distribution units (PDUs) to the "On" position will apply power to the IRU and will start the chassis manager in each IRU. Note that the chassis manager in each IRU stays powered on as long as there is power coming into the unit. Turn off the PDU breaker switch that supplies voltage to the IRU if you want to remove all power from the unit. For detailed information about powering your system on or off, see the "Powering the System On and Off" section in chapter 1 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

The leader node can also query the CMC for monitored environmental data (temperatures, fan speeds, and so on) for the IRU. Power control for each blade is handled by the Baseboard Management Controller (BMC) also under direction of the rack leader controller. Once the leader node has asked the CMC to enable master power, the leader node can then command each BMC to power up its associated blade. The leader node can also query each BMC to obtain some environmental and error log information about each blade.

Compute Node

Figure 1-1 on page 2 shows an IRU with 16 compute nodes. Users submit MPI jobs to run in parallel on the SGI ICE system compute nodes using a public network connection via the service node. The service node provides login services and a batch

scheduling service, such as PBS Professional (PBSPro 9.x), as shown in Figure 1-5 on page 19. The compute nodes are controlled and monitored by the leader node for their rack as shown in Figure 1-3 on page 9. Compute nodes are booted and mount the shared, read-only portion of the root file system from the rack leader controller (leader node). The leader node provides the network connections to the compute nodes in the same rack and to leader nodes in other rack that then provide the network connections to the compute nodes in their racks. These network connections are via the InfiniBand fabric. The system admin controller does not communicate directly with the CMC or compute blades. Actions for the CMC and compute blades are sent to the appropriate leader node, which communicates to the appropriate CMC and compute blades. The compute nodes do not communicate directly to the CMC or admin nodes, or leader nodes outside their rack.

Generally, the CMC controller is not meant to be accessed directly by system administrators, however, in some situations you may need to access it to change a configuration using the CMC interface LCD panel. For example, in a single IRU system, you may need more Ethernet ports for service node or NAS cube connections. You can adjust the CMC to use the **R58** jack or the **L58** jack for this purpose (see Figure 1-6 on page 20). For more information on these jacks, see "Gigabit Ethernet (GigE) and 10/100 Ethernet Connections" on page 20.

For information on the CMC interface LCD panel, see chapter 1 and chapter 6 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

For more information about configuring compute nodes, see the following:

- "Changing the Size of /tmp on Compute Nodes" on page 242
- "Enabling or Disabling the Compute Node iSCSI Swap Device" on page 244
- "Changing the Size of Per-node Swap Space" on page 245

Individual Rack Unit

The individual rack unit (IRU) is one of the basic building blocks of the SGI ICE system as shown in Figure 1-1 on page 2. It is described in detail in "Basic System Building Blocks" on page 1.

Login Service Node

The login service node allows users to login into the system to create, compile, and run applications. The login node is usually combined with batch and gateway service nodes for most configurations. The login service node is connected to the SGI ICE

system via the InfiniBand fabric and GigE to the public customer network as shown in Figure 1-5 on page 19. Additional login service nodes can be added as the total number of user logins grow.

Batch Service Node

The batch service node provides a batch scheduling service, such as PBS Professional. It is commonly combined with login and gateway service nodes for most configurations. It is connected to the SGI ICE system via the InfiniBand fabric and GigE to the public customer network. This node may be separated from gateway and/or login nodes to scale for large configurations or to run multiple batch schedules.

Gateway Service Node

The gateway service node is the gateway from the InfiniBand fabric to services on the public network such as storage, lightweight directory access protocol (LDAP) services, and file transfer protocol (FTP). Typically, it is combined with the login/batch service node. This node may be separated from login and/or batch nodes to scale for large configurations.

Storage Service Node

The storage service node is a network-attached storage (NAS) appliance bundle that provides InfiniBand attached storage for the SGI ICE system. There can be multiple storage service nodes for larger SGI ICE system configurations. Figure 1-4 on page 14 shows a service node and a storage service node (NAS cube).

For smaller SGI ICE systems, with less than one full rack of nodes (64 or less nodes), Network Attached Storage (NAS) is provided off an SGI XE250 system. It can also serve as a login or other support node using NFS. The XE250 is connected to the ICE system using InfiniBand and requires that Internet Protocol over InfiniBand (IPoIB) be properly configured on the system to allow the SGI ICE nodes to be attach to the XE250 provided storage.

Note: All nodes reside in the SGI ICE custom designed rack. Figure 1-3 on page 9 and Figure 1-4 on page 14 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the SGI ICE hierarchical design. They are not meant as cabling diagrams.

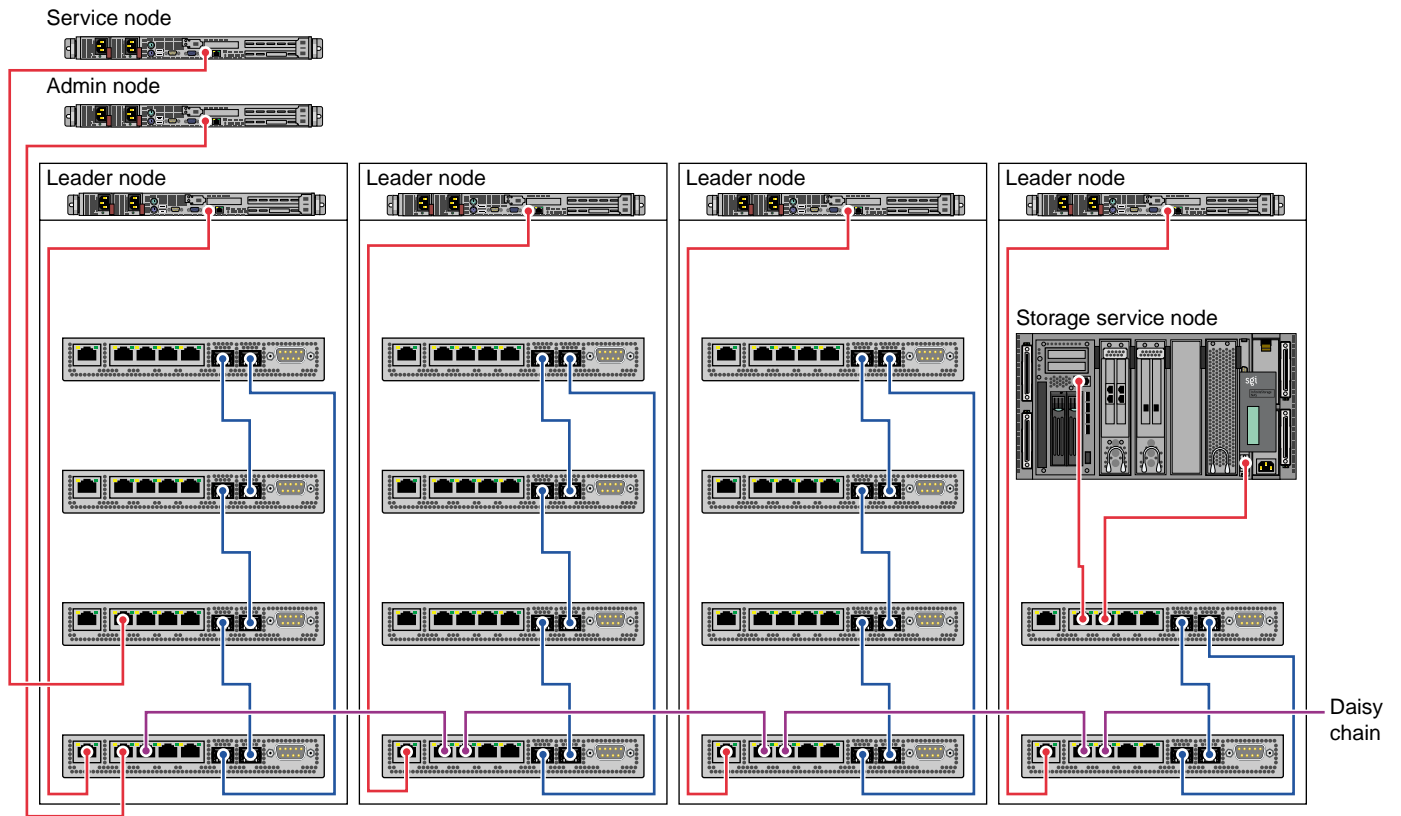


Figure 1-4 Service Nodes

Cluster Management Commands

The SGI ICE cluster management commands are located in the `/opt/sgi/sbin` directory of the system admin controller, login, and leader nodes, as follows:

- `configure-cluster`
Performs initial software, network, and NTP configuration of a new cluster. For more information, see "configure-cluster Command Cluster Configuration Tool" on page 56.
- `discover`

Discovers and provisions rack leader controllers (RLCs) and service nodes. For more information, see "discover Command" on page 80.

- cpower

Powers on or off the system, racks, service or compute nodes, and individual system comments and check their status. For more information, see "cpower Command" on page 220.

- console

Establishes a SOL (serial console) connection to the specified node. For more information, see "Console Management" on page 238.

- crepo

Maintains a software image repository in the /tftpboot directory. For more information, see "crepo Command" on page 16 and "crepo Command" on page 191.

- cinstallman

Builds custom images and maintains and updates software images on nodes. For more information, see "cinstallman Command" on page 16 and "cinstallman Command" on page 195.

- cimage

Creates and pushes software images to rack leader controllers from which the compute nodes use for booting. For more information, see "cimage Command" on page 17 and "cimage Command" on page 206.

- cadmin

Modifies cluster attributes, such as, network specifics. For more information, see "cadmin: SMC for SGI ICE Administrative Interface" on page 232.

Software Management Overview

This section provides an overview of the commands used to manage software running on rack leader controllers (leader nodes), compute nodes, and services nodes, as follows:

- "crepo Command" on page 16

- "cinstallman Command" on page 16
- "cimage Command" on page 17

Compute node images are staged on the leader nodes and pushed out to compute nodes using the `cimage` command. Leader and service nodes do not require staging. Leader and service nodes are installed from the admin node.

crepo Command

If you are installing a system from scratch, the `crepo` command is one of the first commands you will run after the system admin controller has been installed (see "Installing SMC for SGI ICE Admin Node Software" on page 42). You can use the `crepo` command to manage software repositories, such as, SGI Foundation, SMC for SGI ICE, SGI Performance Suite, and the Linux distribution(s) you are using on your system. You also use the `crepo` command to manage any custom repositories you create yourself. The `crepo` command stores RPMs from SGI media, distribution media, and is used to create images to update admin, leader, and service nodes.

Each repository has associated with it a name, directory, update URL, selection status, and suggested package lists. The update URL is used by the `sync-repo-updates` command.

You can use the `crepo --show` command to view the location of the repositories on the admin node.

The `configure-cluster` command calls the `crepo` command when it prompts you for media and then makes it available. You can also use the `crepo` command to add additional media.

For more information, see "crepo Command" on page 191.

cinstallman Command

You can use the `cinstallman` command to create an image from scratch, clone an existing image, update an image, show available images, delete images, assign images to nodes, manage images, and manage nodes. The `crepo` command constructs default RPM lists based on the suggested package lists. The RPM lists can be used by the `cinstallman` command when creating a new image.

When SGI releases updates, you may run `sync-repo-updates` to download the updated packages that are part of a patch. The `sync-repo-updates` command

automatically positions the files properly under `/tftpboot` (see "Overview of Installing SMC for SGI ICE Patches" on page 126).

Once the local repositories contain the updated packages (see "Update the Local Package Repositories on the Admin Node" on page 127), it is possible to update the various SGI ICE admin, leader, and managed service node images using the `cinstallman` command. The `cinstallman` command is used for all package updates including those within images, running nodes, including the admin node itself (see "Installing Updates on Running Admin, Leader, and Service Nodes " on page 132).

The `cinstallman` command does not operate on running compute nodes. For compute nodes, it is an image management tool only. You can use it to create and update compute images and use the `cimage` command to push those images out to leader nodes.

For more information on the `cinstallman` command, see "cinstallman Command" on page 195.

cimage Command

The `cimage` command processes images for diskless booting of the compute nodes. The `cimage` command pushes compute node images to rack leader controllers (RLCs) and uses `rsync` to synchronize the compute images to the RLCs. It selects which compute nodes to boot including the kernel choice and uses NFS boot or `tmpfs` boot.

For NFS boots, the compute nodes mount a read-only copy of the image from the leader that is shared by all compute nodes using the same image. During startup, other writable areas that are unique to each compute node are mounted on top of root. Things like `/etc`, `/var`, and so on. The global read-only image and the per-node read/write areas are all managed by the `cimage` command when an image is pushed from the admin node to the RLC.

For `tmpfs` boots, a version or a read-only version of the compute node image is stored in a tar ball. Each compute node uses multicast TFTP to transfer the tar ball. The tar ball is expanded into the `tmpfs` root memory and the read/write areas synchronized with the new image.

The `cimage` command allows you to list, modify, and set software images on the compute nodes in your system.

For more information on the `cimage` command, see "cimage Command" on page 206.

Networks

This section describes the Gigabit Ethernet (GigE) and 10/100 Ethernet connections and the InfiniBand fabric in an SGI ICE system and covers the following topics:

- "Networks Overview" on page 18
- "Gigabit Ethernet (GigE) and 10/100 Ethernet Connections" on page 20
- "VLANs" on page 22
- "InfiniBand Fabric" on page 26

Networks Overview

This section describes the various network connections in the SGI ICE system. Users access the system via a public network through services nodes such as the login node and the batch service node, as shown in Figure 1-5 on page 19. A single service node can provide both login and batch services.

System administrators provision (install software) and manage the SGI ICE system via the logical VLAN network running over the GigE connection (see Figure 1-7 on page 23, Figure 1-8 on page 24, and Figure 1-9 on page 25). The system admin controller (admin node) is on the house network (public network) and you access it directly.

The rack leader controller (leader node) provides boot and root filesystem images for the compute nodes in the same rack. The leader node is connected to blades in its rack via the GigE VLAN. It is connected to all service nodes and all other leader nodes via the InfiniBand fabric. Leader nodes have access to compute nodes in other racks via the leader node in that rack.

The gateway service node is the gateway from the InfiniBand fabric to services such as storage, lightweight directory access protocol (LDAP) services, file transfer protocol (FTP), and so on, on the public network. Typically, it is combined with the login/batch service node.

The system admin controller (admin node) and service nodes communicate with the leader node over a GigE fabric that has logically separate, virtual local area networks (VLANs). This GigE fabric is embedded in the backplane of each IRU. This GigE fabric electrically connects much of the SGI ICE system (see Figure 1-5 on page 19).

Users access compute nodes strictly from the service nodes. Jobs are started on compute nodes using commands on the service node, such as, the OpenSSH client remote login program `ssh(1)`, the submit a script to create a batch job `qsub(1)`

command, or the `pdsh(1)` command (see "pdsh and pdcop Utilities" on page 231) that enables the execution of any standard command on all SGI ICE system nodes.

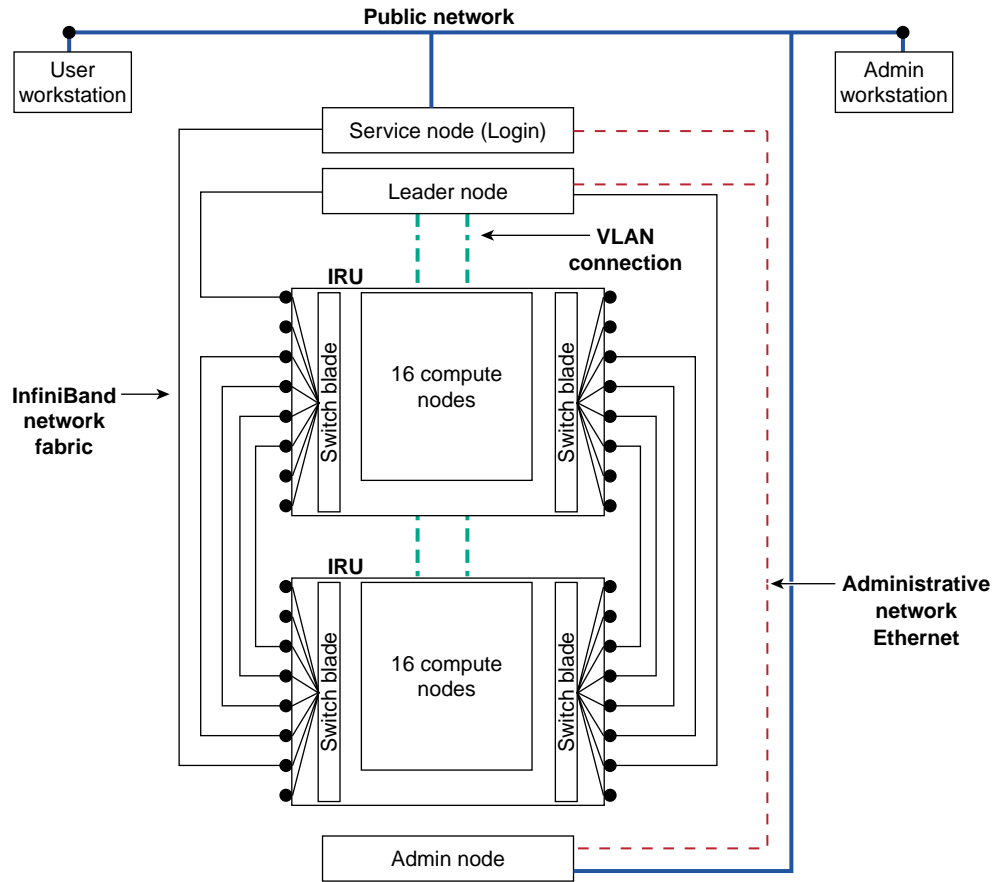


Figure 1-5 Network Connections In a System With Two IRUs

You can use the interconnect verification tool (IVT) to verify that all the various 10/100 Ethernet, Gigabit Ethernet (GigE), and InfiniBand (IB) network links between the various system admin controllers (admin nodes), such as the admin or login node, the leader node, the compute nodes, the CMC and the BMC nodes are correctly connected and working properly after a system is installed or for maintenance purposes. For more information on IVT, see "Inventory Verification Tool" on page 314.

Gigabit Ethernet (GigE) and 10/100 Ethernet Connections

The SGI SGI ICE 8200 system has several Ethernet networks that facilitate booting and managing the system. These networks are built onto the backplane of each IRU for connection to the compute blades and transverse cables between IRUs and between racks. Each compute blade has a Gigabit Ethernet (GigE) and 10/100 Ethernet connection to the backplane.

The GigE connection is an interface that is accessible to the operating system and the basic input/output (BIOS) running on the blade. It is the interface over which the BIOS uses the preboot execution environment (PXE) to PXE boot and it is known as `eth0` on the configured node.

The 10/100 Ethernet interface is accessible to the management interface (BMC) built onto each compute blade. The operating system running on the blade cannot directly access this 10/100 interface. It belongs to the processor on the BMC. Likewise, the BMC cannot access the GigE interface.

Figure 1-6 on page 20 shows a more detailed view of the Chassis manager.

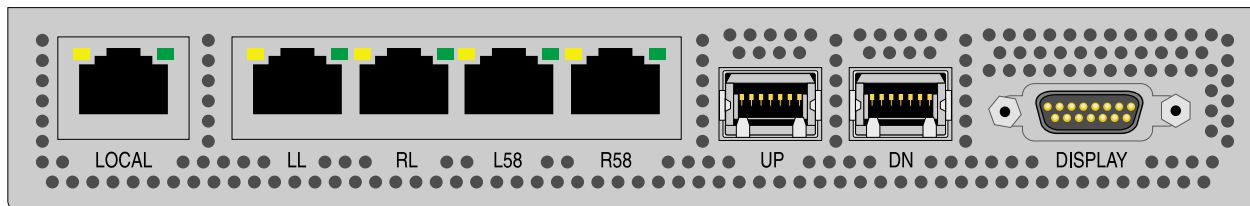


Figure 1-6 Chassis Manager

The chassis management control (CMC) blade has two embedded Ethernet switches. One is a 24-port GigE switch and the other a 24-port 10/100 switch. The 10/100 switch is a sub-switch, hanging off one port, of the GigE switch.

The primary GigE interface from each of sixteen blades connects to the GigE switch and the sixteen blade BMCs connect to the 10/100 switch. The GigE connections also connect the service nodes, including the service storage nodes.

The GigE switches in each IRU are "stacked" using a special stacking connection between each IRU in a rack. This connection runs a special intra-switch protocol. All switches in a rack are ganged together to form one large 96 port switch. The connections from each CMC to another are labeled **UP** and **DN** as shown in Figure 1-6

on page 20. The switches are stacked in a ring. The stacking ring is redundant and works in one direction, at a time, and if one direction breaks, it goes the other way around to ensure connectivity is preserved.

The processor on the CMC manages these switches effectively forming a large, intelligent Ethernet switch. A VLAN mechanism runs on top of this network to allow management control software to query port statistics and other port metrics including the attached peer's MAC address.

The CMC has five additional RJ45 connections on its front panel as shown in Figure 1-6 on page 20. The function of these jacks is, as follows:

- **Local**

This is a connection to the leader node at the top of the rack in which this CMC is located. Only one CMC (of the possible four) is connected to the leader node, as shown in Figure 1-3 on page 9.

- **LL**

Used to connect service nodes and service storage nodes. The RL jack in the far left CMC connects to the LL jack of the right adjacent CMC to create or grow the Ethernet network. Figure 1-3 on page 9 shows this daisy chaining.

- **RL**

Used to connect service nodes and service storage nodes. The RL jack in the far left CMC connects to the LL jack of the right adjacent CMC to create or grow the Ethernet network. Figure 1-3 on page 9 shows this daisy chaining.

- **L58**

This is a connection for the IEEE 1588 timing protocol from this CMC to the one immediately to the left. If this is the left-most rack, this jack is unconnected.

- **R58**

This is a connection for the IEEE 1588 timing protocol from this CMC to the one immediately to the right. If this is the right-most rack, this jack is unconnected.

A NAS cube storage service node uses both the **LL** and **RL** jacks to connect to the SGI ICE system as shown in Figure 1-4 on page 14.

For small, one IRU configurations, the **L58** and **R58** ports (see Figure 1-6 on page 20) can be used to connect service nodes. This functionality can be enabled using the

LCD panel of the CMC. It can also be done in the factory or by your SGI system support engineer (SSE).

VLANs

Several virtual local area networks (VLANs) are used to isolate Ethernet traffic domains within the cluster. The physical Ethernet is a shared network that has a connection to every node in the cluster. The admin node, leader nodes, service nodes, compute nodes, CMCs, BMCs, all have a connection to the Ethernet. To isolate the broadcast domains and other traffic within the cluster, VLANs are used to partition it and are, as follows:

- VLAN_1588

Includes all `1588_left` and `1588_right` connections, as well as an internal port to the CMC processor. This VLAN carries all of the IEEE 1588 timing traffic.

- VLAN_HEAD

Includes all `leader_local`, `leader_left`, and `leader_right` connections. The `VLAN_HEAD` VLAN connects the admin node to all of the leader nodes (including the leader nodes' BMCs) and the service nodes.

- VLAN_BMC

Includes all 10/100 sub-switches and the `leader_local` ports. The `VLAN_BMC` VLAN connects the leader nodes to all of the BMCs on the compute blades and to the CMCs within each IRU. See Figure 1-7 on page 23.

- VLAN_GBE

Includes all GigE blade ports and the `leader_local` port. The `VLAN_GBE` VLAN connects the leader nodes to the GigE interfaces of all the compute blades. See Figure 1-7 on page 23.

`VLAN_GBE` and `VLAN_BMC` do not extend outside of any rack. Therefore, traffic on those VLANs stays local to each rack.

Only `VLAN_HEAD` extends rack to rack. It is the network used by the admin node to communicate to the leader node of each rack and to each service node.

The rack leader controllers (leader nodes) must run 802.1Q VLAN protocol over their downstream GigE connection to the CMC and the CMC LL port must also run 802.1Q. This is done for you when the rack leader controllers are installed from the

system admin controller. For more information, see "Installing SMC for SGI ICE Admin Node Software" on page 42. Each VLAN should present itself as a separate, pseudo interface to the operating system kernel running on that leader node. VLAN_HEAD, VLAN_BMC, and VLAN_GBE must all transition the single Ethernet segment which connects the leader to the CMC in the rack below it.

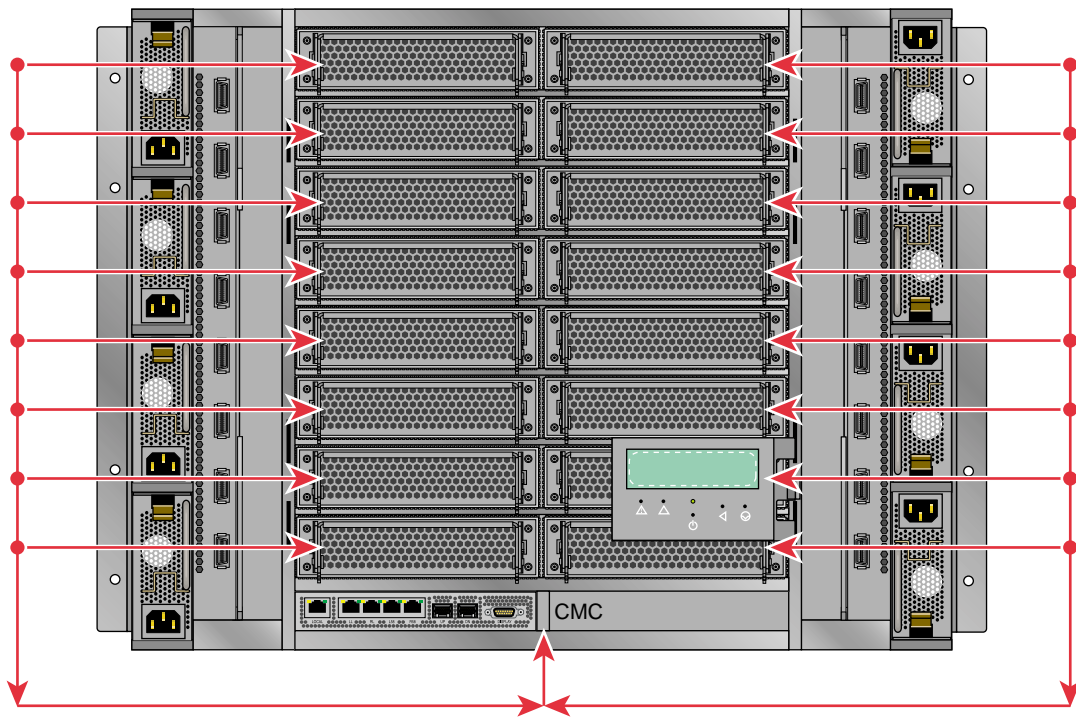


Figure 1-7 VLAN_GBE and VLAN_BMC Network Connections - IRU View

The VLAN_GBE and VLAN_BMC networks connect the leader node in a given rack with the compute nodes (blades). In the case of VLAN_BMC, the network also connects the CMC with the compute blades and rack leader controller (leader node).

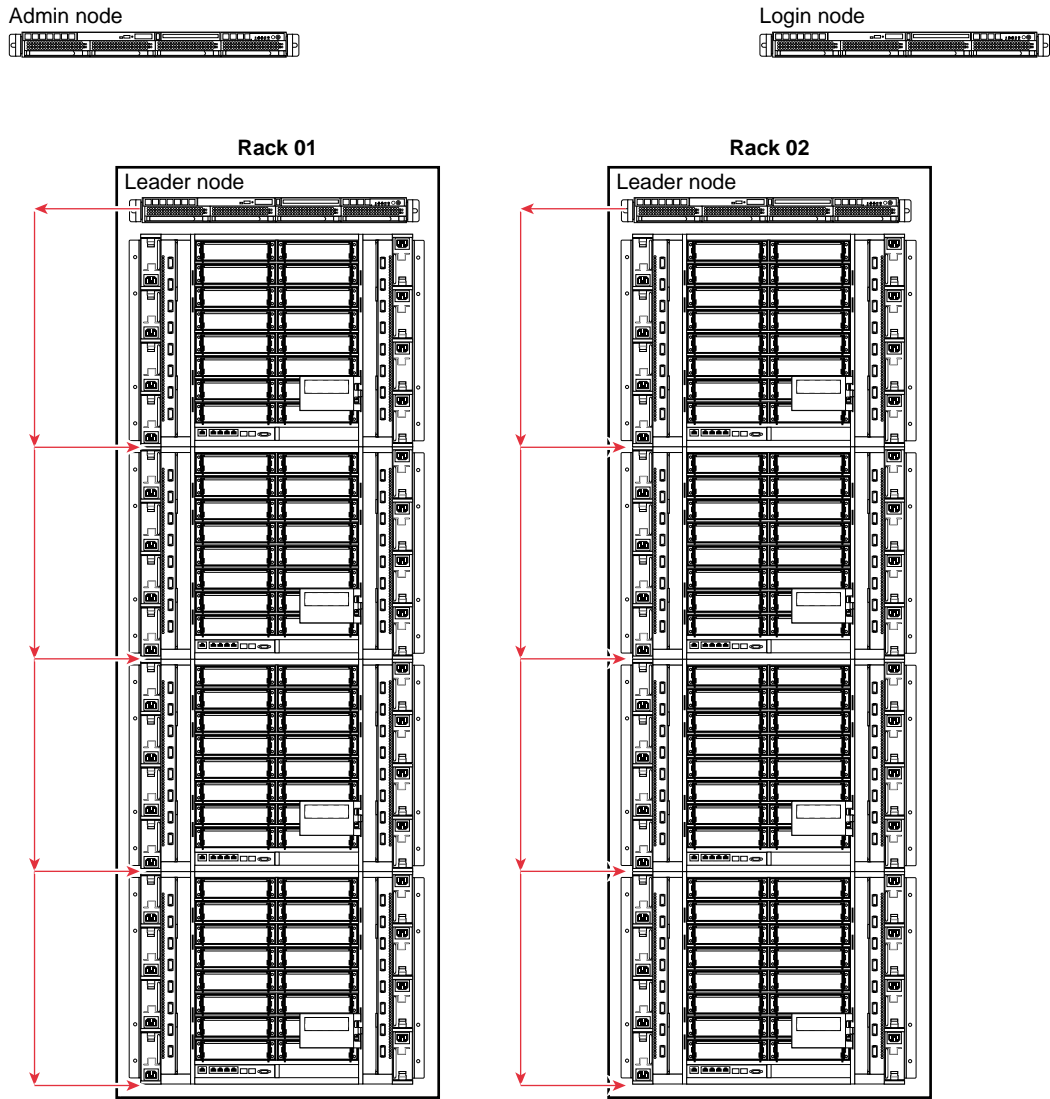


Figure 1-8 VLAN_GBE and VLAN_BMC Network Connections – Rack View

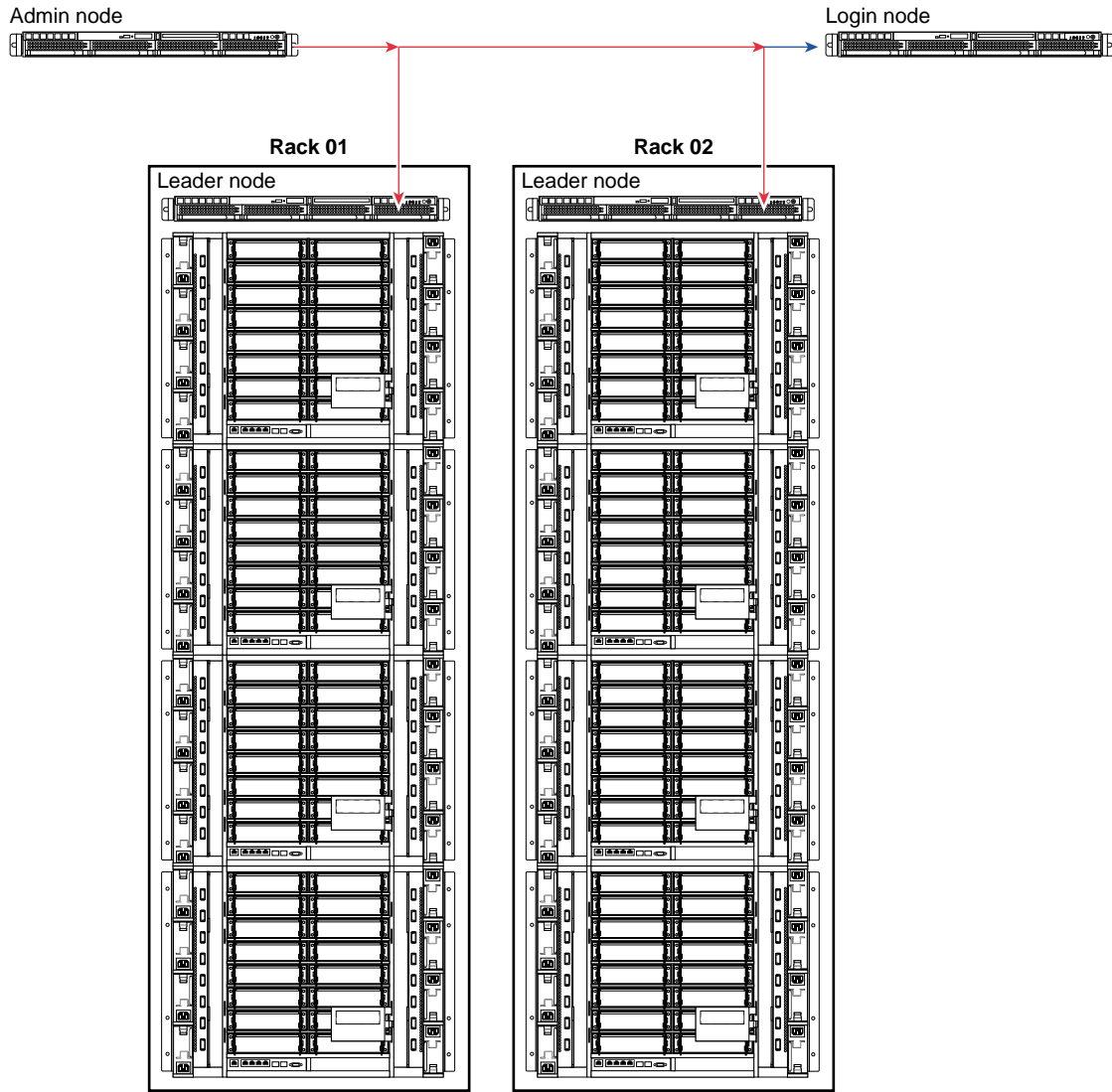


Figure 1-9 VLAN_HEAD Network Connections

In an SGI ICE system with just one IRU, the CMC's R58 and L58 ports are assigned to VLAN_HEAD by a field configurable setting. This provides two additional Ethernet

ports that can be use to connect service nodes to your system. This is done in the factory or by your SGI system support engineer (SSE).

For information on the CMC interface LCD panel shown just about the CMC in Figure 1-7 on page 23, see chapter 1 and chapter 6 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

InfiniBand Fabric

An InfiniBand fabric connects the service nodes, leader nodes, and compute nodes. It does not connect to the admin node or the CMCs. SGI SGI ICE systems usually have two separate network fabrics, `ib0` and `ib1`.

Note: The LX series only has one ib fabric, "`ib0`". Any references to "`ib1`" in what follows do not apply to LX systems

On SGI ICE 8200 series systems, each rack leader controller (RLC), also called a leader node, has an InfiniBand host channel adapter (HCA) with two ports, each of which connects to a different fabric (see Figure 1-10 on page 27).

Each IRU has internal InfiniBand switches which interconnect a fabric within the IRU (see Switch blade in Figure 1-10 on page 27). A particular switch is part of only one fabric.

For a particular switch, each of 16 switch ports connects to one of the 16 compute nodes within the IRU. Some of the remaining switch ports are used for interconnections within the IRU, and the rest of the ports are exposed via connectors on the front of the IRU. InfiniBand cables between these connectors link the fabric between different IRUs, and one IRU in a rack is connected directly to its rack leader node (see Figure Figure 1-10 on page 27).

On SGI ICE 8400 series systems, rack leader controllers (leader nodes) will not always have InfiniBand fabric host channel adapters (HCA) depending on the system configuration. In some cases, one to two RLCs will have HCAs to run the OFED subnet manager. In other cases, this will be done on separate fabric management nodes, in this case no RLCs will have InfiniBand HCAs.

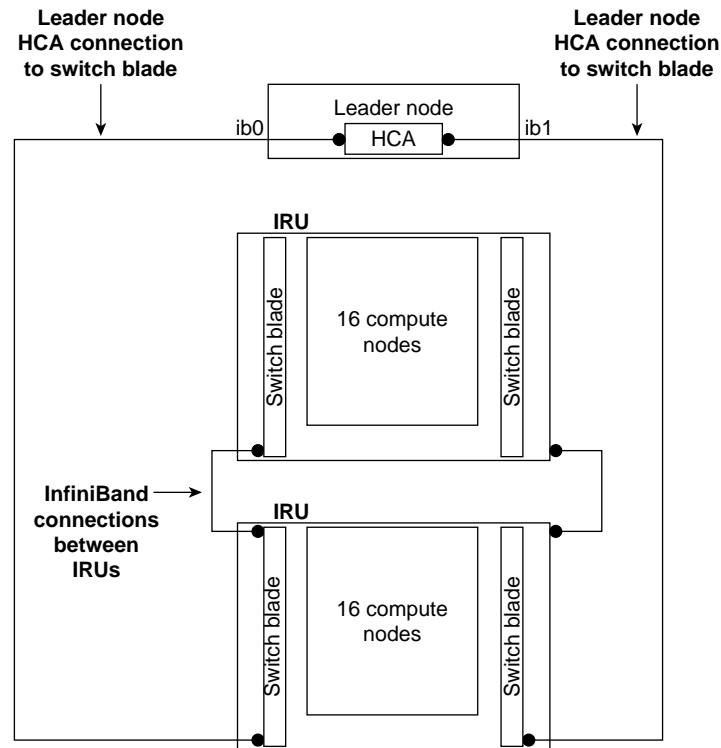


Figure 1-10 Two InfiniBand Fabrics in a System with Two IRUs

Network Interface Naming Conventions

As described in "Networks" on page 18, you can think of an SGI ICE system as having two distinct networks, the connections between the admin nodes, service nodes, and leader nodes, and the connections between the compute blades, CMCs, and the leader node within each rack. In general, these connections are made over one of the VLAN networks described in "VLANs" on page 22, but it is useful to be able to specify over which interface (VLAN) you are attempting to communicate. This section describes the naming strategy for logical type of interface being used. It covers the following topics:

- "System Component Names" on page 28
- "VLAN_Head Network Connections" on page 28

- "VLAN_GBE Network Connections" on page 29
- "VLAN_BMC Network Connections" on page 30
- "VLAN_1588 Network Connections" on page 31
- "Non-resolvable Names" on page 31
- "Hostnames" on page 32
- "InfiniBand Network" on page 33

System Component Names

Even though you may be communicating on different VLANs, you may in fact be communicating with the same physical network interface on the system. Naming the logical connections by function allows flexibility to change the number or type of the underlying physical networks. At the topmost level, the admin and service node nodes can communicate with the leader nodes over the `VLAN_HEAD` virtual network. The system component terms used in this section are described, as follows:

Node	Refers to a building block within an SGI ICE system (see "System Nodes" on page 9)
Connection name	Denotes a resolvable name associated with an IP network
Node name	Represents system-wide unique identifier for the building blocks of the SGI ICE system. These IDs are partly not routable. See "Non-resolvable Names" on page 31.
Hostname	Returns string of the hostname command. Is technically independent from the other names.

System-wide unique names are node names and non-resolvable names.

X, Y, and Z in the following tables in this section are all integers.

VLAN_Head Network Connections

Table 1-1 on page 29 shows the `VLAN_Head` network connection names. See Figure 1-9 on page 25.

Table 1-1 VLAN_HEAD Connections

Node	Connection Name
Admin	admin
Service	serviceX serviceX-bmc
Leader	rXlead rXlead-bmc

There is one admin node per system. You can have multiple service nodes labelled `service0`, `service1`, and so on. The BMC controllers for managed service nodes are accessible inside the network. BMCs for unmanaged service nodes are normally configured on the external network. For more information on managed service nodes, see "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86.

VLAN_GBE Network Connections

Table 1-2 on page 29 shows the VLAN_GBE network connections.

Table 1-2 VLAN_GBE Network Connections.

Node	Connection Name	Node Name
Leader	lead-eth	rXlead
CMC	iYc	rXiYc
Blade	iYnZ-eth	rXiYnZ

The GBE VLAN is entirely internal to each rack (see Figure 1-7 on page 23). The naming scheme is replicated between each rack, so the name `i2n4-eth` (identifying the VLAN_GBE interface on IRU 2, node 4) may match several different nodes, but only ever one in each rack. To identify a node uniquely, use the `rXiYnZ` syntax.

Blade rXiYnZ names are resolvable via DNS. They get the A record for the -ib0 address. The rXiYnZ-ib0 name is a CNAME to the rXiYnZ address. For example:

```
[root@sys-admin ~]# host r1i1n0
r1i1n0.ice.americas.sgi.com has address 10.148.0.20

[root@r1lead ~]# host r1i1n0
r1i1n0.ice.americas.sgi.com has address 10.148.0.20
```

VLAN_BMC Network Connections

Table 1-3 on page 30 shows the VLAN_BMC network connections.

Table 1-3 VLAN_BMC Network Connections

Node	Connection Name	Node Name
Leader	lead-bmc	rXlead
CMC	iYc	rXiYc
Blade	iYnZ-bmc	rXiYc

The BMC VLAN is also local to each rack, in the same way as the GBE VLAN (see Figure 1-7 on page 23).

Note that the interface lead-bmc on the leader node is not an interface to the BMC on the leader, but rather is an interface on the leader to the VLAN_BMC network in that leaders rack. Software running on other nodes in an SGI ICE system, outside of a given rack, cannot directly address the BMC's, or CMC, within said rack. Rather such requests much go through suitable application level software running on that rack's leader, when can in turn access the BMCs and CMC in its rack, via this lead-bmc interface to the racks VLAN_BMC network.

Connecting to the leader node's BMC is only possible from an admin node, service, or other leader node, when you should use rXlead-bmc.

The CMC does not have a BMC connection, but instead the VLAN_BMC connection is to the CMC's console interface.

VLAN_1588 Network Connections

Table 1-4 on page 31 shows the VLAN_1588 network connections.

Table 1-4 VLAN_1588 Network Connections

Node	Connection Name	Node Name
CMC	rXiYc-1588	rXiYc-1588

The 1588 VLAN carries the time synchronization traffic and connects CMCs in all the racks in the SGI ICE system. For this reason, the full rack-qualified name is needed to uniquely identify the target CMC.

Non-resolvable Names

Sometimes a rack, an IRU, or a CMC needs to be uniquely identified within the SGI ICE system. Table 1-5 on page 31 shows the names that may be used for this, but there is no IP address associated with them. Therefore, DNS lookup will not succeed for these names. The names are used by certain SGI ICE management tools and are parsed internally to indicate which leader node to use in order to connect to the destination system.

Table 1-5 Non-resolvable Names

Node	Node Name
Rack	rX
IRU	rXiY
CMC	rXiYc

Hostnames

Hostnames are distinct from the non-resolvable names and are shown in Table 1-6 on page 32. In general, this is the name that you get by typing `hostname` at the command prompt on the system, and is used as a way of identifying the system to the user. Often, the command prompt is set up to contain the hostname. This is a benefit since with multiple windows open to different systems, it allows the user to avoid executing commands in the wrong window.

Table 1-6 Hostnames

Node	Hostnames
Admin	user assigned
Leader	rXlead
Blade	rXiYnZ
CMC	rXiYc
Service	user assigned (see Note below)

Note: By default, the host name for service nodes follow the convention `serviceX`. However, host names of service nodes or admin nodes can be changed using the `cadmin` command (see "cadmin: SMC for SGI ICE Administrative Interface" on page 232).

The internal domain name service (DNS) has changed. The hostname gets the `A` record and name `-ib0` gets a `CNAME` alias. Additionally, if you changed the hostname from the SMC for SGI ICE node name, there will be `CNAME` alias for the SMC for SGI ICE node name, as well.

The zone looks similar to the following:

```
r1lead          IN      A       10.148.0.1
r1lead-ib0     IN      CNAME   r1lead.ice.mycompany.com.
r1lead-ib1     IN      A       10.149.0.1
r1i0n0         IN      A       10.148.0.2
r1i0n0-ib0    IN      CNAME   r1i0n0.ice.mycompany.com.
r1i0n0-ib1    IN      A       10.149.0.2
```

```

r1i0n1           IN      A      10.148.0.3
r1i0n1-ib0      IN      CNAME  r1i0n1.ice.mycompany.com.
r1i0n1-ib1      IN      A      10.149.0.3
[...]

```

In the example above, the node/hostname gets the A record. The `-ib0` name is a CNAME alias to the node/hostname. `ib1` remains same as previous releases.

InfiniBand Network

The InfiniBand fabric is connected to service nodes, rack leader controllers (leader nodes), and compute nodes, but not to the system admin controller (admin node) or CMCs. Table 1-7 on page 33 shows InfiniBand names. There are two IB connections to each of the nodes that use it. Since IB is not local to each rack, you must use the fully-qualified, system-unique node name when specifying a destination interface. It may be necessary to alias the `rXiYnZ` names (currently non-resolvable) to `rXiYnZ-ib0` if this is needed by MPI. Technically, `rXiYnZ` from a leader node points at the `VLAN_GBE` interface for the compute blade while from a service or compute blade, `rXiYnZ` points to the `ib0` interface.

In DNS, the `rXiYnZ` name is the A record, with the `-ib0` address, `rXiYnZ-ib0`, the CNAME alias to the `rXiYnZ` A record. The same applies to service nodes (see "Hostnames" on page 32).

If you change the node name, the new name is the A record, with the `-ib0` address, `newname-ib0`, the CNAME alias to the new name A record. The old name is a CNAME alias to the new name A record.

Table 1-7 InfiniBand Names

Node	Connection Name	Node Name
Service	serviceX-ib0 serviceX-ib1	serviceX
Leader	rXlead-ib0 rXlead-ib1	rXlead
Blade	rXiYnZ-ib0 rXiYnZ-ib1	rXiYnZ

Note: The host name of a service node can be changed from the default.

System Discovery, Installation, and Configuration

This chapter describes how to use the SGI Management Center (SMC) for SGI ICE systems management software to discovery, install, and configure your SGI ICE system and covers the following topics:

- "Configuring Factory-installed SGI ICE System" on page 36
- "Accessing SGI Management Center" on page 39
- "Overview of Installing Software and Configuring Your SGI ICE System" on page 41
- "Installing SMC for SGI ICE Admin Node Software" on page 42
- "cmcdetectd Daemon" on page 79
- "discover Command" on page 80
- "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86
- "blademon Command For Automatic Blade Discovery" on page 89
- "Discovering Compute Nodes" on page 91
- "Service Node Discovery, Installation, and Configuration" on page 92
- "InfiniBand Configuration" on page 96
- "MySQL Database Server Replication" on page 102
- "Redundant Management Network" on page 98
- "Configuring the Service Node" on page 105
- "RHEL Service Node House Network Configuration " on page 116
- "Setting Up an NFS Home Server on a Service Node for Your SGI ICE System" on page 111
- "Setting Up a NIS Server for Your SGI ICE System" on page 117
- "Installing SMC for SGI ICE Patches and Updating SGI ICE Systems " on page 125
- "Installing MPI on a Running SGI ICE System" on page 144

- "System Admin Controller High Availability Solution" on page 147
 - "Rack Leader Controller High Availability Solution" on page 158
-

Note: If you are upgrading from a prior release or installing SMC for SGI ICE software patches, see "Installing SMC for SGI ICE Patches and Updating SGI ICE Systems " on page 125.

Configuring Factory-installed SGI ICE System

This section describes what you should do if you wish to use the pre-installed software on the system admin controller (admin node).

Procedure 2-1 Configuring Factory-installed SGI ICE System (SLES)

You can use the `cpasswd` command to change the password on an already deployed system. You can use this command to change the global default password and per-node passwords with this command.

For a `cpasswd` usage statement, perform the following:

```
system-admin:~ # cpasswd --h
```

Usage:

```
cpasswd [OPTION]...
```

Options:

<code>--debug</code>	print debugging output
<code>-h, --help</code>	print usage and exit
<code>-N, --node=NODE</code>	change password for NODE
<code>--no-encrypt</code>	do not encrypt the password, it is already encrypted
<code>-q, --quiet</code>	only print errors
<code>--reset</code>	reset to the current global password
<code>--stdin</code>	read password from stdin
<code>--salt-id=ID</code>	the salt id to use when encrypting

Details:

If a node is specified with `-N` or `--node`, then that node will have a password that differs from the global password. In order to remove a node-specific password, use `--reset`.

Encrypting the password is done with the standard `crypt(3)` function, and therefore the salt ids supported are the same as specified in the `crypt(3)` man page. The default is 1 (MD5).

To configure the software that comes pre-installed on a SLES SMC for SGI ICE admin node, perform the following steps:

1. Use YaST to configure the first interface of the admin node for your house network. Settings to adjust may include the following:
 - Network settings including IP, default route, and so on
 - Time zone
2. If you need to adjust SGI ICE settings, such as, any internal network ranges, modify the timezone, you will need to reset the database and rediscover the leader nodes and service nodes, as follows:
 - a. Start the `configure-cluster` command (see "configure-cluster Command Cluster Configuration Tool" on page 56).
 - b. Use `configure-cluster` to fix the `networks.conf` file and then the `cadmin --set-subdomain` command to change the sub-domain name.

Note: If you need to change the network ranges, modify the timezone, you need to choose the **Reset Database** operation. Read the on-screen instructions. After the database has been reset, choose **Initial Setup Menu**.

- c. Choose **Initial Setup Menu**.
- d. Start the options in this menu in order starting at **Perform Initial Admin Node Infrastructure Setup**. Note that if you are changing any network ranges or the cluster subdomain, you should choose **Network Settings** before proceeding to **Perform Initial Admin Node Infrastructure Setup**.

Note: You will get a message about the systemimager images already existing. You may choose to use the existing images instead of re-creating them. This will save about 30 minutes. Either choice is OK. Do **not** choose **use existing images** if you changed the root password or time zone as these settings are stored in the image when the image is created.

- e. At this point, you can begin to discover leader and service nodes and continue cluster installation. See "discover Command" on page 80.

Procedure 2-2 Configuring Factory-installed SGI ICE System (RHEL)

You can use the `cpasswd` command to change the password on an already deployed system. You can use this command to change the global default password and per-node passwords with this command.

For a `cpasswd` usage statement, perform the following:

```
system-admin:~ # cpasswd --h
```

Usage:

```
cpasswd [OPTION]...
```

Options:

<code>--debug</code>	print debugging output
<code>-h, --help</code>	print usage and exit
<code>-N, --node=NODE</code>	change password for NODE
<code>--no-encrypt</code>	do not encrypt the password, it is already encrypted
<code>-q, --quiet</code>	only print errors
<code>--reset</code>	reset to the current global password
<code>--stdin</code>	read password from stdin
<code>--salt-id=ID</code>	the salt id to use when encrypting

Details:

If a node is specified with `-N` or `--node`, then that node will have a password that differs from the global password. In order to remove a node-specific password, use `--reset`.

Encrypting the password is done with the standard `crypt(3)` function, and therefore the salt ids supported are the same as specified in the `crypt(3)` man page. The default is 1 (MD5).

To configure the software that comes pre-installed on a RHEL SMC for SGI ICE admin node, perform the following steps:

1. Configure the first interface (`eth0`) of the admin node for your house network. Settings to adjust may include the following:
 - Network settings including IP, default route, and so on
 - Time zone

2. If you need to adjust SGI ICE settings, such as, any internal network ranges, modify the timezone, or change the system-wide root password, you will need to reset the database and rediscover the leader nodes and service nodes, as follows:
 - a. Start the `configure-cluster` command (see "configure-cluster Command Cluster Configuration Tool" on page 56).
 - b. Use `configure-cluster` to fix the `networks.conf` file and then the `cadmin --set-subdomain` command to change the sub-domain name.

Note: If you need to change the network ranges, modify the timezone, or change the system-wide root password, you need to choose the **Reset Database** operation. Read the on-screen instructions. After the database has been reset, choose **Initial Setup Menu**.

- c. Choose **Initial Setup Menu**.
- d. Start the options in this menu in order starting at **Perform Initial Admin Node Infrastructure Setup**. Note that if you are changing any network ranges or the cluster subdomain, you should choose **Network Settings before** proceeding to **Perform Initial Admin Node Infrastructure Setup**.

Note: You will get a message about the systemimager images already existing. You may choose to use the existing images instead of re-creating them. This will save about 30 minutes. Either choice is OK. Do **not** choose **use existing images** if you changed the root password or time zone as these settings are stored in the image when the image is created.

- e. At this point, you can begin to discover leader and service nodes and continue cluster installation. See "discover Command" on page 80.

Accessing SGI Management Center

SGI Management Center (SMC) software running on the system admin controller (admin node) provides a robust graphical interface for system configuration, operation, and monitoring. For more information on using SMC, see *SGI Management Center System Administrator's Guide*.

This section describes two different methods you can use to access the SMC with SMC for SG ICE 1.5 software.

Install SMC During a Fresh Install

You can install SMC using the SGI Admin Install DVD with SGI Management Center software bundle. In this method, SMC is automatically installed during infrastructure setup on the admin node (see "configure-cluster Command Cluster Configuration Tool" on page 56).

Note: SMC requires an LK license. Save the license to `/etc/lk/keys.dat`.

Install SMC on a System Running SMC for SGI ICE 1.5

This section describes how to install the SGI Management Center (SMC) software on a system already running SMC for SGI ICE 1.5 software.

Note: SMC requires an LK license. Save the license to `/etc/lk/keys.dat`.

Procedure 2-3 Install SGI Management Center

To install SGI Management Center (SMC) on a system running SMC for SGI ICE 1.5, perform the following steps:

1. Install and select the SMC repository with the `crepo`, as follows:

```
sys-admin:~ # crepo --add sgimc-1.2-cd1-media-sles11-x86_64.iso
sys-admin:~ # crepo --select SGI-Management-Center-1.2-sles11
```

2. Start `configure-cluster` (see "configure-cluster Command Cluster Configuration Tool" on page 56) and run **Install SGI Management Center (optional, if available)**. Quit `configure-cluster` when finished.

Note: This example applies to a SLES11 SP1 installation, RHEL 6 will be similar.

3. SMC requires an LK license. Save the license to `/etc/lk/keys.dat`.

4. Restart `/etc/init.d/mgr`, as follows:

```
sys-admin:~ # /etc/init.d/mgr restart
```

5. To run `mgrclient`, which will launch the SMC GUI, logout and log back in, to ensure that all environment variables are initialized appropriately.

Overview of Installing Software and Configuring Your SGI ICE System

This section provides a high-level overview of installing and configuring your SGI ICE system.

Procedure 2-4 Overview of Installing Software and Configuring Your SGI ICE System

To install and configure software on your SGI ICE system, perform the following steps:

1. Follow "Installing SMC for SGI ICE Admin Node Software" on page 42 to install software on your system admin controller (admin node).
2. Follow "`configure-cluster` Command Cluster Configuration Tool" on page 56 to configure the overall cluster.
3. Follow "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86 to install software on the leader nodes and service nodes.
4. Follow "Discovering Compute Nodes" on page 91 to discover compute nodes.
5. Follow "Service Node Discovery, Installation, and Configuration" on page 92 to discover, install and configure service nodes.
6. Ensure that all hardware components of the cluster have been discovered successfully, that is, admin, leader, service, and compute nodes and then follow "InfiniBand Configuration" on page 96 to configure and check the status of the InfiniBand fabric.
7. Follow "Configuring the Service Node" on page 105, "Setting Up an NFS Home Server on a Service Node for Your SGI ICE System" on page 111, and "Setting Up a NIS Server for Your SGI ICE System" on page 117 to complete your system setup.

Installing SMC for SGI ICE Admin Node Software

This section describes how to install software on the system admin controller (admin node). The system admin controller contains software for provisioning, administering, and operating the SGI ICE system. The SGI Admin Node Autoinstallation DVD contains a software image for the system admin controller (admin node) and contains SMC for SGI ICE and SGI Performance Suite packages, used in conjunction with the packages from the SLES11 DVD or RHEL 6 DVD to create leader, service, and compute images.

The root image for the admin node appliance is created by SGI and installed on to the admin node using the admin install DVD.

Note: If you are reinstalling the admin node, you may want to make a backup of the cluster configuration snapshot that comes with your system so that you can recover it later. You can find it in the `/opt/sgi/var/ivt` directory on the admin node; it is the earliest snapshot taken. You can use this information with the interconnect verification tool (IVT) to verify that the current system shows the same hardware configuration as when it was shipped. For more information on IVT, see "Inventory Verification Tool" on page 314.

This section covers the following topics:

- "Installing SLES11 on the Admin Node" on page 42
- "Installing RHEL 6 on the Admin Node" on page 52
- "Initial Configuration of a RHEL 6 Admin Node" on page 54
- "`configure-cluster` Command Cluster Configuration Tool" on page 56
- "Set Network Time Configuration for RHEL6 Admin Node" on page 78

Installing SLES11 on the Admin Node

When using the `configure-cluster` command and adding the SLES media, only the first DVD is needed. The second SLES DVD, that contains the source RPMs, is not required for cluster operation and cannot be added to the repositories.

Procedure 2-5 Installing SLES 11 Software on the System Admin Controller

To install SLES 11 software images on the system admin controller, perform the following steps:

1. Turn on, reset, or reboot the system admin controller. The power on button is on the right of the system admin controller, as shown in Figure 2-1 on page 43.

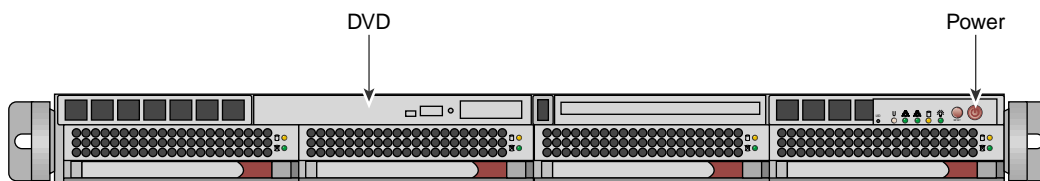


Figure 2-1 System Admin Controller Power On Button and DVD Drive

The serial console was always used even if the admin node install itself went to the VGA screen.

The new method configures the default serial console used by the system to match the console used for installation.

If you type "serial" at the Admin DVD install prompt, the system is also configured for serial console operations after installation and the `yast2-firstboot` questions appear on the serial console.

If you press `Enter` at the prompt or type `vga`, the VGA screen is used for installation, as previously, but also, the system is configured to use VGA as the default console, thereafter.

If you want to install to the VGA screen, but also want the serial console to be used for operations after initial installation, you should add a `console=` parameter to `/boot/grub/menu.lst` for each kernel line. This is done when the admin node boots for the first time after installation is completed. An example of this is, as follows:

```
kernel /boot/vmlinuz-2.6.16.46-0.12-smp root=/dev/disk/by-label/sgiroot console=ttyS1,3840n8
splash=silent showopts
```

The appropriate entries were added to the `inittab` and `/etc/security`. The change, above, is the only one needed to switch the default console from VGA to

serial. Likewise, to move from serial to VGA, simply remove the console= parameter, altogether.

2. Insert the SGI Admin Node Autoinstallation DVD in the DVD drive on the left of the system admin controller as shown in Figure 2-1 on page 43.
3. An autoinstall message appears on your console, as follows:

SGI Admin Node Autoinstallation DVD

The first time you boot after installation, you will be prompted for system setup questions early in the startup process. These questions will appear on the same console you use to install the system.

You may optionally append the "netinst" option with an nfs path to an ISO.

Cascading Dual-Boot Support:

install_slot=: install to a specific root slot, default 1
re_partition_with_slots=: re-partition with number of slot positions, up to 5.
default is 2. Reminder: applies to the whole cluster, not just admin node
destructive=1: allow slots with existing filesystems to be re-created
or signify ok to re-partition non-blank disks for re_partition_with_slots=

You may install from the vga screen or from the serial console.

The default system console will match the console you used for installation. Type "vga" for the vga screen or "serial" for serial.

Append any additional parameters after "vga" or "serial".

EXAMPLE: vga re_partition_with_slots=3 netinst=server:/mntpoint/admin.iso

Note: If you want to use the serial console, enter **serial** at the **boot:** prompt, otherwise, output for the install procedure goes to VGA screen.

It is important to note that the command line arguments you supply to the boot prompt will have implications for your entire cluster including things such as how many root slots are available, and which root slot to install to. Please read "Cascading Dual-Boot" on page 136 before you install the admin node so you are prepared for these crucial decisions.

You can hit the **ENTER** button at the boot prompt. The boot `initrd.image` executes, the hard drive is partitioned creating a swap area and a root file system, the Linux operating system and the cluster manager software is installed and a repository is set up for the rack leader controller, service node, and compute node software RPMs.

Note: When you boot the admin install DVD and choose to repartition an existing disk, all data is lost. If you are making use of cascading dual-boot (see "Cascading Dual-Boot" on page 136) and are reinstalling a given slot, the data in that slot will be deleted but the other slots will not be modified.

4. Once installation of software on the system admin controller is complete, remove the DVD from the DVD drive.
5. After the reboot completes, you will eventually see the **YaST2 - firstboot@Linux Welcome** screen, as shown in Figure 2-2 on page 45. Select the **Next** button to continue.

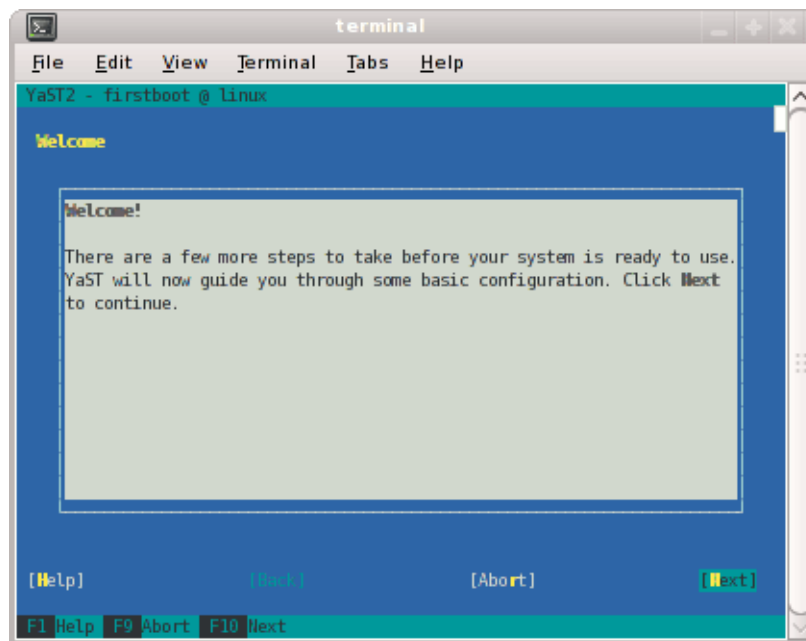


Figure 2-2 YaST2 - firstboot@Linux Welcome Screen

Note: The YaST Installation Tool has a main menu with sub-menus. You will be redirected back to the main menu, at various times, as you follow the steps in this procedure.

You will be prompted by YaST firstboot installer to enter your system details including the root password, network configuration, time zone, and so on.

6. From the **Hostname and Domain Name** screen, as shown in Figure 2-3 on page 46, enter the hostname and domain name of your system in the appropriate fields. Make sure that **Change Hostname via DHCP** is **not** selected (no **x** should appear in the box). Note that the hostname is saved to `/etc/hosts` in step 10, below. Click the **Next** button to continue.

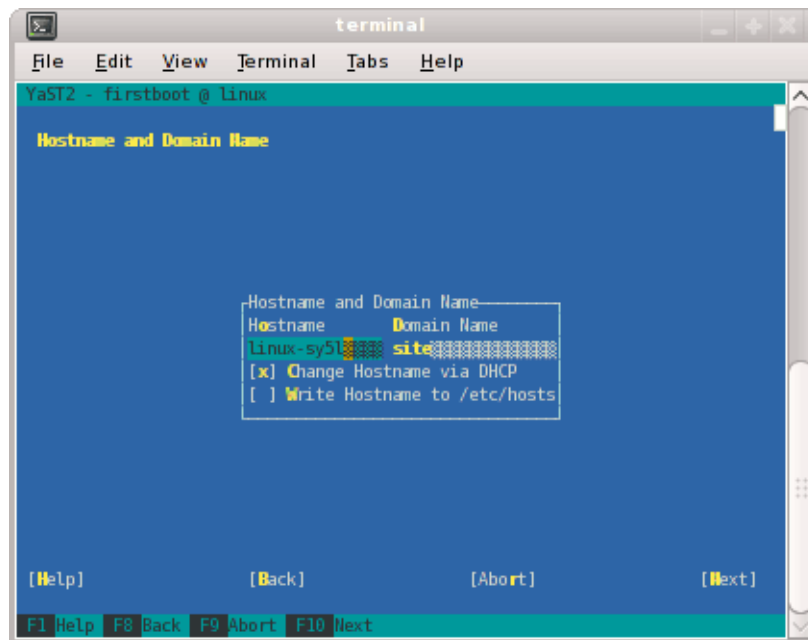


Figure 2-3 Hostname and Domain Name Screen

Note: The mostly used keys are Tab and Shift + Tab to move forward and backward in modules, the arrow keys to move up and down or left and right in lists, the shortcuts (press Alt + highlighted letter) and Enter to execute the selected action or activate a menu item.

You can use Ctrl L to refresh the YaST screen as necessary.

7. The **Network Configuration II** screen appears, as shown in Figure 2-4 on page 47. Select **Change** and a small window pops up that lets you choose **Network Interfaces...** or **Reset to Defaults**. Choose **Network Interfaces**.

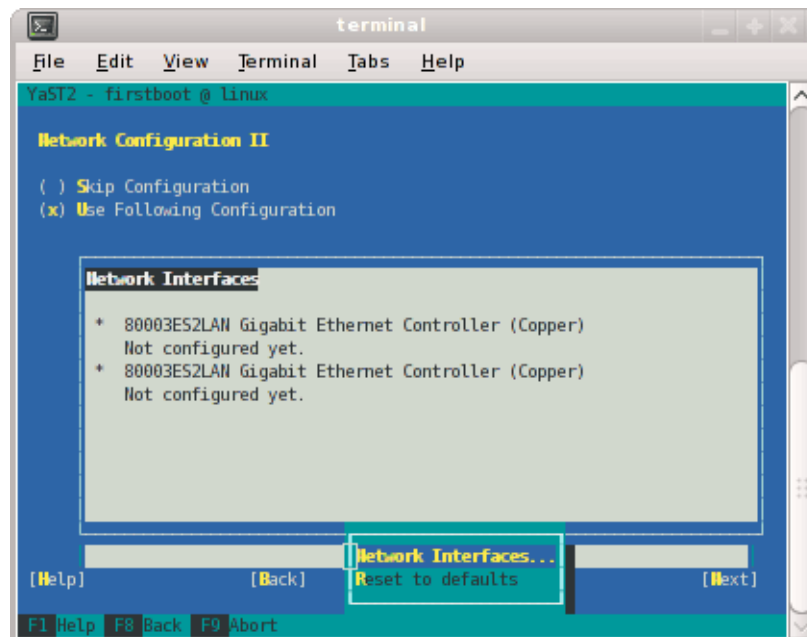


Figure 2-4 Network Configuration II Screen

8. From the **Network Settings** screen, as shown in Figure 2-5 on page 48, configure the first card under **Name** to establish the public network (sometimes called the house network) connection to your SGI ICE system. To do this, highlight the first card and select **Edit**.

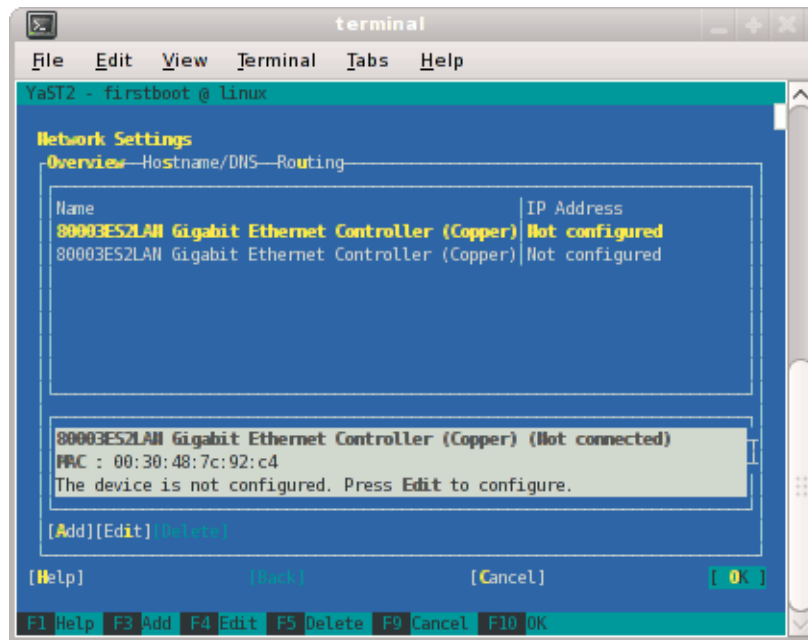


Figure 2-5 Network Settings Screen

Note: In SLES11, this screen is also where we will come back to in order to set up things like the default route and DNS. You can see all of those menu choices just to the right of **Overview** in Figure 2-5 on page 48.

9. The **Network Card Setup** screen appears, as shown in Figure 2-6 on page 49. SGI suggests using static IP addresses and not DHCP for admin nodes. Select **Statically assigned IP Address**. Once selected, you can enter the IP Address, Subnet Mask, and Hostname.

Note: You must use a fully qualified hostname (host + domain), such as, *mssystem-admin.domainname.mycompany.com* or the `configure-cluster` command will fail.

These are the settings for your admin node's house/public network interface. You will enter the default route, if needed, in a different step. Select **Next** to continue.

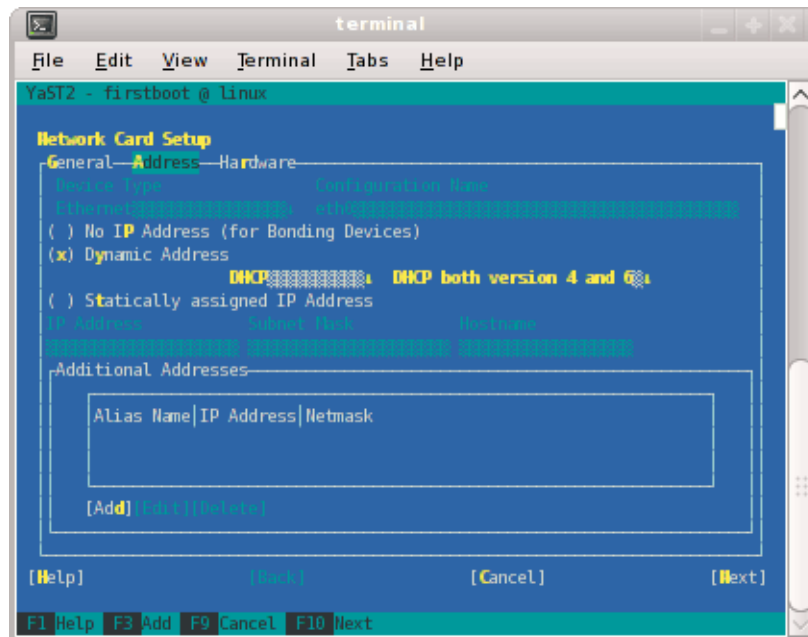


Figure 2-6 Network Card Setup Screen

10. At this point, you are back at the **Network Settings** screen as shown in Figure 2-7 on page 50. At this time, select **Hostname/DNS**. In this screen, you should enter your house/public network hostname and fully qualified domain names. In addition, any name servers for your house/public network should be supplied. Please select (ensure an x is in the box) for **Write hostname to /etc/hosts**. Do **not** select **OK** yet.

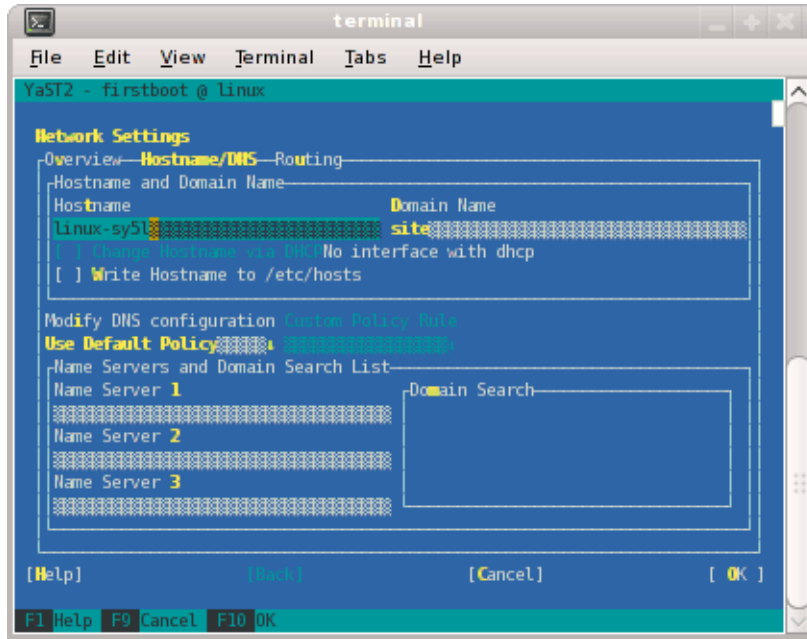


Figure 2-7 Network Settings Screen

11. Select **Routing** shown in Figure 2-8 on page 51 and enter your house/public network default router information there. Now you can select **OK**.

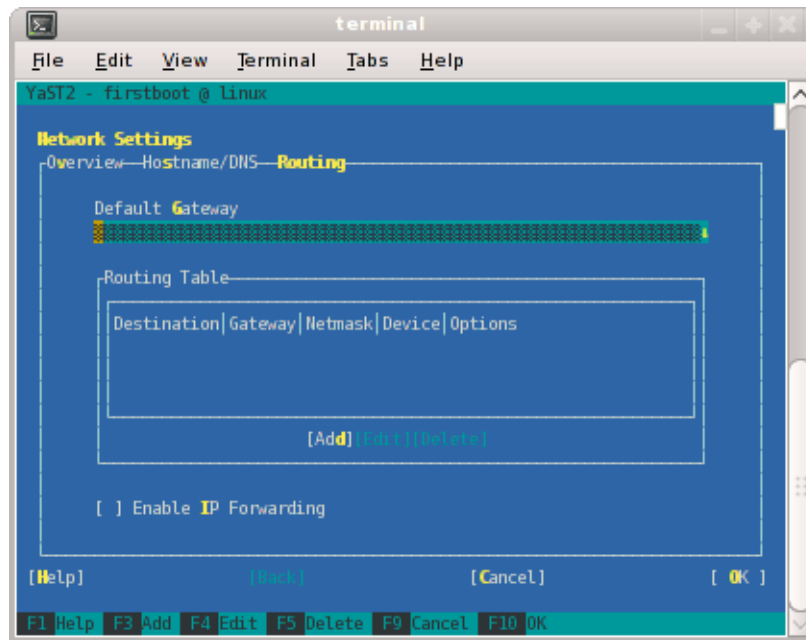


Figure 2-8 Network Settings Routing Screen

12. You are now back at the **Network Configuration II** screen, Click **Next**.
13. In the **Clock and Time Zone** screen, you can enter the appropriate details. Select **Next** to continue.
14. In the **Password for the System Administrator "root"** screen, enter the password you wish to use. This password will be used **throughout** the cluster, not just the admin node. Select **Next** to continue.
15. In the **User Authentication Method** screen, most customers will want to stick with the default (**Local**). Select **Next** to continue.
16. In the **New Local User** screen, you can just select **Next** (and say **Yes** to the **Empty User Login** warning). Select **Next** to continue.
17. In **Installation Completed**, select **Finish**.

18. After you have completed the YaST first boot installation instructions, login into the system admin controller. You can use YaST to confirm or correct any configuration settings.

Note: It is important that you make sure that you network settings are correct before proceeding with cluster configuration

19. You are now ready to run the `configure-cluster` command.

The `configure-cluster` command does not always draw the screen properly from the serial console. Therefore, log in to your admin node as root using `ssh` prior to running the `configure-cluster` command. For more information on the `configure-cluster` command, see "configure-cluster Command Cluster Configuration Tool" on page 56.

Installing RHEL 6 on the Admin Node

Use the `configure-cluster` command and RHEL media to install software on the admin node

Procedure 2-6 Installing RHEL 6 Software on the System Admin Controller

To install RHEL 6 software images on the system admin controller, perform the following steps:

1. Turn on, reset, or reboot the system admin controller. The power on button is on the right of the system admin controller, as shown in Figure 2-1 on page 43.
2. An autoinstall message appears on your console, as follows:

```
SGI Admin Node Autoinstallation DVD
```

```
The first time you boot after installation, you will be prompted for
system setup questions early in the startup process.  These questions will
appear on the same console you use to install the system.
```

```
You may optionally append the "netinst" option with an nfs path to an ISO.
```

```
Cascading Dual-Boot Support:
```

```
install_slot=: install to a specific root slot, default 1
re_partition_with_slots=: re-partition with number of slot positions, up to 5.
```

default is 2. Reminder: applies to the whole cluster, not just admin node
destructive=1: allow slots with existing filesystems to be re-created
or signify ok to re-partition non-blank disks for re_partition_with_slots=

You may install from the vga screen or from the serial console.
The default system console will match the console you used for
installation. Type "vga" for the vga screen or "serial" for serial.
Append any additional parameters after "vga" or "serial".
EXAMPLE: vga re_partition_with_slots=3 netinst=server:/mntpoint/admin.iso

Note: If you want to use the serial console, enter **serial** at the **boot:** prompt, otherwise, output for the install procedure goes to VGA screen.

It is important to note that the command line arguments you supply to the boot prompt will have implications for your entire cluster including things such as how many root slots are available, and which root slot to install to. Please read "Cascading Dual-Boot" on page 136 before you install the admin node so you are prepared for these crucial decisions.

You can hit the **ENTER** button at the boot prompt. The boot `initrd.image` executes, the hard drive is partitioned creating a swap area and a root file system, the Linux operating system and the cluster manager software is installed and a repository is set up for the rack leader controller, service node, and compute node software RPMs.

Note: When you boot the admin install DVD and choose to repartition an existing disk, all data is lost. If you are making use of cascading dual-boot (see "Cascading Dual-Boot" on page 136) and are reinstalling a given slot, the data in that slot will be deleted but the other slots will not be modified.

3. Once installation of software on the system admin controller is complete, remove the DVD from the DVD drive.
4. Proceed to the next section ("Initial Configuration of a RHEL 6 Admin Node" on page 54), to configure RHEL 6 on the admin node.

Initial Configuration of a RHEL 6 Admin Node

This section describes how to configure Red Hat Enterprise Linux 6 on the system admin controller (admin node).

Procedure 2-7 Initial Configuration of a RHEL 6 Admin Node

To perform the initial configuration of a RHEL6 Admin node, perform the following steps:

1. Add the `IPADDR`, `NETMASK`, and `NETWORK` values appropriate for the public (house) network interface to the `/etc/sysconfig/network-scripts/ifcfg-eth0` file similar to the following example:

```
IPADDR=128.162.244.88
NETMASK=255.255.255.0
NETWORK=128.162.244.0
```

2. Create the `/etc/sysconfig/network` file similar to the following example:

```
[root@localhost ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=my-system-admin
GATEWAY=128.162.244.1
```

3. Add the IP address of the house network interface and the name(s) of the admin node to `/etc/hosts` file similar to the following example:

```
# echo "128.162.244.88 my-system-admin.domain-name.mycompany.com my-system-admin" >> /etc/hosts
```

4. Set the admin node hostname, as follows:

```
# hostname my-system-admin
```

5. Configure the `/etc/resolv.conf` file with your DNS server set up. Later in the cluster set up process, these name servers will be used as the defaults for the House DNS Resolvers you configure in a later `configure-cluster` command step. Setting this now allows you to register with RHN and allows you to access your house network to access any DVD images or other settings you need. You may choose to defer this step, but then you will need to also defer `rhncfg`. Here is an example `resolv.conf`:

```
search mydomain.mycompany.com
nameserver 192.168.0.1
```

```
nameserver 192.168.0.25
```

6. Force the invalidation of the host cache of `nscd` with the `nscd(8)` command on the `hosts` file, as follows:

```
# nscd -i hosts
```

7. Restart the following services (in this order), as follows:

```
# /etc/init.d/network restart
# /etc/init.d/rpcbind start
# /etc/init.d/nfslock start
```

8. Set the local timezone. The timezone is set with `/etc/localtime`, a timezone definition file. The timezone defined in `/etc/localtime` can be determined, as follows:

```
# strings /etc/localtime | tail -1
CST6CDT,M3.2.0,M11.1.0
```

Link the appropriate timezone file from directory `/usr/share/zoneinfo` to `/etc/localtime`. For example, set timezone to Pacific Time / Los Angeles, as follows:

```
# /bin/cp -l /usr/share/zoneinfo/PST8PDT /etc/localtime.$$
# /bin/mv /etc/localtime.$$ /etc/localtime
```

Confirm the timezone, as follows:

```
# strings /etc/localtime | tail -1
PST8PDT,M3.2.0,M11.1.0
```

9. Set network time configuration.

On a RHEL6 admin node, the default `ntp` configuration file is replaced when the `configure-cluster` task **Configure Time Client/Server (NTP)** is run. Insert local `ntp` server information after running `configure-cluster` (see "Set Network Time Configuration for RHEL6 Admin Node" on page 78).

10. Make sure you have registered with the Red Hat Network (RHN). If you have not yet registered, run the following command:

```
% /usr/bin/rhn_register
```

11. Run the `configure-cluster` command. See "configure-cluster Command Cluster Configuration Tool" on page 56.

`configure-cluster` Command Cluster Configuration Tool

Note: SGI suggests that you run the `configure-cluster` command either from the VGA screen or from an `ssh` session to the admin node. Avoid running the `configure-cluster` command from a serial console.

The `configure-cluster` command launches a cluster configuration tool as shown in Figure 2-9 on page 56 on a previously configured machine. For information on initial configuration, see Procedure 2-8.

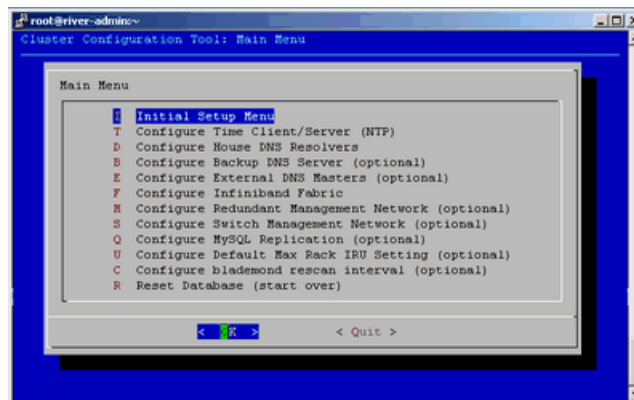


Figure 2-9 Cluster Configuration Tool: Main Menu Screen

It allows you to perform the following:

- Creates the root images for the service nodes, leader nodes, and compute blades
- Prompts for installation media including SLES11 SP1 or RHEL 6 and optionally SGI Performance Suite software. The media is used to construct repositories that are used for software installation and updates.
- Runs a set of commands that allows you to setup the cluster

- Change the subnet numbers for the various cluster networks
- Configure the subdomain of the cluster (which is likely different than the domain of `eth0` on the system admin controller itself)
- Configure the Backup DNS Server (see "Configure the Backup DNS Server" on page 94)
- Configure the InfiniBand network (see "InfiniBand Configuration" on page 96)
- Configure the Redundant Management network (see "Redundant Management Network" on page 98)
- Configure the Switch Management network (see "Switch Management Network" on page 100)
- Configure MySQL database server replication (see "MySQL Database Server Replication" on page 102)
- Configure maximum blade enclosures supported by a RLC (see "Configure Maximum IRUs Supported by a RLC" on page 104)

Information on using this tool is described in the procedure in the following section, see "Installing SMC for SGI ICE Admin Node Software" on page 42.

This section describes how to use `configure-cluster` command to configure the system administrator controller (admin node) for your SGI ICE system.

Procedure 2-8 Using the Cluster Configuration Tool to Configure Your Admin Node

To use the `configure-cluster` command to configure system admin controller (admin node), perform the following steps:

1. To start cluster configuration, enter the following command:

```
% /opt/sgi/sbin/configure-cluster
```

2. The **Cluster Configuration Tool: Initial Configuration Check** screen appears, as shown in Figure 2-10 on page 58. This tool provides instructions on the steps you need to take to configure your cluster. Click **OK** to continue.

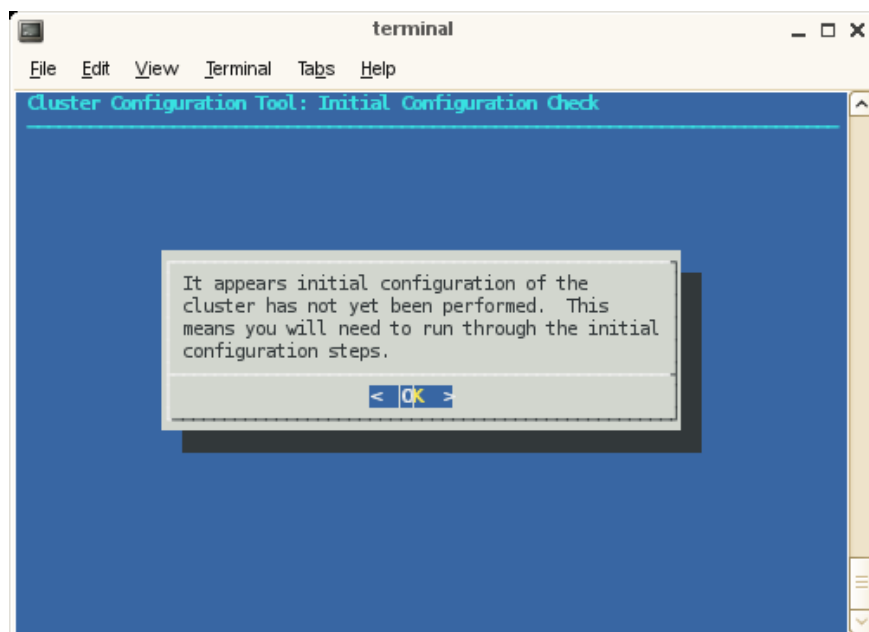


Figure 2-10 Cluster Configuration Tool: Initial Configuration Check Screen

3. The **Cluster Configuration Tool: Initial Cluster Setup** screen appears, as shown in Figure 2-11 on page 59. Read the notice and then click **OK** to continue.

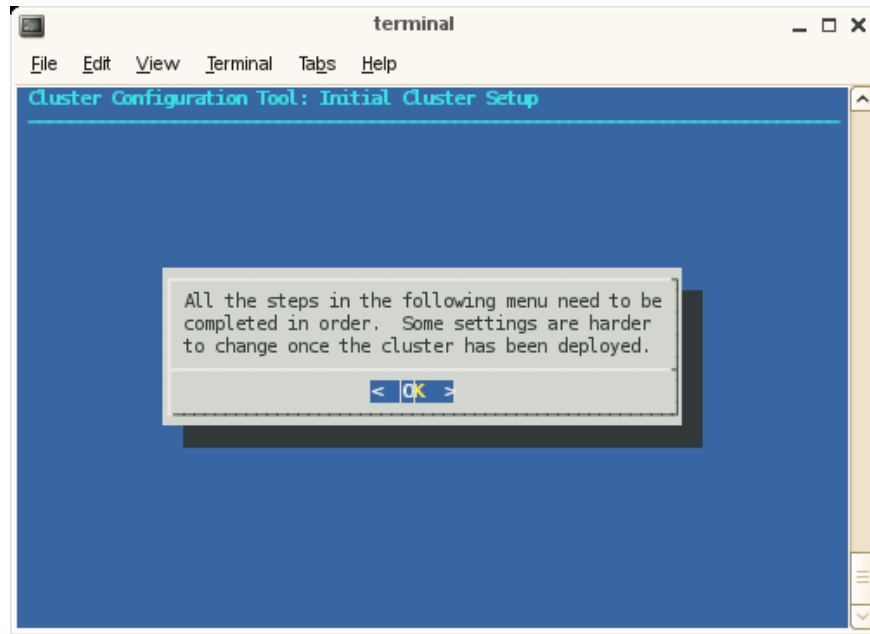


Figure 2-11 Cluster Configuration Tool: Initial Cluster Setup Screen

Note: The **Cluster Configuration Tool** has a main menu with sub-menus. You will be redirected back to the main menu, at various times, as you follow the steps in this procedure.

4. From the **Initial Cluster Setup** screen, select **Repo Manager: Set up Software Repos** and click **OK**.

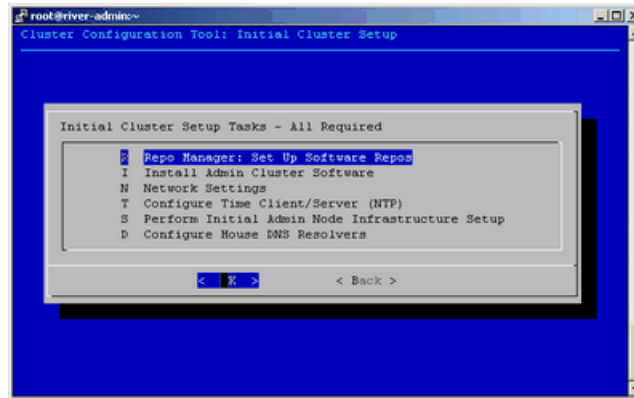


Figure 2-12 Initial Cluster Setup Tasks Screen

5.

Note: The next four screens use the `crepo` command to set up software repositories, such as, SGI Foundation, SMC for SGI ICE, SGI Performance Suite, SLES11 or RHEL 6. For more information, see "crepo Command" on page 191. If you have licensed either the `mpi` or the `accelerate` software bundles from the SGI Performance Suite, now is the time to add them.

To register ISO images from the admin node with SMC for SGI ICE and make them available to your cluster, click the **Yes** button.

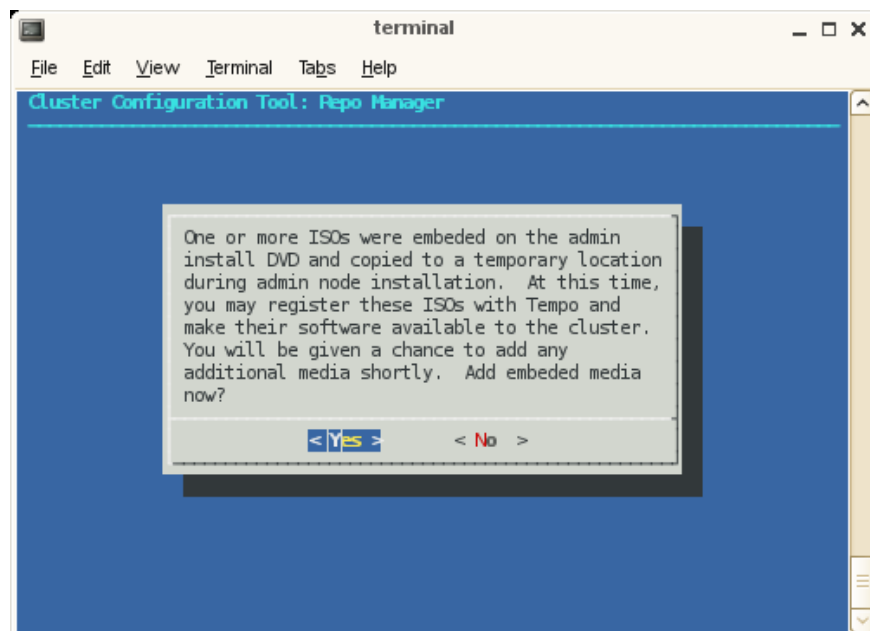


Figure 2-13 Cluster Configuration Tool: Repo Manager Screen One

6. To add the SLES media and other media, such as, SGI Performance Suite, click **OK**.

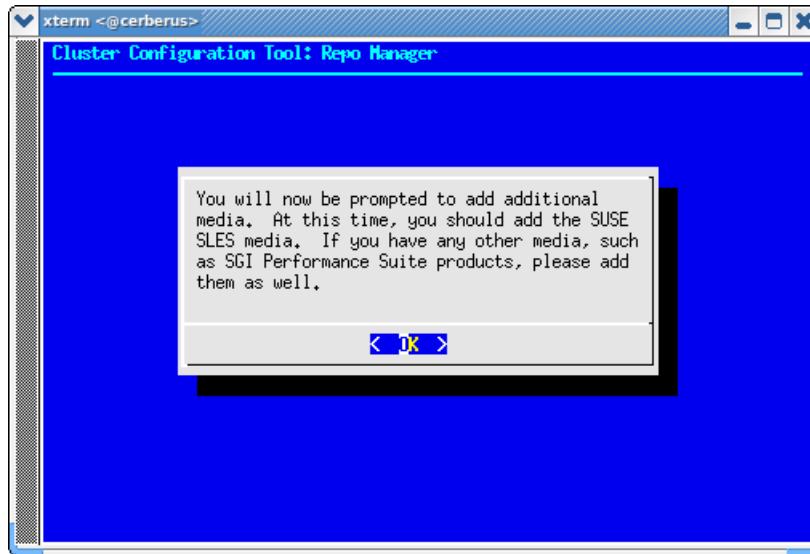


Figure 2-14 Cluster Configuration Tool: Repo Manager Screen Two

7. To register additional media with SMC for SGI ICE, click **Yes**.

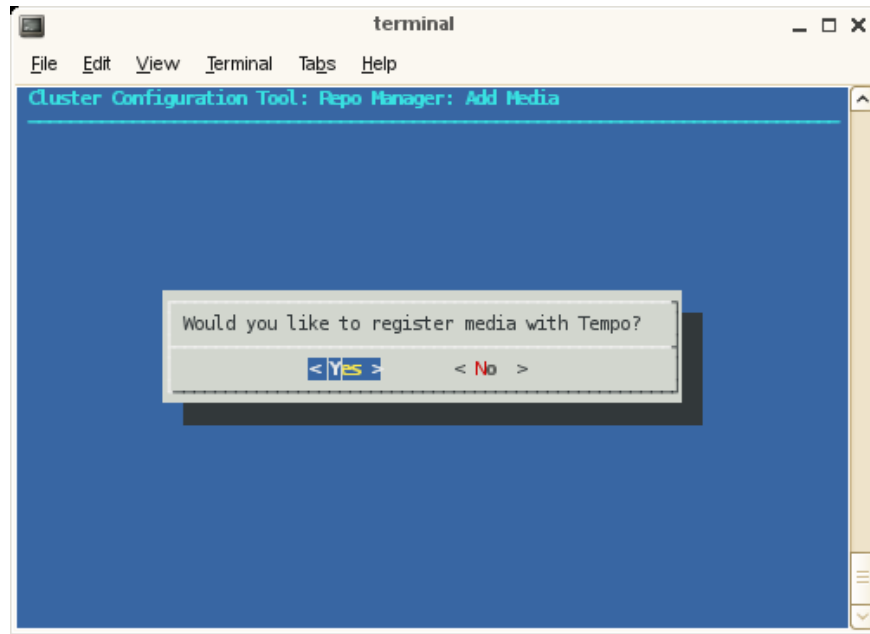


Figure 2-15 Cluster Configuration Tool: Repo Manager Screen Three

8. Enter the full path to the mount point or the ISO file or a URL or NFS path that points to an ISO file. Click **OK** to continue.

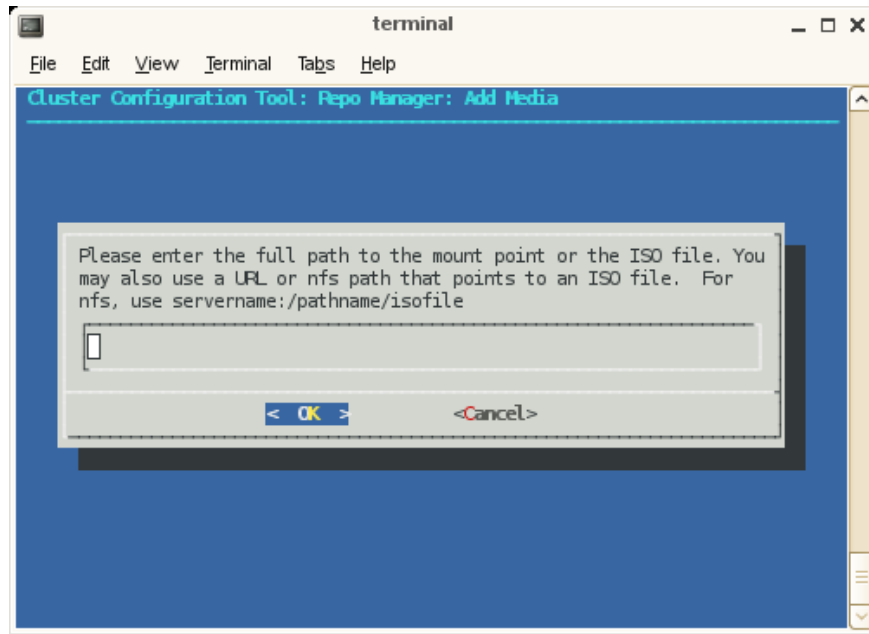


Figure 2-16 Cluster Configuration Tool: Repo Manager Screen Four

9. From the **Repo Manager: Add Media** screen, click **OK** to continue and eject you DVD if you used physical media.

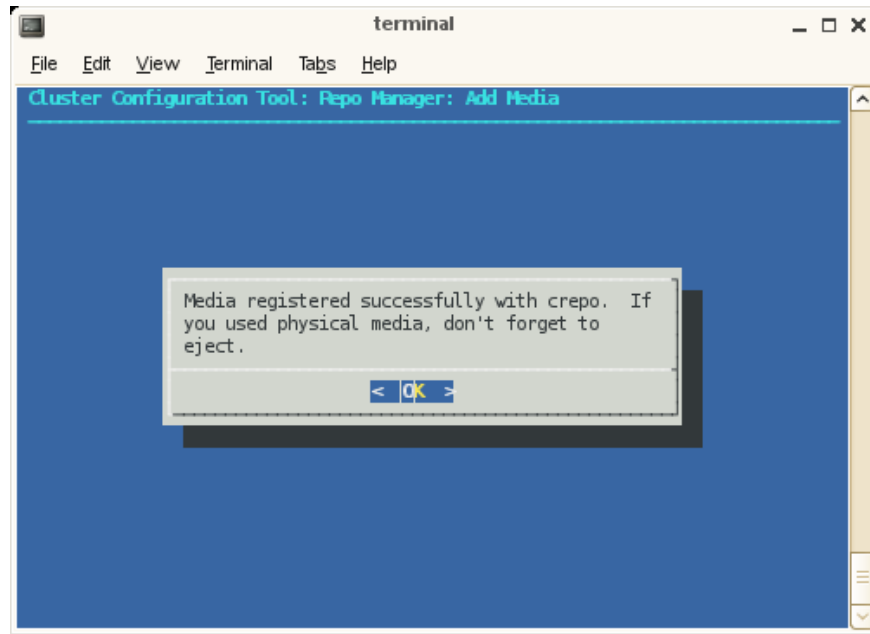


Figure 2-17 Cluster Configuration Tool: Repo Manager: Add Media Screen Four

Note: You will continue to be prompted to add additional media until you answer no. Once you answer no, you are directed back to the **Initial Cluster Setup Tasks** menu.

10. After choosing the **Network Settings** option, the **Cluster Network Setup** screen appears, as shown in Figure 2-18 on page 66.

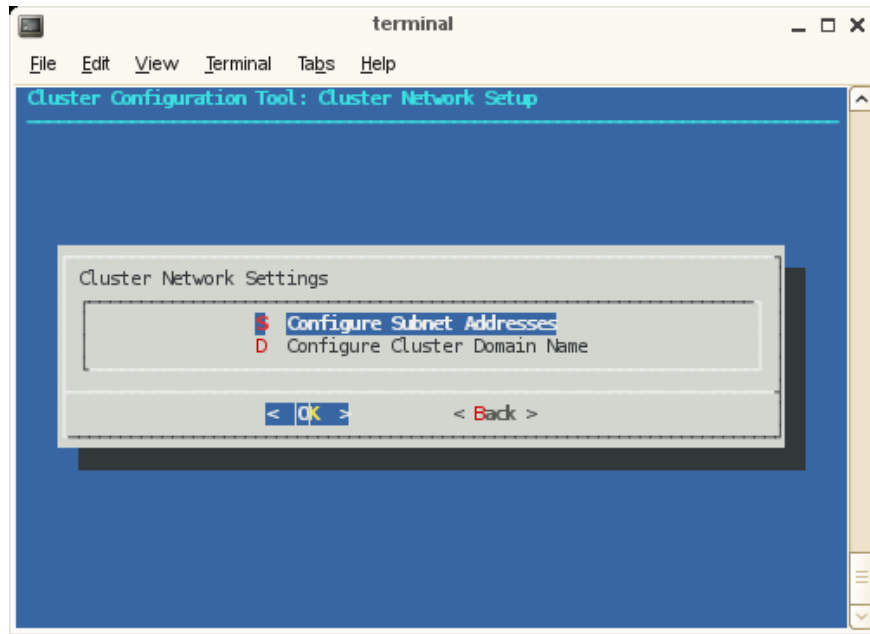


Figure 2-18 Cluster Network Setup Screen

The subnet addresses allows you to change the cluster internal network addresses. SGI recommends that you do NOT change these. Click **OK** to continue to adjust subnets. Otherwise, select **Domain Name: Configure Cluster Domain Name** and then skip to step 31. A warning screen appears, as shown in Figure 2-19 on page 67.

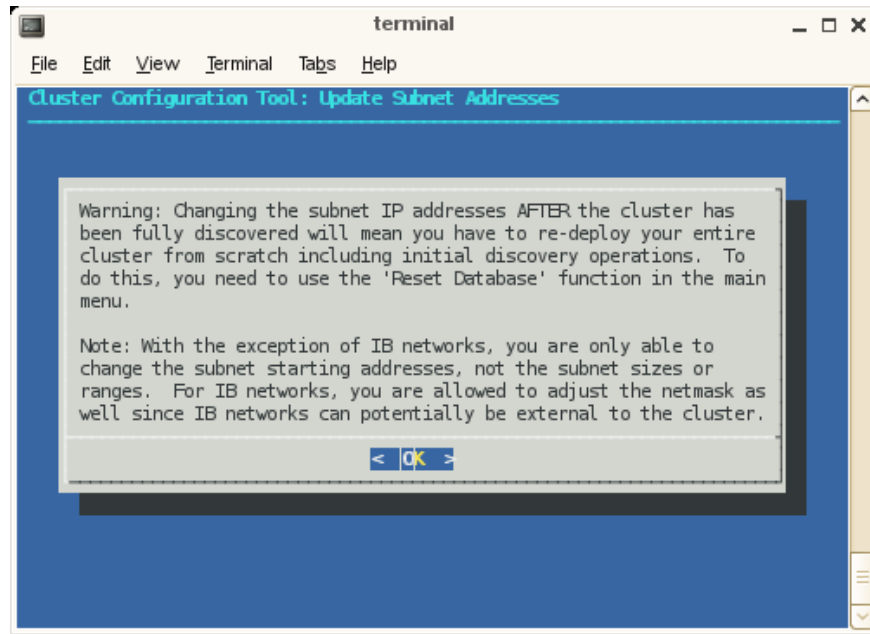


Figure 2-19 Update Subnet Address Warning Screen

Once you deploy your SGI ICE system, to change the network IP values or change domain names, you must reset the system data base and then rediscover the system. You do not need to reinstall the admin node, however. Click **OK** to continue.

11. The **Update Subnet Addresses** screen appears, as shown in Figure 2-20 on page 68.

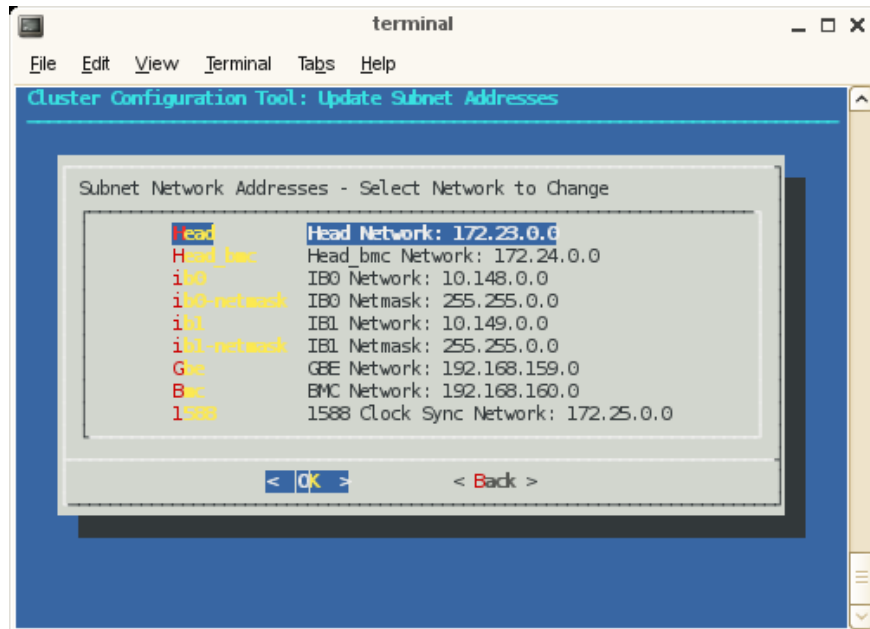


Figure 2-20 Update Subnet Addresses Screen

The default IP address of the system admin controller which is the **Head Network** for the SGI ICE system is shown. SGI recommends that you do NOT change the IP address of the system admin controller (admin node) or rack leader controllers (leader nodes) if at all possible. You can adjust the IP addresses of the InfiniBand network (**ib0** and **ib1**) to match the IP requirements of the house network. Click **OK** to continue.

12. Enter the domain name for your SGI ICE system, as shown in Figure 2-21 on page 69. Click **OK** to continue (this will be a subdomain to your house network, by default).

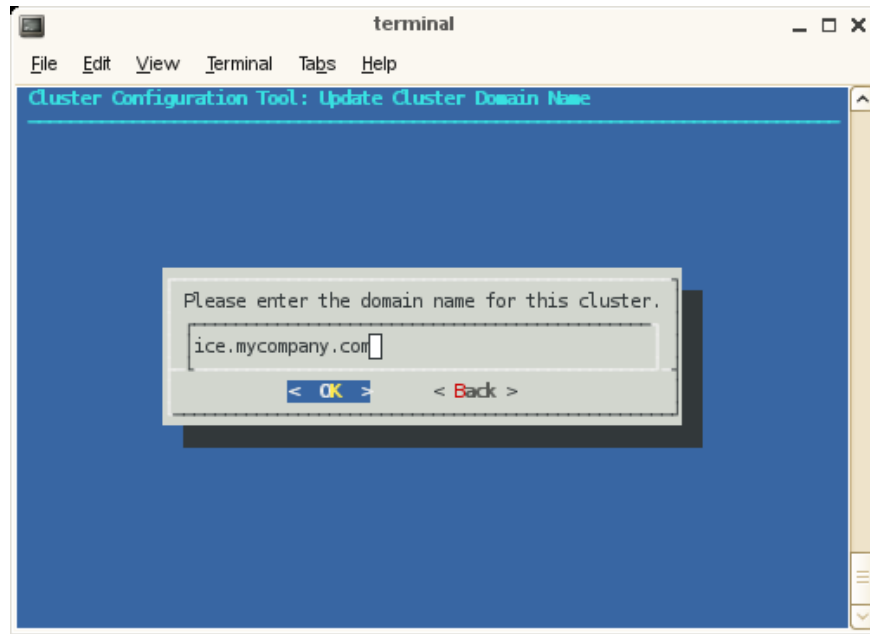


Figure 2-21 Update Cluster Domain Name Screen

13. The configure-cluster NTP configuration screens with for RHEL 6 are different from those for SLES. The next operation in the **Initial Cluster Setup** menu is **NTP Time Server/Client Setup** is for SLES. This procedure changes your NTP configuration file. Click on **OK** to continue. This sets the system admin controller to serve time to the SGI ICE system and allows you to add time servers on your house networks, which you may optionally use.

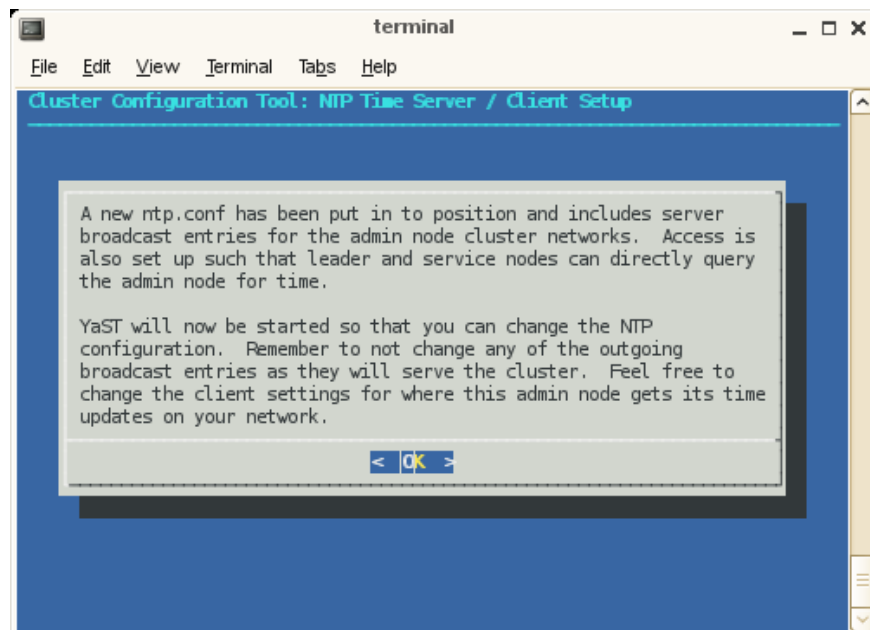


Figure 2-22 NTP Time Server/Client Setup Screen

14. Configure NTP time service as shown in Figure 2-23 on page 71. Click **Next** to continue.

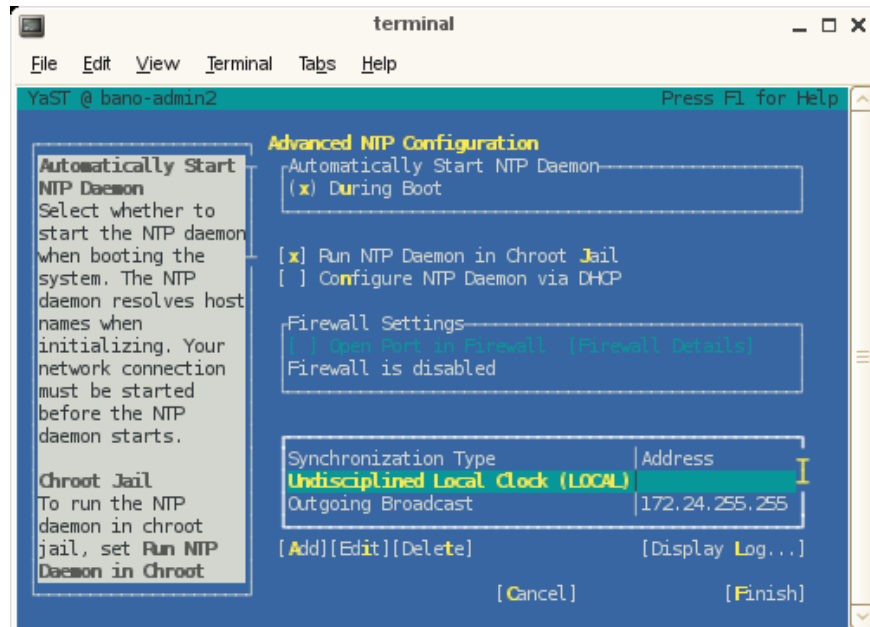


Figure 2-23 Advanced NTP Configuration Screen

15. From the **New Synchronization** screen, select a synchronization peer and click **Next** to continue.

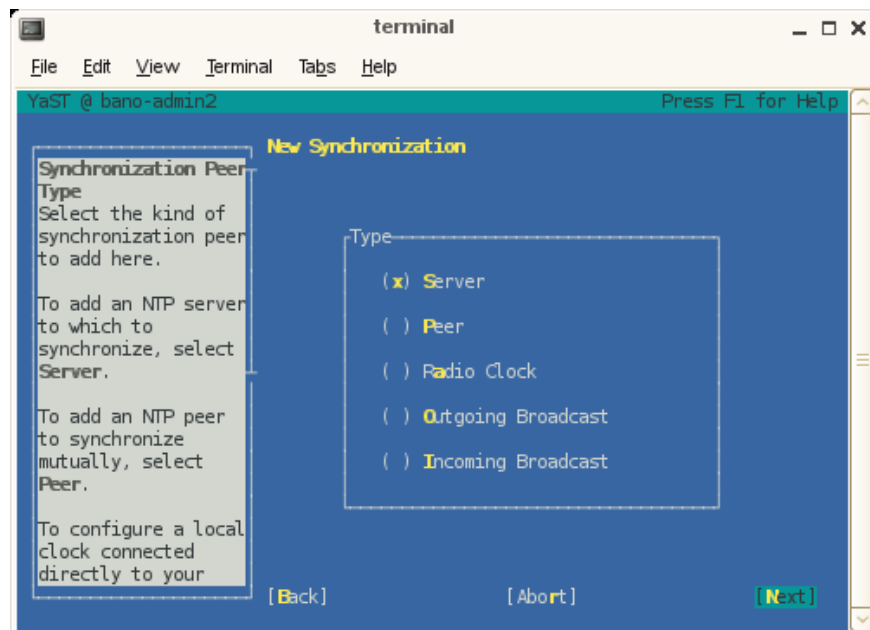


Figure 2-24 New Synchronization Screen

16. From the NTP Server screen, set the address of the NTP server and click OK to continue.

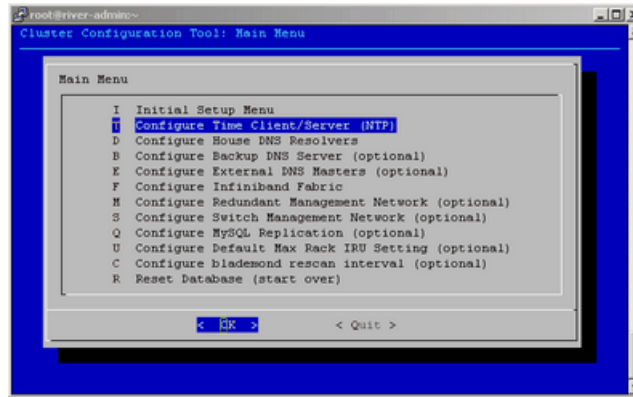


Figure 2-25 NTP Server Screen

17. The YaST tool completes. Click **OK** to continue.

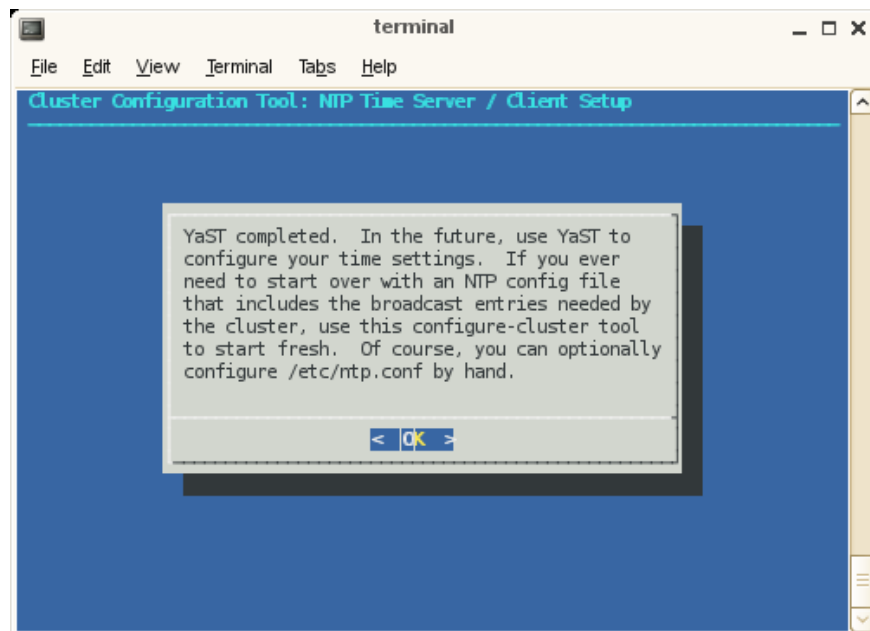


Figure 2-26 NTP Time Server/ Client Setup Screen Three

On a RHEL 6 installation, from the menu screen in which "set ntp servers" is selected, selecting setting ntp servers just displays Figure 2-27 on page 74 and then returns you to the previous menu.

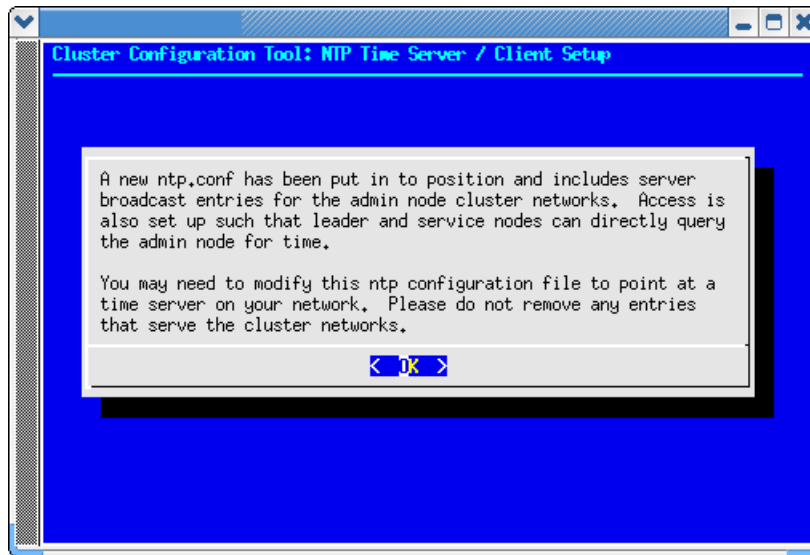


Figure 2-27 Cluster Configuration Tool: NTP Time Server / Client Setup Screen (RHEL)

18. The next step in the **Initial Cluster Setup** menu directs you to select **Perform Initial Admin Node Infrastructure Setup**. This step runs a series of scripts that will configure the system admin controller of the SGI ICE system.

The script installs and configures your system and you should see an **install-cluster completed** line in the output.

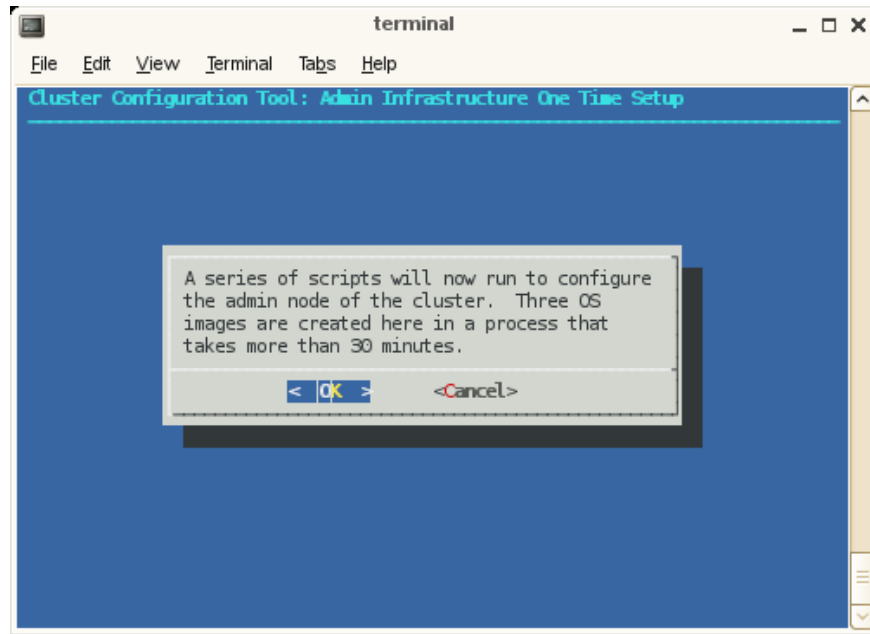


Figure 2-28 Admin Infrastructure One Time Setup Screen One

The root images for the service, rack leader controller, and compute nodes are then created. The output of the `mkssiimage` commands are stored in a log file at the following location:

```
/var/log/cinstallman
```

You can review the output if you so choose.

The final output of the script reads, as follows:

```
/opt/sgi/sbin/create-default-sgi-images Done!
```

Note: As it notes on the **Admin Infrastructure One Time Setup** screen, this step takes about 30 minutes.

Click **OK** to continue.

19. The next step in the **Initial Cluster Setup** menu is to configure the house DNS resolvers. It is OK to set these resolvers to the same name servers used on the system admin controller itself. Configuring these servers is what allows service nodes to resolve host names on your network. For a description of how to set up service nodes, see "Service Node Discovery, Installation, and Configuration" on page 92. This menu has default values printed that match your current admin node resolver setup. If this is ok, just select **OK** to continue. Otherwise, make any changes you wish to the resolver listing and select **OK**. If you do not wish to have any house resolvers, select **Disable House DNS**.

After entering the IPs, click **OK** to enable, click **Disable House DNS** to stop using house DNS resolution, click **Back** to leave house DNS resolution as it was when you started (disabled at installation).

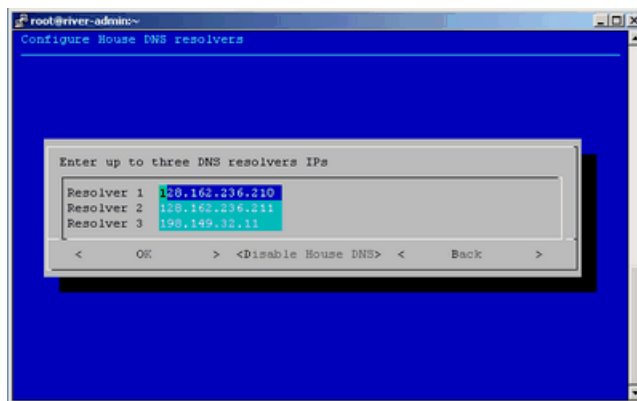


Figure 2-29 Configure House DNS Resolvers Screen

20. The setting DNS forwarding screen appears. Click **Yes** to continue.

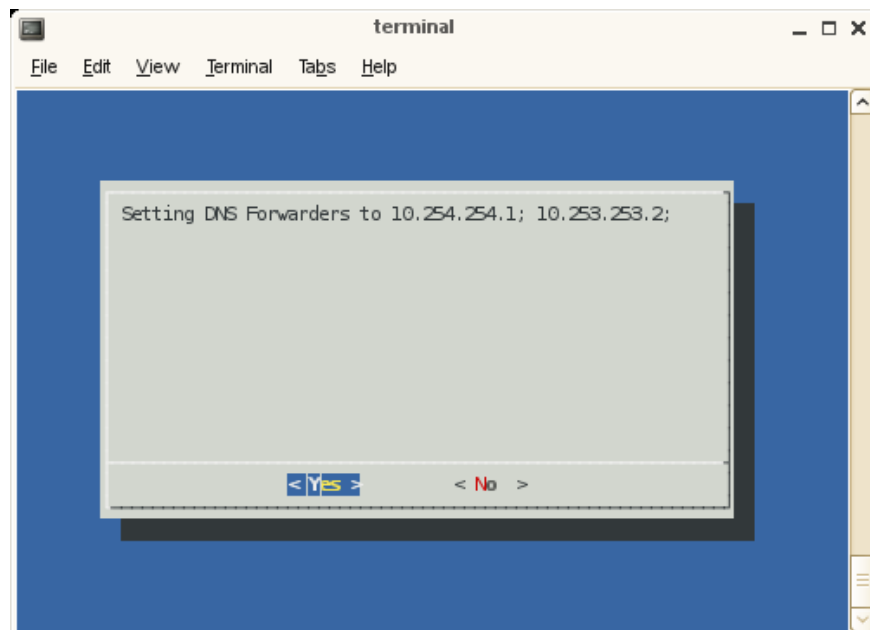


Figure 2-30 Setting DNS Forwarding Screen

21. The **Initial Cluster Setup complete message** appears. Click OK to continue.

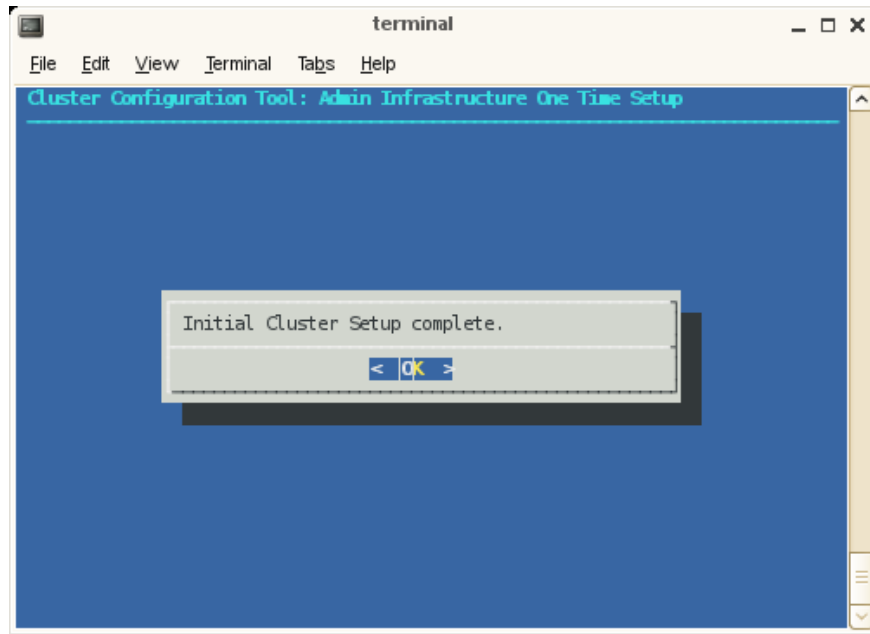


Figure 2-31 Cluster Configuration Tool: Admin Infrastructure One Time Setup Screen

22. Proceed to "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86. It describes the discovery process for the rack leader controllers in your system and how to install software on the rack leader controllers.

Note: The main menu contains a **reset** the database function that allows you to start software installation over without having to reinstall the system admin controller.

Set Network Time Configuration for RHEL6 Admin Node

After finishing with the `configure-cluster` command, you probably will want to modify the `ntp` configuration to point at a time server on your network. Please do not remove entries that serve the cluster networks.

By default, the configuration in `/etc/ntp.conf` directs requests to public servers of the `pool.ntp.org` project.

```
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
```

To direct requests to the your company ntp server, for example `ntp.mycompany.com`, comment or delete the `rhel.pool.ntp.org` entries and insert the local entry, as follows:

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org
#server 1.rhel.pool.ntp.org
#server 2.rhel.pool.ntp.org
server ntp.mycompany.com
```

Restart the ntp server, as follows:

```
# /etc/init.d/ntpd restart
```

cmcdetectd Daemon

The `cmcdetectd` daemon runs on the admin node. When it sees a chassis management controller (CMC) asking for an IP address, it looks at the client ID of the request. That client ID contains the rack number and slot number. The `cmcdetectd` daemon provides this information to the `switchConfig` application programming interface (API) and configures the top level switch fabric.

The `cmcdetectd` daemon performs the following:

- The `cmcdetectd` daemon really only starts working when at least one management switch is discovered.
- It configures the switch and "moves" the CMCs to the appropriate VLAN.
- If there are two switches in the switch stack, `cmcdetectd` through the `switchConfig` API configures the CMC ports for manual trunking (the CMC-0 and CMC-1 ports on the physical CMC).

- Once moved, the dynamic host configuration protocol (DHCP) requests are directed to the rack VLAN for a given rack and detected by the rack leader (when it is discovered).
- If the `cmcdetectd` daemon detects any SGI ICE X CMCs, it automatically sets to true the switch management network variable to true (see "Switch Management Network" on page 100).
- If you install a second slot or re-install a system, the switch is already configured and the `cmcdetectd` daemon does not see the requests any more. It is a good practice to manually configure the switch management network setting using the `configure-cluster` option.

discover Command

The `discover` command is used to discover rack leader controllers (leader nodes), service nodes, including their associated BMC controllers, in an entire system or in a set of one or more racks that you select. Rack numbers generally start at one. Service nodes generally start at zero. When you use the `discover` command to perform the discovery operation on your SGI ICE system, you will be prompted with instructions on how to proceed (see "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86).

The operation of the `discover` command `--delrack` and `--delservice` options has changed. Now when using these options, the node is not removed completely from the database but it is marked with the administrative status `NOT_EXIST`. When you go to discover a node that previously existed, you now get the same IP allocations you had previously and the node is then marked with the administrative status of `ONLINE`. If you have a service node, for example, `service0`, that has a custom host name of "myhost" and you later go to delete `service0` using the `discover --delservice` command, the host name associated with it will still be present. This can cause conflicts if you wish to reuse the custom host name "myhost" on a node other than `service0` in the future. You can use the `cadmind --db-purge --node service0` command that will remove the node entirely from the database (for more information, see "cadmind: SMC for SGI ICE Administrative Interface" on page 232). You can then reuse the "myhost" name.

There is a new hardware typed named `generic`. This hardware type has its MAC address discovered, but it is for devices that only have a single MAC address and do not need to be managed by SMC for SGI ICE software. The likely usage scenario is

Ethernet switches that extend the management network that are necessary in large SGI ICE 8400 configurations.

When the `generic` hardware type is used for external management switches on large SGI ICE 8400 systems, the following guidelines should be followed:

- The management switches should be the first things discovered in the system
- The management switches should both start with their power cords unplugged (analogous to how SMC for SGI ICE discovers rack leader nodes and service nodes)
- The external switches can be given higher numbered service numbered if your site does not want them to take lower numbers.
- You can also elect to give these switches an alternate host name using the `cadmin` command after discovery is complete.
- Examples of using the `discover` command `generic` hardware type are, as follows:

```
admin:~ # discover --service 98,generic
admin:~ # discover --service 99,generic
```

Note: When you use the `discover` command to discover an SGI XE500 service node, you **must** specify the hardware type. Otherwise, the serial console will not be set up properly. Use a command similar to the following:

```
admin:~ # discover --service 1,xe500
```

For a `discover` command usage statement, perform the following:

```
[sys-admin ~]# discover --h
Usage: discover [OPTION]...
Discover lead nodes, service nodes, and external switches.
```

Options:

<code>--delrack NUM[,FLAG]...</code>	mark rack leaders as deleted
<code>--delservice NUM</code>	mark a service node as deleted
<code>--delibswitch NUM</code>	mark an external ib switch as deleted
<code>--delmgmtswitch NUM</code>	mark a mgmt network switch as deleted
<code>--force</code>	avoid sanity checks that require input
<code>--ignoremac MAC</code>	ignore the specified MAC address

2: System Discovery, Installation, and Configuration

```
--macfile FILE                read mac addresses from FILE
--rack NUM[,FLAG]...          discover a specific rack or set of racks
--rackset NUM,COUNT[,FLAG]... discover count racks starting at #
--service NUM[,FLAG]...       discover the specified service node
--ibswitch NUM[,FLAG]...       discover the specified external ib switch
--mgmtswitch NUM[,FLAG]...     discover the specified mgmt switch
--show-macfile                print output usable for --macfile to stdout
```

Details:

Any number of management switches, racks, service nodes, or external switches can be discovered in one command line. Rack numbers generally start at 1, service nodes, management switches, and infiniband switches generally start at 0. An existing node can be re-discovered by re-running the discover command. An easier way to simply re-image a node is by using the `cinstallman` command, see the `--next-boot` and `--assign-image` options.

A comma searated set of optional FLAGS modify how discover proceeds for the associated node and sets it up for installation. FLAGS can be used to specify hardware type, image, console device, etc.

The 'generic' hardware type is for hardware that should be discovered but that only has one IP address associated with it. Tempo will treat this hardware as an unmanaged service node. An example use would be for the administrative interface of an ethernet switch being used for the Tempo management network. When this type is used, the generic hardware being discovered should be doing a DHCP request.

The 'other' hardware type should be used for a service node which is not managed by Tempo. This mode will allocate IPs for you and print them to the screen. Since Tempo only prints IP addresses to the screen in this mode, the device being discovered does not even need to exist at the moment the operation is performed.

The `--macfile` option can be used instead of discovering MACs by power cycling. All MACs to be discovered must be in the file. External switches should simply repeat the same MAC twice in this file. File format:

Example file contents:

```
r1lead 00:11:22:33:44:55 66:77:88:99:EE:FF
service0 00:00:00:00:00:0A 00:00:00:00:00:0B
extsw1 00:00:00:00:00:11 00:00:00:00:00:11
```

Hardware Type Flags:

```
altix4000 altix450 altix4700 default generic h2106-g7 ice-csn iss3500-intel
kvm other uv10 xe210 xe240 xe250 xe270 xe310 xe320 xe340 xe500
```

Switch Type Flags:

```
voltaire-isr-9288 voltaire-isr-9096 voltaire-isr-9024 voltaire-isr-2004
voltaire-isr-2012 voltaire4036 mellanox5030 mellanox5600
```

Other Flags:

```
image=IMAGE          specify an alternate image to install
console_device=DEVICE use DEVICE for console
net=NET              ib0 or ib1, for external IB switches only
type=TYPE            leaf or spine, for external IB switches only
redundant_mgmt_network=YESNO yes or no, determines how network is configured
switch_mgmt_network=YESNO no if node is in an ICE8200/ICE8400 system
mgmt_bonding=TYPE    type of bonding to use: active-backup or 802.3ad
ha=all               High Availability solution for the rack (HA-RLC)
ha=1                 the command applies for the HA-RLC #1
ha=2                 the command applies for the HA-RLC #2
only_bmc=YESNO      yes: only BMC discovered (but all IPs allocated)
bt=YESNO             yes: use bittorrent while imaging, default no
```

Examples:

```
Discover a top level management switch
```

```
# discover --mgmtswitch 0
```

You can later use the 'cadmin' command to give it a custom hostname if you so choose.

```
Discover rack 1 and service node 0:
```

```
# discover --rack 1 --service 0
```

```
Discover service 0, using myimage and disabling redundnat_mgmt_network.
```

```
# discover --service 0,image=myimage,redundant_mgmt_network=no
```

```
Discover racks 1 and 4, service node 1, ignores MAC address 00:04:23:d6:03:1c:
```

```
# discover --ignoremac 00:04:23:d6:03:1c --rack 1 --rack 4 --service 1
```

```
Discover racks 1-5, service node 0-2, where service node 1 is Altix 450
hardware and service node 2 is "other":
```

```
# discover --rackset 1,5 --service 0,xe240 --service 1,altix450 --service 2,other
```

Discover an external ib switch, corresponding to the voltaire-isr-9024 hardware and IB0 fabric.

```
# discover --ibswitch 0,voltaire-isr-9024,net=ib0,type=spine
```

You can later use the 'cadmin' command to give it a custom hostname if you so choose.

Discover a switch used to extend the Tempo management network - a generic device.

```
# discover --service 99,generic
```

Discover two leaders for rack 1 (High Availability):

```
# discover --rack 1,ha=all
```

Discover r1lead1 (High Availability):

```
# discover --rack 1,ha=1
```

Discover r1lead2 (High Availability):

```
# discover --rack 1,ha=2
```

Discover two leaders per rack for racks 1, 2, and 3 (High Availability):

```
# discover --rackset 1,3,ha=all
```

Delete r1lead1 (High Availability):

```
# discover --delrack 1,ha=1
```

Delete r1lead2 (High Availability):

```
# discover --delrack 1,ha=2
```

EXAMPLES

Example 2-1 discover Command Examples

The following examples walk you through some typical discover command operations.

To discover rack 1 and service node 0, perform the following:

```
admin:~ # /opt/sgi/sbin/discover --rack 1 --service0,xe210
```

In this example, service node 0 is an XE210 system.

To discover racks 1-5, and service node 0-2, perform the following:

```
admin:~ # /opt/sgi/sbin/discover --rackset 1,5 --service0,xe240 --service 1,altix450 --service 2,other
```

In this example, service node 1 is an Altix 450 system. Service node 2 is *other* hardware type.

To discover service 0, but use `service-myimage` instead of `service-sles11` (default), perform the following:

```
admin:~ # /opt/sgi/sbin/discover --service0,image=service-myimage
```

Note: You may direct a service node to image itself with a custom image later, without re-discovering it. See "cinstallman Command" on page 195.

To discover racks 1 and 4, service node 1, and ignore MAC address 00:04:23:d6:03:1c, perform the following:

```
admin:~ # /opt/sgi/sbin/discover --ignoremac 00:04:23:d6:03:1c --rack 1 --rack 4 --service0
```

The `discover` command supports external switches in a manner similar to racks and service nodes, except that switches do not have BMCs and there is no software to install. The syntax to add a switch is, as follows:

```
admin:~ # discover --ibswitch name,hardware,net=fabric,type=spine
```

where *name* can be any alphanumeric string, *hardware* is any one of the supported switch types (run `discover --help` to get a list), and *net=fabric* is either *ib0* or *ib1*, and *type=* is *leaf* or *spine*, for external IB switches only.

An example command is, as follows:

```
# discover --ibswitch extsw,voltaire-isr-9024,net=ib0,type=spine
```

Once `discover` has assigned an IP address to the switch, it will call the fabric management `sgifmcli` command to initialize it with the information provided. The `/etc/hosts` and `/etc/dhcpd.conf` files should also have entries for the switch as named, above. You can use the `cnodes --ibswitch` command to list all such nodes in the cluster.

To remove a switch, perform the following:

```
admin:~ # discover --delibswitch name
```

where *name* is that of a previously discovered switch.

An example command is, as follows:

```
admin:~ # discover --delibswitch extsw
```

When your are discovering a node, you can use an additional option to turn on or off the redundant management network for that node. For example:

```
admin:~ # discover --service0,xs500,redundant_mgmt_network=no
```

Discover a switch used to extend the SMC for SGI ICE management network, a generic device, as follows:

```
admin:~ # discover --service 99,generic
```

Installing Software on the Rack Leader Controllers and Service Nodes

The `discover` command, described in "discover Command" on page 80, sets up the leader and managed service nodes for installation and discovery. This section describes the discovery process you use to determine the Media Access Control (MAC) address, that is, the unique hardware address, of each rack leader controller (leader nodes) and then how to install software on the rack leader controllers.

Note: When leader and service nodes come up and are configured to install themselves, they determine which Ethernet devices are the integrated ones by only accepting DHCP leases from SMC for SGI ICE. They then know that the interface they got a lease from must be an integrated Ethernet device. This is facilitated by using a DHCP option code. SMC for SGI ICE uses option code 149 by default. In rare situations, a house network DHCP server could be configured to use this option code. In that case, nodes that are connected to the house network could misinterpret a house DHCP server as being a SMC for SGI ICE one and auto detect the interface incorrectly. This would lead to an installation failure.

To change the `dhcp` option code number used for this operation, see the `cadmin --set-dhcp-option` option. The `--show-dhcp-option` will show the current value. For more information on the using the `cadmin` command, see "cadmin: SMC for SGI ICE Administrative Interface" on page 232.

Procedure 2-9 Installing Software on the Rack Leader Controllers and Service Nodes

To install software on the rack leader controllers, perform the following steps:

1. Use the `discover` command from the command line, as follows:

```
# /opt/sgi/sbin/discover --rack 1
```

Note: You can discover multiple racks at a time using the `--rackset` option. Service nodes can be discovered with the `--service` option.

The `discover` script executes. When prompted, turn the power on to the node being discovered and only that node.

Note: Make sure you only power on the node being discovered and nothing else in the system. Make sure not to power the system up itself.

When the node has electrical power, the BMC starts up even though the system is not powered on. The BMC does a network DHCP request that the `discover` script intercepts and then configures the cluster database and DHCP with the MAC address for the BMC. The BMC then retrieves its IP address. Next, this script instructs the BMC to power up the node. The node performs a DHCP request that the script intercepts and then configures the cluster database and DHCP with the MAC address for the node. The rack leader controller installs itself using the `systemimager` software and then boots itself.

The `discover` script will turn on the chassis identify light for 2 minutes. Output similar to the following appears on the console:

```
Discover of rack1 / leader node r1lead complete
r1lead has been set up to install itself using systemimager
The chassis identify light has been turned on for 2 minutes
```

2. The blue chassis identify light is your cue to power on the next rack leader controller and start the process all over.

You may watch install progress by using the `console` command. For example, `console r1lead` connects you to the console of the `r1lead` so that you can watch installation progress. The sessions are also logged. For more information on the `console` command, see "Console Management" on page 238.

3. Using the identify light, you can configure all the rack leader controllers and service nodes in the cluster without having to go back and fourth to and from

your workstation between each discovery operation. Just use the identify light on the node that was just discovered as your cue to move to the next node to plug in.

4. Shortly after the `discover` command reports that discovery is complete for a given node, that node installs itself. If you supplied multiple nodes on the `discover` command line, it is possible multiple nodes could be in different stages of the imaging/installation process at the same time. For rack leaders, when the leader boots up for the first time, one process it starts is the `blademon` process. This process discovers the IRUs and attached blades and sets them up for use. The `blademon` process is described in "blademon Command For Automatic Blade Discovery" on page 89, including which files to watch for progress.

If your `discover` process does **not** find the appropriate BMC after a few minutes, the following message appears:

```
=====
Warning: Trouble discovering the BMC!
=====
3 minutes have passed and we still can't find the BMC we're looking for.
We're going to keep looking until/if you hit ctrl-c.
```

Here are some ideas for what might cause this:

- Ensure the system is really plugged in and is connected to the network.
- This can happen if you start `discover` AFTER plugging in the system. Discover works by watching for the DHCP request that the BMC on the system makes when power is applied. Only nodes that have already been discovered should be plugged in. You should only plug in service and leader nodes when instructed.
- Ensure the CMC is operational and passing network traffic.
- Ensure the CMC firmware is up to date and that it's configured to do VLANs.
- Ensure the BMC is properly configured to use `dhcp` when plugged in to power.
- Ensure the BMC, `frusdr`, and bios firmware is up to date on the node.
- Ensure the node is connected to the correct CMC port.

Still Waiting. Hit `ctrl-c` to abort this process. That will abort discovery at this problem point -- previously discovered components will not be affected.

```
=====
```

If your discover process finds the appropriate BMC, but cannot find the leader or service node that is powered up after a few minutes, the following message appears:

```
=====
Warning: Trouble discovering the NODE!
=====
4 minutes have passed and we still can't find the node.
We're going to keep looking until/if you hit ctrl-c.
```

If you got this far, it means we did detect the BMC earlier, but we never saw the node itself perform a DHCP request.

Here are some ideas for what might cause this:

- Ensure the BIOS boot order is configured to boot from the network first
- Ensure the BIOS / frusdr / bmc firmware are up to date.
- Is the node failing to power up properly? (possible hardware problem?)
Consider manually pressing the front-panel power button on this node just in case the ipmitool command this script issued failed.
- Try connecting a vga screen/keyboard to the node to see where it's at.
- Is there a fault on the node? Record the error state of the 4 LEDs on the back and contact SGI support. Consider moving to the next rack in the mean time, skipnig this rack (hit ctrl-c and re-run discover for the other racks and service nodes).

```
Still Waiting. Hit ctrl-c to abort this process. That will abort discovery
at this problem point -- previously discovered components will not be affected.
=====
```

5. You are now ready to discover and install software on the compute blades in the rack. For instructions, see "Discovering Compute Nodes" on page 91.

blademond Command For Automatic Blade Discovery

You no longer need to explicitly call the `discover-rack` command to discover a rack and integrate new blades. This is done automatically by a the `blademond` daemon that runs on the leader nodes.

The `blademond` daemon is started up when the leader node boots after imaging and begins to poll the chassis management control (CMC) blade in each IRU to determine

if any new blades are present. It polls the CMCs every two minutes to see if anything has changed. If something has changed (a new blade, a blade removed, or a blade swapped), it sends the new slot map to the admin node and calls the `discover-rack` command to integrate the changes. It then boots new nodes on the default compute image.

The `blademond` daemon maintains its log file at `/var/log/blademond` on the leader nodes.

You can turn on debug mode in the `blademond` daemon by sending it a `SIGUSR1` signal from the leader node, as follows:

```
# kill -USR1 pid
```

To turn debug mode off, send it another `SIGUSR1` signal. You should see a message in the `blademond` log about debug mode being enabled or disabled.

The `blademond` daemon maintains the slot map at `/var/opt/sgi/lib/blademond/slot_map` on the leader nodes. This appears as `/var/opt/sgi/lib/blademond/slot_map.rack_number` on the admin node.

You can configure the `blademond` daemon rescan interval from the `configure-cluster` GUI, as shown in Figure 2-32 on page 90.

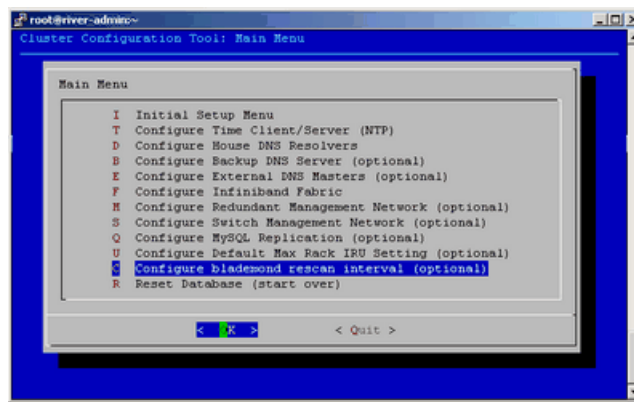


Figure 2-32 Configure `blademond` rescan interval Option Screen

From the **Configure blademond Rescan Interval** screen, set the default number of seconds to wait between attempts to look for blade changes, as shown in Figure 2-33 on page 91.

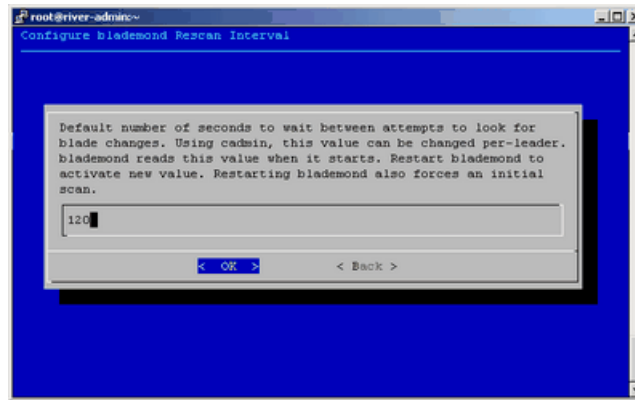


Figure 2-33 Configure blademond Rescan Interval Screen

For a blademond `--help` statement, ssh onto the r1lead leader node, as follows:

```
[root@admin ~]# ssh r1lead
Last login: Tue Jan 17 13:21:34 2012 from admin
[root@r1lead ~]#
[root@r1lead ~]# /opt/sgi/lib/blademond --help
Usage: blademond [OPTION] ...
```

Discover CMCs and blades managed by CMCs.

Note: This daemon normally takes no arguments.

```
--help      Print this usage and exit.
--debug     Enable debug mode (also can be enabled by setting CM_DEBUG)
--fakecmc   Development only: Discover fake CMCs instead of real ones
--scan-once Initialize, scan for blades, set blades up. Do not daemonize.
            Do not keep looping - do one pass and exit.
```

Discovering Compute Nodes

This section describes how to discover compute nodes in your SGI ICE system.

Note: You no longer need to explicitly call the `discover-rack` command to discover a rack and integrate new compute nodes (blades). This is done automatically by the `blademon` daemon that runs on the leader nodes (see "blademon Command For Automatic Blade Discovery" on page 89).

Procedure 2-10 Discovering Compute Nodes

To discover compute nodes (blades) in your SGI ICE system, perform the following:

1. Complete the steps in "Installing Software on the Rack Leader Controllers and Service Nodes" on page 86.
2. For instructions on how to configure, start, verify, or stop the InfiniBand Fabric management software on your SGI ICE system, see Chapter 4, "System Fabric Management" on page 265.

Note: The InfiniBand fabric does not automatically configure itself. For information on how to configure and start up the InfiniBand fabric, see Chapter 4, "System Fabric Management" on page 265.

Service Node Discovery, Installation, and Configuration

Service nodes are discovered and deployed similar to rack leader controllers (leader nodes). The `discover` command, with the `--service` related commands, allow you to discover service nodes in the same discover operation that discovered the leader nodes.

Like rack leader controllers, the service node is automatically installed. The service node image associated with the given service node is used for installation.

Service nodes have one (or possibly two, see "Redundant Management Network" on page 98) Ethernet connection(s) to the SGI ICE network. Service nodes may also be connected to your house network. Typically, interfaces with lower numbers are connected to the SGI ICE network (for example, `eth0`, or `eth0` and `eth1`), and any remaining network interfaces are used to connect to the house network.

The firstboot system setup script does not start automatically on the system console after the first boot after installation (unlike the admin node).

Use YAST to set up the public/house network connection on the service node, as follows:

- Select the interface which is connected to your house network to configure in firstboot (for example, eth1 or eth2).
- If you change the default host name, you need to make sure that the cluster service name is still resolvable as tools depend on that.
- Name service configuration is handled by the admin and leader nodes. Therefore, service node `resolv.conf` files need to always point to the admin and leader nodes in order to resolve cluster names. If you wish to resolve host names on your "house" network, use the `configure-cluster` command to configure the house name servers. The admin and leader nodes will then be able to resolve your house network addresses, in addition to the internal cluster hostnames. Besides, the cluster configuration update framework may replace your `resolv.conf` file anyway when cluster configuration adjustments are made.

Do not change `resolv.conf` and do not configure different name servers in `yast`.

In some rare cases, it is possible that your house networks uses the the same DHCP option identifier as the SMC for SGI ICE systems software. In this case, two events could happen:

- The imaging client could get a DHCP lease from the your house network DHCP server.
- Imaging could fail because it cannot reach the admin node.

The SMC for SGI ICE DHCP option identifier is 149, as shown by the `cadmin` command:

```
admin:~ # cadmin --show-dhcp-option  
149
```

You can use the `cadmin --set-dhcp-option {value}` option, to change the SMC for SGI ICE DHCP option identifier so it is different from your house network. For more information on the `cadmin` command, see "cadmin: SMC for SGI ICE Administrative Interface" on page 232.

Configure the Backup DNS Server

When this feature is configured, the compute nodes are able to use a service node as a secondary DNS server. If the rack leader goes down, the compute nodes running `tmpfs` will switch to use the service nodes as DNS server allowing them to function properly while the rack leader is being serviced. You can use the **Configure Backup DNS Server** option on the **Cluster Configuration Tool: Main Menu** to configure the backup DNS server, as shown in Figure 2-34 on page 94.

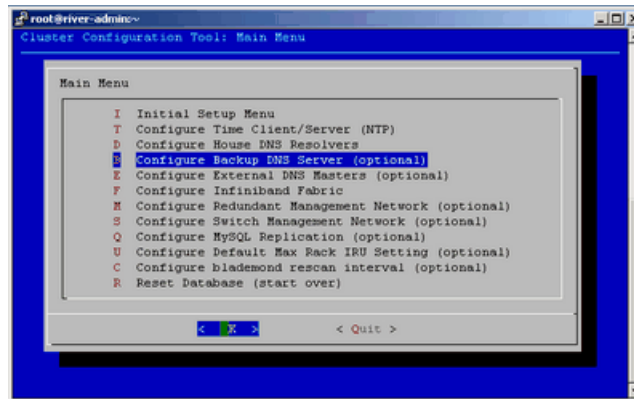


Figure 2-34 Configure Backup DNS Server Option Screen

Enter the service node that you want to use for DNS backup server, for example `service0`, and click on **OK**, as shown in Figure 2-35 on page 95.

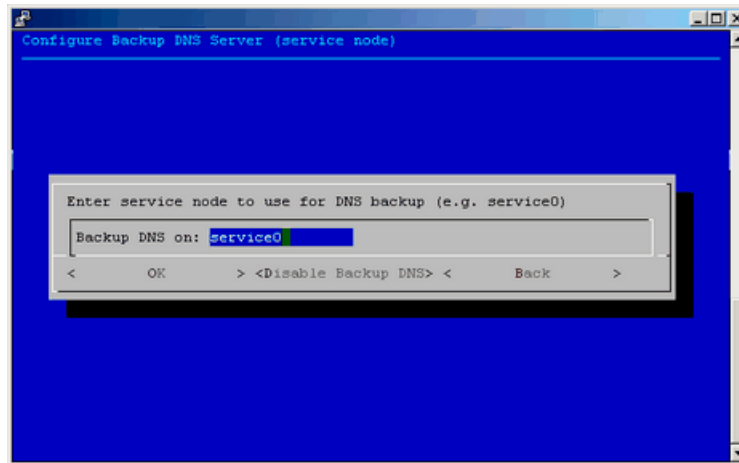


Figure 2-35 Enter Service Node to Use for DNS Backup Pop-up Menu

If you want to disable this feature, just select **Disable Backup DNS** from the same menu and confirm that **Yes** you have selected to disable DNS backup.

You can also use the `/opt/sgi/sbin/backup-dns-setup` utility to enable or disable the DNS backup, as follows:

```
# backup-dns-setup
usage: /opt/sgi/sbin/backup-dns-setup [--set-backup |
--delete-backup | --distribute-config | --show-backup]

# backup-dns-setup --show-backup
service0

# backup-dns-setup --delete-backup
Shutting down name server BIND
done
sys-admin: update-configs: updating SMC for SGI ICE configuration files
sys-admin: update-configs: -> dns
...

# backup-dns-setup --set-backup service0
Shutting down name server BIND waiting for named to shut down (29s)
done
```

```
sys-admin: update-configs: updating SMC for SGI ICE configuration files
sys-admin: update-configs: -> dns
...
```

InfiniBand Configuration

Before you start configuring the InfiniBand network, you need to ensure that all hardware components of the cluster have been discovered successfully, that is, admin, leader, service and compute nodes. You also need to be finished with the cluster configuration steps in "configure-cluster Command Cluster Configuration Tool" on page 56.

Sometimes, InfiniBand switch monitoring errors can appear, before the InfiniBand network has been fully configured. To disable InfiniBand switch monitoring, perform the following command:

```
% cattr set disableIbSwitchMonitoring true
```

To configure the InfiniBand network, start the `configure-cluster` command again on the admin node. Since the **Initial Setup** has been done already, you can now use the **Configure InfiniBand Fabric** option to configure the InfiniBand fabric as shown in Figure 2-36.

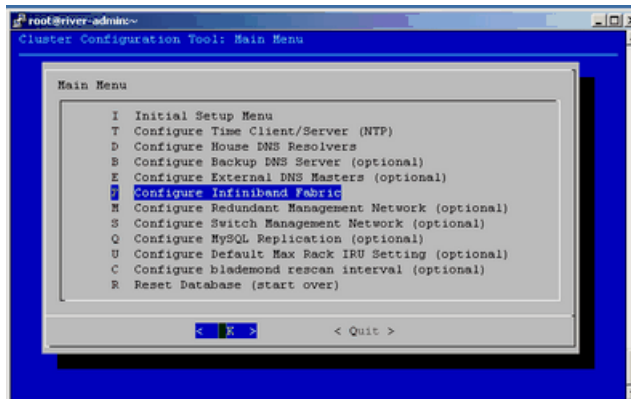


Figure 2-36 Configure InfiniBand Fabric from Cluster Configuration Tool

Select the **Configure InfiniBand Fabric** option, the InfiniBand Fabric Management tool appears, as shown in Figure 2-37.

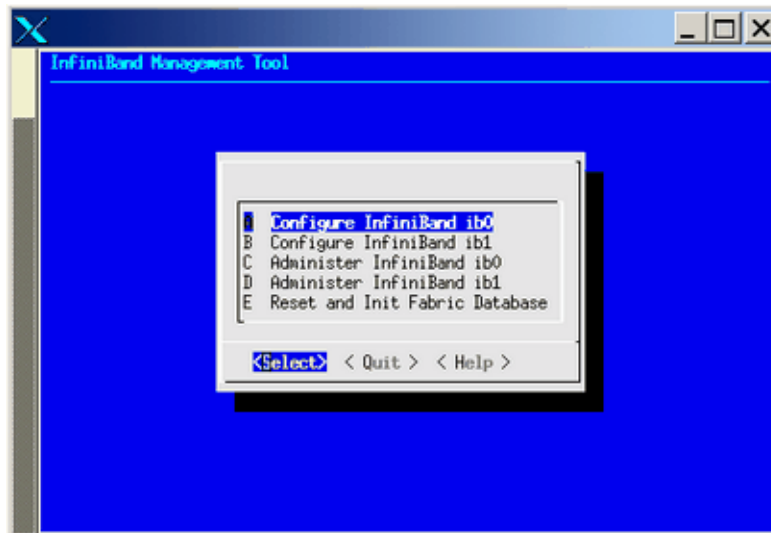


Figure 2-37 InfiniBand Management Tool Screen

Use the the online help available with this tool to guide you through the InfiniBand configuration. After configuring and bringing up the InfiniBand network, select the **Administer InfiniBand ib0** option or the **Administer InfiniBand ib1** option, the **Administer InfiniBand** screen appears as shown in Figure 2-38. Verify the status using the **Status** option.

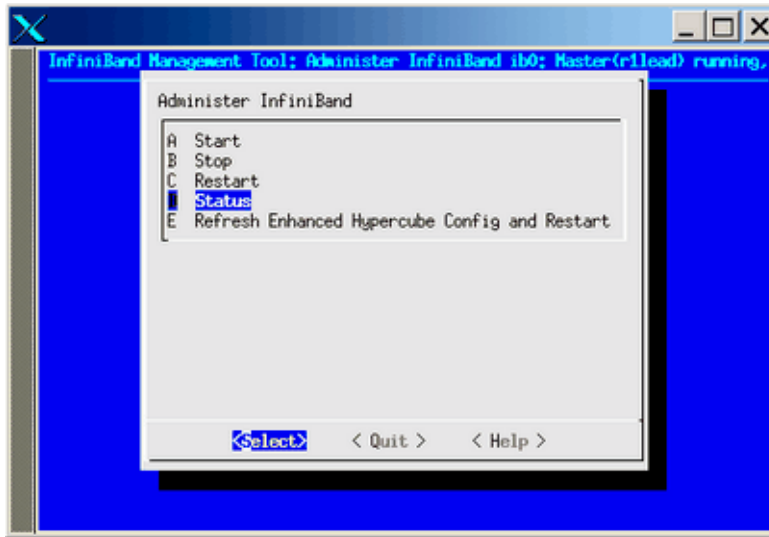


Figure 2-38 Administer InfiniBand GUI

Redundant Management Network

This section describes how to configure the redundant management network. For the SGI ICE 8400 series systems, when system nodes are discovered for the first time, the redundant management network value is turned on. **On** is the **default value**.

On SGI ICE X systems, when redundant management networking is enabled, the default Linux bonding mode for leader, service nodes is 802.3ad link aggregation. This can be set, adjusted with parameters to the `discover` command and the `cadmin` command. For more information, see the “Redundant Management Network” section in the *SGI Management Center for ICE X* guide.

For systems lacking a redundant ethernet interface, such as, the SGI ICE 8200 series systems, the redundant management network support is **off** by default. You can use the **Configure Redundant Management Network** option on the **Cluster Configuration Tool: Main Menu** to configure the redundant management network, as shown in Figure 2-39 on page 99.

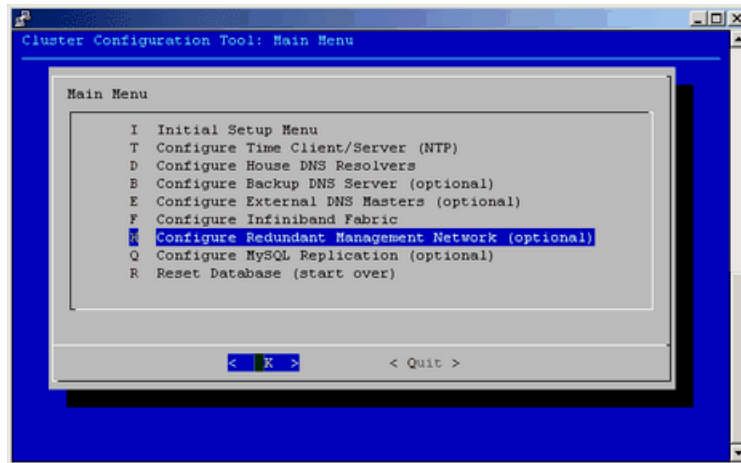


Figure 2-39 Configure Redundant Management Network Option Screen

Select the option and turn off or turn on the redundant network management value, as shown in Figure 2-40 on page 99.

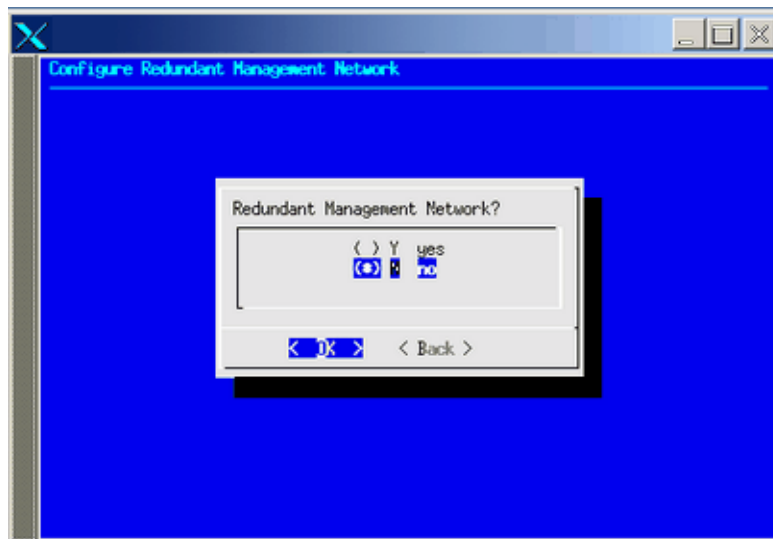


Figure 2-40 Redundant Management Network? Screen

Each node has the redundant network management either turned on or off. The redundant management network feature is **on**, by default, if the admin node has an `eth2` interface and **off**, by default, if the admin lacks `eth2`. The first time the `configure-cluster` command runs, it sets the admin node's value to the same as the global value. Additional executions of `configure-cluster` command will **not** set the admin nodes value.

When your are discovering a node, you can use an additional option to turn on or off the redundant management network for that node. For example:

```
# discover --service0,xe500,redundant_mgmt_network=no
```

This turns the value of `redundant_mgmt_network` to `no` (off) for `service0`. Valid values are `yes` and `no`.

Later, you can use the `cadmin` to change the `redundant_mgmt_network` value on or off, as follows:

```
# cadmin --set-redundant-mgmt-network --node service0 yes
```

```
# cadmin --set-redundant-mgmt-network --node service0 no
```

Example 2-2 Turning On the Redundant Management Network On Leader Node

To turn on the redundant management network on a rack leader controller (leader node), perform the following command:

```
# cadmin --set-redundant-mgmt-network --node r1lead yes  
r1lead should now be rebooted.
```

The `cadmin` command returns a message to reboot the `r1lead` node.

To reboot the node, perform the following command:

```
# cpower --reboot r1lead
```

Switch Management Network

This section describes how to configure the switch management network. You can use the **Configure Switch Management Network** option on the **Cluster Configuration Tool: Main Menu** to configure the switch management network, as shown in Figure 2-41 on page 101.

Note: The system software attempts to automatically set this value to an appropriate value.

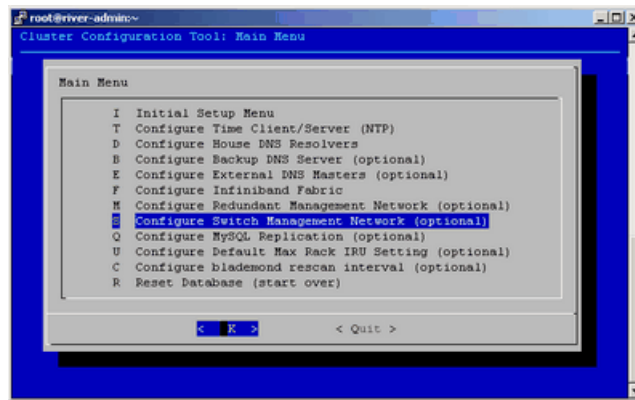


Figure 2-41 Configure Switch Management Network Option Screen

If your site has SGI ICE X racks integrated with earlier SGI ICE 8400/8200 racks, you should set the default value to **yes** as shown in Figure 2-42 on page 101, otherwise, set the value to **no**.

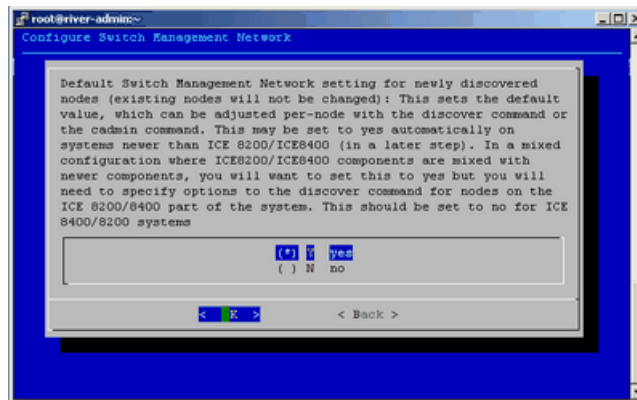


Figure 2-42 Default Switch Management Network Setting Screen

MySQL Database Server Replication

Replication in MySQL enables data from the master MySQL database server (admin node) to be replicated to one or more MySQL database slaves (leader and service nodes).

If your site has a large number of racks, using this feature can reduce the amount of contention for database resources on the admin node.

Replication is enabled by default. From the **Cluster Configuration Tool: Main Menu** (launched by the `configure-cluster` command), select **Configure MySQL Replication (optional)**, as shown in Figure 2-43 on page 102.

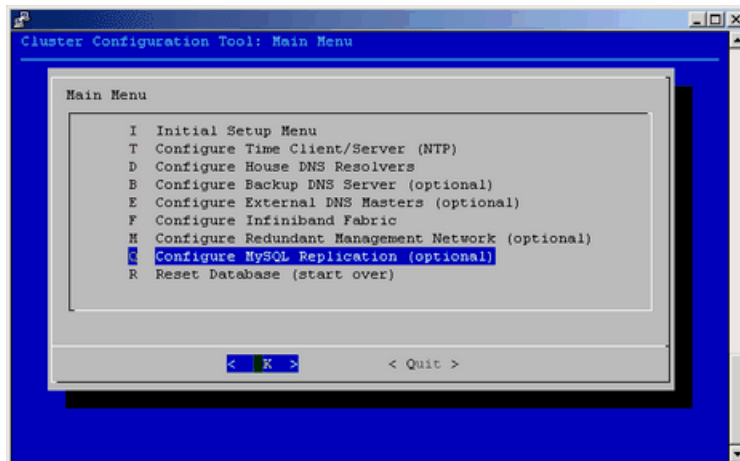


Figure 2-43 Configure MySQL Replication (optional) Option

Select the **yes** on the **Configure MySQL Replication** screen, as shown in Figure 2-44 on page 103, and press **OK** to enable MySQL replication. Press **no** to disable this feature.

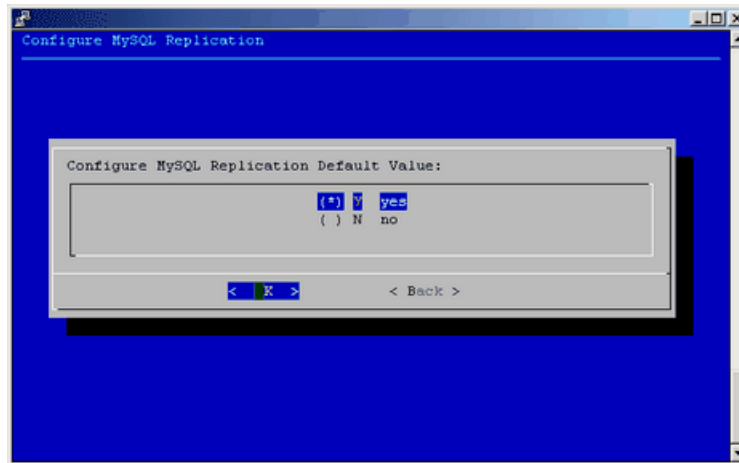


Figure 2-44 Configure MySQL Replication Screen

When enabling or disabling this feature, the `configure-cluster` command will back up the database, save some system attributes, and call `/etc/opt/sgi/conf.d/80-update-mysql` on the admin node, leader nodes, and service nodes.

When replication is OFF and the `cattr` command is run by a script on a leader or service node, it uses the database on the admin node. You can verify this, as follows:

```
r1lead:~ # chkconfig -l mysql
mysql                                0:off 1:off 2:off 3:off 4:off 5:off 6:off
r1lead:~ # grep -e hostname /etc/opt/sgi/cattr.conf
hostname = admin
r1lead:~ # cattr list | grep my_sql_replication
my_sql_replication                  : no
```

When replication is ON and `cattr` is run by a script on a leader or service node, it uses the replicated database on the node itself. You can verify this, as follows:

```
r1lead:~ # chkconfig -l mysql
mysql                                0:off 1:off 2:on 3:on 4:off 5:on 6:off
r1lead:~ # grep -e hostname /etc/opt/sgi/cattr.conf
hostname = localhost
r1lead:~ # cattr list | grep my_sql_replication
```

```
my_sql_replication      : yes
```

To verify if database replication is working on a leader or service node, perform the following:

```
sys-admin:~ # cadmin --show-replication-status --node {node}: Show current value.
```

See Chapter 15 of the *MySQL 5.0 Reference Manual* for detailed information regarding how replication is implemented and configured. This manual is available at <http://dev.mysql.com/doc/refman/5.0/en/replication.html>.

Configure Maximum IRUs Supported by a RLC

The **Configure Default Max Rack IRU Setting** option, allows you to set the default number of individual rack units (IRUs) supported by a rack leader controller (RLC), as shown in Figure 2-45 on page 104. For systems with a smaller number of IRUs, setting this value appropriate to your system size can decrease the time it takes the `image --push-rack` command to push new software images to blades in an IRU.

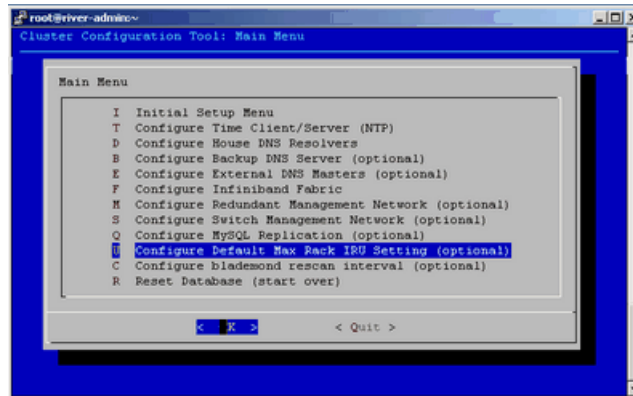


Figure 2-45 Configure Default Max Rack IRU Setting Option Screen

The default maximum number of IRUs per RLC (leader node) is set to four (4) for legacy systems (SGI ICE 8200 or SGI ICE 8400 systems), as shown in Figure 2-46 on page 105. Most SGI ICE X systems have an equivalent of eight IRU (blade enclosure

pairs in an SGI ICE X system) and eight CMCs, therefore, this value should be set to eight (8).

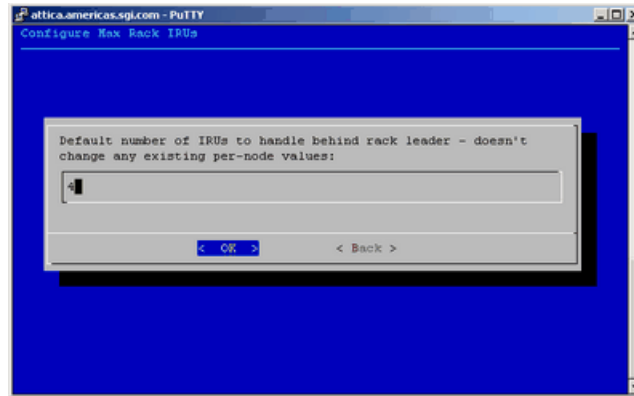


Figure 2-46 Configure Max Rack IRUs Screen

The `configure-cluster` value sets the default for nodes discovered from this point onward. If rack leader controllers (RLCs) have already been discovered, you need to use the `cadmin --show-max-rack-irus` and `--show-max-rack-irus` options to show and change this value.

Configuring the Service Node

This section describes how to configure a service node and covers the following topics:

- "Service Node Configuration for NAT" on page 106
- "Using External DNS for Compute Node Name Resolution " on page 108
- "Service Node Configuration for DNS" on page 109
- "Service Node Configuration for NFS " on page 109
- "Service Node Configuration for NIS for the House Network" on page 109

Service Node Configuration for NAT

You may want to reach network services outside of your SGI ICE system. For this type of access, SGI recommends using Network Address Translation (NAT), also known as IP Masquerading or Network Masquerading. Depending on the amount of network traffic and your site needs, you may want to have multiple service nodes providing NAT services.

Procedure 2-11 Service Node Configuration for NAT

To enable NAT on your service node, perform the following steps:

1. Use the configuration tools provided on your service node to turn on IP forwarding and enable NAT/IP MASQUERADE.

Specific instructions should be available in the third-party documentation provided for your storage node system. Additional documentation is available at `/opt/sgi/docs/setting-up-NAT/README`. This document describes how to get NAT working for both IB interfaces.

Note: This file is only on the service node. You need to `# ssh service0` and then from service0 `# cd /opt/sgi/docs/setting-up-NAT`.

2. Update the all of the compute node images with default route configured for NAT.
SGI recommends a script on the system admin controller at `/opt/sgi/share/per_host_customization/global/sgi-static-routes` that can customize the routes based upon rack, IRU, and slot of the compute blade. Some examples are available in that script.
3. Use the `cimage --push-rack` command to propagate the changes to the proper location for compute nodes to boot. For more information on using the `cimage` command, see "cimage Command" on page 206 and "Customizing Software On Your SGI ICE System" on page 198.
4. Use the `cimage --set` command to select the image.
5. Reboot/reset the compute nodes using that desired image.
6. Once the service node(s) has NAT enabled, is attached to an operational house network, and the compute nodes are booted from an image which sets their routing to point at the service node, test the NAT operation by using the `ping(8)`

command to ping known IP addresses on the house network from an interactive session on the compute blade.

7. See the troubleshooting discussion that follows.

Troubleshooting Service Node Configuration for NAT

Troubleshooting can become very complex. The first steps are to determine that the service node(s) are correctly configured for the house network and can ping the house IP addresses. Good choices are house name servers possibly found in the `/etc/resolv.conf` or `/etc/name.d.conf` files on the admin node. Additionally, the default gateway addresses for the service node may be a good choice. You can use the `netstat -rn` command for this information, as follows:

```
system-1:/ # netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
128.162.244.0    0.0.0.0         255.255.255.0   U        0 0        0 eth0
172.16.0.0       0.0.0.0         255.255.0.0     U        0 0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U        0 0        0 eth0
172.17.0.0      0.0.0.0         255.255.0.0     U        0 0        0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U        0 0        0 lo
0.0.0.0         128.162.244.1  0.0.0.0         UG       0 0        0 eth0
```

If the ping command executed from the service node to the selected IP address gets responses, network monitoring tools such as `tcpdump(1)` should be used. On the service node, monitor the `eth1` interface and simultaneously in a separate session monitor the `ib[01]` interface. You should specify monitoring specific-enough to not have additional noise then attempt execute a ping command from the compute node.

Example 2-3 tcpdump Command Examples

```
tcpdump -i eth1 ip proto ICMP # Dump ping packets on the public side of service node.
tcpdump -i ib1 ip proto ICMP # Dump ping packets on the IB fabric side of service node.
tcpdump -i eth1 port nfs # Dump NFS traffic on the eth1 side of service node.
tcpdump -i ib1 port nfs # Dump NFS traffic on the eth1 side of service node.
```

If packets do not reach the service nodes respective IB interface, perform the following:

- Check the system admin controller's compute image configuration of the default route.

- Verify that this image has been pushed to the compute nodes.
- Verify that the compute nodes have booted with this image.

If the packets reach the service nodes IB interface, but do not exit the `eth1` interface, verify the NAT configuration on the service node.

If the packets exit the `eth1` interface, but replies do not return, verify the house network configuration and that IP masquerading is properly configured so that the packets exiting the interface appear to be originating from the service node and not the compute node.

Using External DNS for Compute Node Name Resolution

You may want to configure service node(s) to act as NAT gateways for your cluster (see "Service Node Configuration for NAT" on page 106) and to have the host names for the compute nodes in the cluster resolve through external DNS servers.

You need to reserve a large block of IP addresses on your house network. If you configure to resolve via external DNS, you need to do it for both the `ib0` and `ib1` networks, for all node types. In other words, **ALL** `-ib*` addresses need to be provided by external DNS. This includes compute nodes, leader nodes, and service nodes. Careful planning is required to use this feature. Allocation of IP addresses will often require assistance from a network administrator of your site.

Once the IP addresses have been allocated on the house network, you need to tell the SMC for SGI ICE software the IP addresses of the DNS servers on the house network that the SMC for SGI ICE software can query for hostname resolution.

To do this, use the `configure-cluster` tool (see "configure-cluster Command Cluster Configuration Tool" on page 56). The menu item that handles this operation is **Configure External DNS Masters (optional)**.

Some important considerations are, as follows:

- It is important to note that if you choose to use external DNS, you need to make this change **before** discovering anything. The change is **not** retroactive. If you have already discovered some nodes, then turn on external DNS support, the IP addresses assigned by SMC for SGI ICE for the nodes already discovered will remain.
- This is an optional feature that only a small set of customers will need to use. It should not be used by default.

- This feature only makes sense if the compute nodes can reach the house network. This is not the default case for SGI ICE systems.
- It is assumed that you have already configured a service node to act as a NAT gateway to your house network (see "Service Node Configuration for NAT" on page 106) and that the compute nodes have been configured to use that service node as their gateway.

Service Node Configuration for DNS

For information on setting up DNS, see Figure 2-29 on page 76.

Service Node Configuration for NFS

Assuming the installation has either NAT or Gateway operations configured on one or more service nodes, the compute nodes can directly mount the house NFS server's exports (see the `exports(5)` man page).

Procedure 2-12 Service Node Configuration for NFS

To allow the compute nodes to directly mount the house NFS server's exports, perform the following steps:

1. Edit the system admin controller's `/opt/sgi/share/per_host_customization/global/sgi-fstab` file or alternatively an image-specific script.
2. Add the mount point, push the image, and reset the node.
3. The server's export should get mounted. If it is not, use the technique for troubleshooting outlined in "Troubleshooting Service Node Configuration for NAT" on page 107.

Service Node Configuration for NIS for the House Network

This section describes two different ways to configure NIS for service nodes and compute blades when you want to use the house network NIS server, as follows:

- NIS with the compute nodes directly accessing the house NIS infrastructure
- NIS with a service node as a NIS slave server to the house NIS master

The first approach would be used in the case where a service node is configured with network address translation (NAT) or gateway operations so that the compute nodes can access the house network directly.

The second approach may be used if the compute nodes do not have direct access to the house network.

Procedure 2-13 NIS with Compute Nodes Directly Accessing the House NIS Infrastructure

To setup NIS with the compute nodes directly accessing the house NIS infrastructure, perform the following steps:

1. In this case, you do not have to set up any additional NIS servers. Instead, each service node and compute node should be configured to bind to the existing house network servers. The nodes should already have the `ypbind` package installed. The following steps should work with most Linux distributions. You may need to vary them slightly to meet your specific needs.
2. For service nodes, the instructions are very similar to those found in "Setting Up a SLES Service Node as a NIS Client" on page 120.

The only difference is that you should configure `yp.conf` to look at the IP address of your house network NIS server and not the leader node as is described in the sections listed, above.

Procedure 2-14 NIS with a Service Node as a NIS Slave Server to the House NIS Master

To setup NIS with a service node as a NIS slave server to the house NIS master, perform the following:

1. Any service nodes that are NOT acting as an NIS slave server can be pointed at the existing house network NIS servers as described in Procedure 2-13, page 110. This is because they have house interfaces.
2. One (or more) service node(s) should be then be configured as NIS slave server(s) to the existing house network NIS Master server.

Since SGI can not anticipate what operating system or release the house network NIS Master server is running, no suggestions on any configuration you need to do to tell it that you are adding new NIS slave servers can be offered.

Setting Up an NFS Home Server on a Service Node for Your SGI ICE System

This section describes how to make a service node an NFS home directory server for the compute nodes.

Note: Having a single, small server provide filesystems to the whole SGI ICE system could create network bottlenecks that the hierarchical design of SGI ICE is meant to avoid, especially if large files are stored there. Consider putting your home filesystems on an NAS file server. For instructions on how to do this, see "Service Node Configuration for NFS " on page 109.

The instructions in this section assume you are using the service node image provided with the SMC for SGI ICE software. If you are using your own installation procedures or a different operating system, the instructions will not be exact but the approach is still appropriate.

Note: The example below specifically avoids using `/dev/sdX` style device names. This is because `/dev/sdX` device names are not persistent and may change as you adjust disks and RAID volumes in your system. In some situations, you may assume `/dev/sda` is the system disk and that `/dev/sdb` is a data disk; this is **not** always the case. To avoid accidental destruction of your root disk, follow the instructions given below.

When you are choosing a disk, please consider the following:

To pick a disk device, first find the device that is being currently used as root. Avoid re-partitioning the installation disk by accident. To find which device is being used for root, use this command:

```
# ls -l /dev/disk/by-label/sgiroot
lrwxrwxrwx 1 root root 10 2008-03-18 04:27 /dev/disk/by-label/sgiroot ->
../../sda2
```

At this point, you know the `sd` name for your root device is `sda`.

SGI suggests you use `by-id` device names for your data disk. Therefore, you need to find the `by-id` name that is NOT your root disk. To do that, use `ls` command to list the contents of `/dev/disk/by-id`, as follows:

```
# ls -l /dev/disk/by-id
total 0
lrwxrwxrwx 1 root root 9 2008-03-20 04:57 ata-MATSHITADVD-RAM_UJ-850S_HB08_020520 -> ../../hdb
lrwxrwxrwx 1 root root 9 2008-03-20 04:57 scsi-3600508e00000000307921086e156100 -> ../../sda
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e00000000307921086e156100-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e00000000307921086e156100-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e00000000307921086e156100-part5 -> ../../sda5
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e00000000307921086e156100-part6 -> ../../sda6
lrwxrwxrwx 1 root root 9 2008-03-20 04:57 scsi-3600508e000000008dced2cfc3c1930a -> ../../sdb
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000008dced2cfc3c1930a-part1 -> ../../sdb1
lrwxrwxrwx 1 root root 9 2008-03-20 09:57 usb-PepperC_Virtual_Disc_1_0e159d01a04567ab14E72156DB3AC4FA -> .....
```

In the output, above, you can see that ID `scsi-3600508e00000000307921086e156100` is in use by your system disk because it has a symbolic link pointing back to `../../sda`. So do not consider that device. The other disk in the listing has ID `scsi-3600508e000000008dced2cfc3c1930a` and happens to be linked to `/dev/sdb`.

Therefore, you know the `by-id` name you should use for your data is `/dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a` because it is not connected with `sda`, which we found in the first `ls` example happened to be the root disk.

Partitioning, Creating, and Mounting Filesystems

Procedure 2-15 Partitioning and Creating Filesystems for an NFS Home Server on a Service Node

The following example uses `/dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a` ID as the empty disk on which you will put your data. It is very important that you know this for sure. In "Setting Up an NFS Home Server on a Service Node for Your SGI ICE System", an example is provided that allows you to determine where your root disk is located so you can avoid accidentally destroying it. Remember, in some cases, `/dev/sdb` will be the root drive and `/dev/sda` or `/dev/sdc` may be the data drive.

Please confirm that you have selected the right device, and use the persistent device name to help prevent accidental overwriting of the root disk.

Note: Steps 1 through 7 of this procedure are performed on the service node. Steps 8 and 9 are performed from the system admin controller (admin node).

To partition and create filesystems for an NFS home server on a service node, perform the following steps:

1. Use the `parted(8)` utility, or some other partition tool, to create a partition on `/dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a`. The following example makes one filesystem out of the disk. You can use the `parted` utility interactively or in a command-line driven manner.
2. Make a new `msdos` label, as follows:

```
# # parted /dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a mkpart primary ext2 0 249GB
Information: Don't forget to update /etc/fstab, if necessary.
```

3. Find the size of the disk, as follows:

```
# # parted /dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a print
Disk geometry for /dev/sdb: 0kB - 249GB
Disk label type: msdos
Number  Start   End     Size    Type    File system  Flags
Information: Don't forget to update /etc/fstab, if necessary.
```

4. Create a partition that spans the disk, as follows:

```
# # parted /dev/disk/by-id/scsi-3600508e000000008dced2cfc3c1930a mkpart
primary ext2 0 249GB
Information: Don't forget to update /etc/fstab, if necessary.
```

5. Issue the following command to cause the `/dev/disk/by-id` partition device file is in place and available for use with the `mkfs` command that follows:

```
# udevtrigger
```

6. Create a filesystem on the disk. You can choose the filesystem type.

Note: The `mkfs.ext3` command takes more than 10 minutes to create a single 500GB filesystem using default `mkfs.ext3` options. If you do not need the number of inodes created by default, use the `-N` option to `mkfs.ext3` or other options that reduce the number of inodes. The following example creates 20 million inodes. XFS filesystems can be created in much shorter time.

An `ext3` example is, as follows:

```
# mkfs.ext3 -N 20000000 /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a-part1
```

An `xfs` example is, as follows:

```
# mkfs.xfs /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a-part1
```

7.

Add the newly created filesystem to the server's `fstab` file and mount it. Ensure that the new filesystem is exported and that the NFS service is running, as follows:

a. Append the following line to your `/etc/fstab` file.

```
/dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a-part1 /home ext3 defaults 1
```

Note: If you are using XFS, replace `ext3` with `xfs`. This example uses the `/dev/disk/by-id` path for the device and not a `/dev/sd` device.

b. Mount the new filesystem (the `fstab` entry, above, enables it to mount automatically the next time the system is rebooted), as follows:

```
# mount -a
```

c. Be sure the filesystem is exported. Add the following line to `/etc/exports` file. Adjust this line to match your site's access policies.

```
/home *(no_subtree_check,rw,async,no_root_squash)
```

d.

Note: In some distros, the NFS server init script is simply `"nfs"`

Make sure the NFS server service is enabled. For SLES, use these commands:

```
# chkconfig nfsserver on
# /etc/init.d/nfsserver restart
```

Note: Steps 8 and 9 are performed from the system admin controller (admin node).

8. The following steps describe how to mount the home filesystem on the compute nodes, as follows:

Note: SGI recommends that you always work on clones of the SGI-supplied compute image so that you always have a base to copy to fall back to if necessary. For information on cloning a compute node image, see "Customizing Software Images" on page 203.

- a. Make a mount point in the blade image. In the following example, `/home` already is a mount point. If you used a different mount point, you need to do something similar to the following on the system admin controller. Note that the rest of the examples will resume using `/home`.

```
# mkdir /var/lib/systemimager/images/compute-sles11-clone/my-mount-point
```

- b. Add the `/home` filesystem to the compute nodes. SGI supplies an example script for managing this. You just need to add your new mount point to the `sgi-fstab` post-host-customization script.
- c. Use a text editor to edit the following file:

```
/opt/sgi/share/per-host-customization/global/sgi-fstab
```

- d. Insert the following line just after the `tmpfs` and `devpts` lines in the `sgi-fstab` file:

```
service0-ib1:/home /home nfs hard 0 0
```

Note: In order to maximize performance, SGI advises that the `ib0` fabric be used for all MPI traffic. The `ib1` fabric is reserved for storage related traffic.

- e. Use the `cimage` command to push the update to the rack leader controllers serving each compute node, as follows:

```
# cimage --push-rack compute-sles11-clone "r**"
```

Using `--push-rack` on an image that is already on the rack leader controllers has the simple affect of updating them with the change you made above. For more information on using the `cimage`, see "cimage Command" on page 206.

9. When you reboot the compute nodes, they will mount your new home filesystem.

For information on centrally managed user accounts, see "Setting Up a NIS Server for Your SGI ICE System" on page 117. It describes NIS master set up. In this design, the master server residing on the service node provides the filesystem and the NIS slaves reside on the rack leader controllers. If you have more than one home server, you need to export all home filesystems on all home servers to the server acting as the NIS master. You also need to export the filesystems to the NIS master using the `no_root_squash exports` flag.

Home Directories on NAS

If you want to use NAS server for scratch storage or make home filesystems available on NAS, you can follow the instructions in "Setting Up an NFS Home Server on a Service Node for Your SGI ICE System" on page 111. In this example, you need to replace `service0-ib1` with the `ib1` InfiniBand host name for the NAS server and you need to know where on the NAS server the home filesystem is mounted to craft the `sgi-fstab` script properly.

RHEL Service Node House Network Configuration

If you plan to put your service node on the house network, you need to configure it for networking. For this, you may use the `system-config-network` command. It is better to use the graphical version of the tool if you are able. Use the `ssh -X` command from your desktop to connect to the admin node and then again to connect to the service node. This should redirect graphics over to your desktop.

Some helpful hints are, as follows:

- On service nodes, the cluster interface is `eth0`. Therefore, do not configure this interface as it is already configured for the cluster network.

- Do not make the public interface a `dhcp` client as this can overwrite the `/etc/resolv.conf` file.
- Do not configure name servers, the name server requests on a service node are always directed to the admin leader nodes for resolution. If you wish to resolve network addresses on your house network, just be sure to enable the **House DNS Resolvers** using `configure-cluster` command on the admin node.
- Do not configure or change the search order, as this again could adjust what cluster management has placed in the `/etc/resolv.conf` file.
- Do not change the host name using the RHEL tools. You can change the hostname using the `cadmin` tool on the admin node.
- After configuring your house network interface, you can use the `ifupethX` command to bring the interface up. Replace `X` with your house network interface.
- If you wish this interface to come up by default when the service node reboots, be sure `ONBOOT` is set to `yes` in `/etc/sysconfig/network-scripts/ifcfg-ethX` (again, replace `X` with the proper value). The graphical tool allows you to adjust this setting while the text tool does not.
- If you happen to wipe out the `resolv.conf` file by accident and end up replacing it, you may need to issue this command to ensure that DNS queries work again:

```
# nscd --invalidate hosts
```

Setting Up a NIS Server for Your SGI ICE System

This section describes how to set up a network information service (NIS) server running SLES 11 for your SGI ICE system. If you would like to use an existing house network NIS server, see "Service Node Configuration for NIS for the House Network" on page 109. This section covers the following topics:

- "Setting Up a NIS Server Overview" on page 118
- "Setting Up a SLES Service Node as a NIS Master" on page 118
- "Setting Up a SLES Service Node as a NIS Client" on page 120
- "Setting up a SLES Rack Leader Controller as a NIS Slave Server and Client" on page 121
- "NAS Configuration for Multiple IB Interfaces" on page 122
- "Setting up the SLES Compute Nodes to be NIS Clients" on page 122

- "Creating User Accounts" on page 125
- "Tasks You Should Perform After Changing a Rack Leader Controller" on page 125

Setting Up a NIS Server Overview

In the procedures that follow in this section, here are some of the tasks you need to perform and system features you need to consider:

- Make a service node the NIS master
- Make the rack leader controllers (leader nodes) the NIS slave servers
- Do **not** make the system admin controller as the NIS master because it may not be able to mount all of the storage types. Having the storage mounted on the NIS master server makes it far less complicated to add new accounts using NIS.
- If multiple service nodes provide home filesystems, the NIS master should mount all remote home filesystems. They should be exported to the NIS master service node with the `no_root_squash export` option. The example in the following section assumes a single service node with storage and that same node is the NIS master.
- No NIS traffic goes over the InfiniBand network.
- Compute node NIS traffic goes over Ethernet, not InfiniBand, by way of using a the `lead-eth` server name in the `yp.conf` file. This design feature prevents NIS traffic from affecting the InfiniBand traffic between the compute nodes.

Setting Up a SLES Service Node as a NIS Master

This section describes how to set up a service node as a NIS master. This section only applies to service nodes running SLES.

Procedure 2-16 Setting Up a SLES Service Node as a NIS master

To set up a SLES service node as a NIS master, from the service node, perform the following steps:

Note: These instructions use the text-based version of YaST. The graphical version of YaST may be slightly different.

1. Start up YaST, as follows:

```
# yast nis_server
```

2. Choose **Create NIS Master Server** and click on **Next** to continue.
3. Choose an NIS domain name and place it in the NIS Domain Name window. This example, uses **ice**.
 - a. Select **This host is also a NIS client**.
 - b. Select **Active Slave NIS server exists**.
 - c. Select **Fast Map distribution**.
 - d. Select **Allow changes to passwords**.
 - e. Click on **Next** to continue.
4. Set up the NIS master server slaves.

Note: You are now in the **NIS Master Server Slaves Setup**. Just now, you can enter the already defined rack leader controllers (leader nodes) here. If you add more leader nodes or re-discover leader nodes, you will need to change this list. For more information, see "Tasks You Should Perform After Changing a Rack Leader Controller" on page 125.

5. Select **Add** and enter **r1lead** in the **Edit Slave** window. Enter any other rack leader controllers you may have just like above. Click on **Next** to continue.
6. You are now in **NIS Server Maps Setup**. The default selected maps are okay. Avoid using the **hosts** map (not selected by default) because can interfere with SGI ICE system operations. Click on **Next** to continue.
7. You are now in **NIS Server Query Hosts Setup**. Use the default settings here. However, you may want to adjust settings for security purposes. Click on **Finish** to continue.

At this point, the NIS master is configured. Assuming you checked the **This host is also a NIS client box**, the service node will be configured as a NIS client to itself and start `yp ypbind` for you.

Setting Up a SLES Service Node as a NIS Client

This section describes how to use YaST to set up your other service nodes to be broadcast binding NIS clients. This section only applies to service nodes running SLES11.

Note: You do not do this on the NIS Master service node that you already configured as a client in "Setting Up a SLES Service Node as a NIS Master" on page 118.

Procedure 2-17 Setting Up a SLES Service Node as a NIS Client

To set up a service node as a NIS client, perform the following steps:

1. Enable `ypbind`, perform the following:

```
# chkconfig ypbind on
```

2. Set the default domain (already set on NIS master). Change `ice` (or whatever domain name you choose above) to be the NIS domain for your SGI ICE system, as follows:

```
# echo "ice" > /etc/defaultdomain
```

3. In order to ensure that no NIS traffic goes over the IB network, SGI does **not** recommend using NIS broadcast binding on service nodes. You can list a few leader nodes in the `/etc/yp.conf` file on non-NIS-master service nodes. The following is an example `/etc/yp.conf` file. Add or remove rack leader nodes as appropriate. Having more entries in the list allows for some redundancy. If `r1lead` is hit by excessive traffic or goes down, `ypbind` can use the next server in the list as its NIS server. SGI does not suggest listing other service nodes in `yp.conf` file because all resolvable names for service nodes on service nodes use IP addresses that go over the InfiniBand network. For performance reasons, it is better to keep NIS traffic off of the InfiniBand network.

```
ypserver r1lead
ypserver r2lead
```

4. Start the `ypbind` service, as follows:

```
# rcypbind start
```

The service node is now bound.

5. Add the NIS include statement to the end of the password and group files, as follows:

```
# echo "+:::" >> /etc/group
# echo "+:::::" >> /etc/passwd
# echo "+" >> /etc/shadow
```

Setting up a SLES Rack Leader Controller as a NIS Slave Server and Client

This section provides two sets of instructions for setting up rack leader controllers (leader nodes) as NIS slave servers. It is possible to make all these adjustments to the leader image in `/var/lib/systemimager/images`. Currently, SGI does not recommend using this approach.

Note: Be sure the InfiniBand interfaces are up and running before proceeding because the rack leader controller gets its updates from the NIS Master over the InfiniBand network. If you get a "can't enumerate maps from service0" error, check to be sure the InfiniBand network is operational.

Procedure 2-18 Setting up a Rack Leader Controller as a NIS Slave Server and Client

Use the following set of commands from the system admin controller (admin node) to set up a rack leader controller (leader node) as a NIS slave server and client.

Note: Replace `ice` with your NIS domain name and `service0` with the service node you set up as the master server.

```
admin:~ # cexec --head --all chkconfig ypserv on
admin:~ # cexec --head --all chkconfig ypbind on
admin:~ # cexec --head --all chkconfig portmap on
admin:~ # cexec --head --all chkconfig nscd on
admin:~ # cexec --head --all rcportmap start
admin:~ # cexec --head --all "echo ice > /etc/defaultdomain"
admin:~ # cexec --head --all "ypdomainname ice"
admin:~ # cexec --head --all "echo ypserver service0 > /etc/yp.conf"
admin:~ # cexec --head --all /usr/lib/yp/ypinit -s service0
admin:~ # cexec --head --all rcportmap start
admin:~ # cexec --head --all rcypserv start
admin:~ # cexec --head --all rcypbind start
admin:~ # cexec --head --all rcnscd start
```

Setting up the SLES Compute Nodes to be NIS Clients

This section describes how to set up the compute nodes to be NIS clients. You can configure NIS on the clients to use a server list that only contains the their rack leader controller (leader node). All operations are performed from the system admin controller (admin node).

Procedure 2-19 Setting up the Compute Nodes to be NIS Clients

To set up the compute nodes to be NIS clients, perform the following steps:

1. Create a compute node image clone. SGI recommends that you always work with a clone of the compute node images. For information on how to clone the compute node image, see "Customizing Software Images" on page 203.
2. Change the compute nodes to use the cloned image/kernel pair, as follows:

```
admin:~ # cimage --set compute-sles11-clone 2.6.16.46-0.12-smp "r*i*n"
```

3. Set up the NIS domain, as follows (`ice` in this example):

```
admin:~ # echo "ice" > /var/lib/systemimager/images/compute-sles11-clone/etc/defaultdomain
```

4. Set up compute nodes to get their NIS service from their rack leader controller (fix the domain name as appropriate), as follows:

```
admin:~ # echo "ypserver lead-eth" > /var/lib/systemimager/images/compute-sles11-clone/etc/yp.conf
```

5. Enable the `ypbind` service, using the `chroot` command, as follows:

```
admin:~# chroot /var/lib/systemimager/images/compute-sles11-clone chkconfig ypbind on
```

6. Set up the password, shadow, and group files with NIS includes, as follows:

```
admin:~# echo "+:::" >> /var/lib/systemimager/images/compute-sles11-clone/etc/group
```

```
admin:~# echo "+:::::" >> /var/lib/systemimager/images/compute-sles11-clone/etc/passwd
```

```
admin:~# echo "+" >> /var/lib/systemimager/images/compute-sles11-clone/etc/shadow
```

7. Push out the updates using the `cimage` command, as follows:

```
admin:~ # cimage --push-rack compute-sles11-clone "r"
```

NAS Configuration for Multiple IB Interfaces

The NAS cube needs to get configured with each InfiniBand fabric interface in a separate subnet. These fabrics will be separated from each other logically, but attached to the same physical network. For simplicity, this guide assumes that the `-ib1` fabric

for the compute nodes has addresses assigned in the 10.149.0.0/16 network. This guide also assumes the lowest address the cluster management software has used is 10.149.0.1 and the highest is 10.149.1.3 (already assigned to the NAS cube).

For the NAS cube, you need to configure the large physical network into four, smaller subnets, each of which would be capable of containing all the nodes and service nodes. It will have subnets 10.149.0.0/18, 10.149.64.0/18, 10.149.128.0/18, and 10.149.192.0/18.

After the discovery of the storage node has happened, SGI personnel will need to log onto the NAS box and change the network settings to use the smaller subnets, and then define the other three adapters with the same offset within the subnet; for example: Initial configuration of the storage node had set ib0 fabric's IP to 10.149.1.3 netmask 255.255.0.0. After the addresses are changed, ib0=10.149.1.3:255.255.192.0, ib1=10.149.65.3:255.255.192.0, ib2=10.149.129.3:255.255.192.0, ib3=10.149.193.3:255.255.192.0. The NAS cube should now have all four adapter connections connected to the fabric with IP addresses which can be pinged from the service node.

Note: The service nodes and the rack leads will remain in the 10.149.0.0/16 subnet.

For the compute blades, log into the admin node and modify /opt/sgi/share/per-host-customization/global/sgi-setup-ib-configs file. Following the line iruslot=\$1, insert:

```
# Compute NAS interface to use
IRU_NODE=`basename ${iruslot}`
RACK=`cminfo --rack`
RACK=$(( ${RACK} - 1 ))
IRU=`echo ${IRU_NODE} | sed -e s/i// -e s/n.*//`
NODE=`echo ${IRU_NODE} | sed -e s/.*/n//`
POSITION=$(( ${IRU} * 16 + ${NODE} ))
POSITION=$(( ${RACK} * 64 + ${POSITION} ))
NAS_IF=$(( ${POSITION} % 4 ))
NAS_IPS[0]="10.149.1.3"
NAS_IPS[1]="10.149.65.3"
NAS_IPS[2]="10.149.129.3"
NAS_IPS[3]="10.149.193.3"
```

Then following the line `$iruslot/etc/opt/sgi/cminfo` add:

```
IB_1_OCT12=`echo ${IB_1_IP} | awk -F "." '{ print $1 "." $2 }`  
IB_1_OCT3=`echo ${IB_1_IP} | awk -F "." '{ print $3 }`  
IB_1_OCT4=`echo ${IB_1_IP} | awk -F "." '{ print $4 }`  
IB_1_OCT3=$(( ${IB_1_OCT3} + ${NAS_IF} * 64 )  
IB_1_NAS_IP="${IB_1_OCT12}.${IB_1_OCT3}.${IB_1_OCT4}"
```

Then change the `IPADDR='${IB_1_IP}'` and `NETMASK='${IB_1_NETMASK}'` lines to the following:

```
IPADDR='${IB_1_NAS_IP}'  
NETMASK='255.255.192.0'
```

Then add the following to the end of the file:

```
# ib-1-vlan config  
cat << EOF >$iruslot/etc/sysconfig/network/ifcfg-vlan1  
# ifcfg config file for vlan ib1  
BOOTPROTO='static'  
BROADCAST=''  
ETHTOOL_OPTIONS=''  
IPADDR='${IB_1_IP}'  
MTU=''  
NETMASK='255.255.192.0'  
NETWORK=''  
REMOTE_IPADDR=''  
STARTMODE='auto'  
USERCONTROL='no'  
ETHERDEVICE='ib1'  
EOF  
if [ $NAS_IF -eq 0 ]; then  
    rm $iruslot/etc/sysconfig/network/ifcfg-vlan1  
fi
```

To update the `fstab` for the compute blades, edit `/opt/sgi/share/per-host-customization/global/sgi-fstab` file. Perform the equivalent steps as above to add the `# Compute NAS interface to use` section into this file. Then to specify mount points, add lines similar to the following example:

```
# SGI NAS Server Mounts  
${NAS_IPS[${NAS_IF}]}:/mnt/data/scratch /scratch nfs defaults 0 0
```

Creating User Accounts

The example used in this section assumes that the home directory is mounted on the NIS Master service and that the NIS master is able to create directories and files on it as root. The following example use command line commands. You could also create accounts using YaST.

Procedure 2-20 Creating User Accounts on a NIS Server

To create user accounts on the NIS server, perform the following steps:

1. Log in to the NIS Master service node as root.
2. Issue a `useradd` command similar to the following:

```
# useradd -c "Joe User" -m -d /home/juser juser
```

3. Provide the user a password, as follows:

```
# passwd juser
```

4. Push the new account to the NIS servers, as follows:

```
# cd /var/yp && make
```

Tasks You Should Perform After Changing a Rack Leader Controller

If you add or remove a rack leader controller (leader node), for example, if you use `discover` command to discover a new rack of equipment, you will need to configure the new rack leader controller to be an NIS slave server as described in "Setting Up a SLES Service Node as a NIS Client" on page 120.

In addition, you need to add or remove the leader from the `/var/yp/ypservers` file on NIS Master service node. Remember to use the `-ib1` name for the leader, as service nodes cannot resolve `r2lead` style names. For example, use `r2lead-ib1`.

```
# cd /var/yp && make
```

Installing SMC for SGI ICE Patches and Updating SGI ICE Systems

This section describes how to update the software on an SGI ICE system.

Note: To use the Subscription Management Tool (SMT) and run the `sync-repo-updates` script you must register your system with Novell using **Novell Customer Center Configuration**. This is in the **Software** category of YaST (see "Register with Novell " on page 128 and "Configuring the SMT Using YaST" on page 129).

Overview of Installing SMC for SGI ICE Patches

SGI supplies updates to SMC for SGI ICE software via the SGI update server at <https://update.sgi.com/>. Access to this server requires a Supportfolio login and password. Access to SUSE Linux Enterprise Server updates requires a Novell login account and registration.

The initial installation process for the SGI ICE system set up a number of package repositories in the `/tftpboot` directory on the admin node. The SMC for SGI ICE related packages are in directories located under the `/tftpboot/sgi` directory. For SUSE Linux Enterprise Linux 11 (SLES11), they are in `/tftpboot/distro/sles11`.

When SGI releases updates, you may run `sync-repo-updates` (described later) to download the updated packages that are part of a patch. The `sync-repo-updates` command automatically positions the files properly under `/tftpboot`.

Once the local repositories contain the updated packages, it is possible to update the various SGI ICE admin, leader, and managed service node images using the `cinstallman` command. The `cinstallman` command is used for all package updates including those within images, running nodes, including the admin node itself.

There is a small amount of preparation required, in order to setup an SGI ICE system, so that updated packages can be downloaded from the SGI update server and the Linux distro server and then installed with the `cinstallman` command.

The following sections describe these steps, as follows:

- "Update the Local Package Repositories on the Admin Node" on page 127
- "Installing Updates on Running Admin, Leader, and Service Nodes " on page 132

Update the Local Package Repositories on the Admin Node

This section explains how to update the local product package repositories needed to share updates on all of the various nodes on an SGI ICE system.

Mirroring Distribution Updates

In order to keep your system up to date, there are various methods for getting package updates to your SGI ICE system.

SGI has integrated with the distribution update tools for SLES and RHEL. However, this integration only works if the distribution the system admin controller (admin node) is running is the same. For example, it's difficult for a SLES 11 system to get Red Hat updates from RHN. Below, you will find a description of managing package updates when the distro installed on the admin node matches the rest of the system. Finally, some ideas will be presented for SGI ICE systems that have a mix of distributions available.

Update the SGI Package Repositories on the Admin Node

SGI provides a `sync-repo-updates` script to help keep your local package repositories on the admin node synchronized with available updates for the SMC for SGI ICE, SGI Foundation, SGI Performance Suite, and SLES products. The script is located in `/opt/sgi/sbin/sync-repo-updates` on the admin node.

The `sync-repo-updates` script requires your Supportfolio user name and password. You can supply this on the command line or it will prompt you for it. With this login information, the script contacts the SGI update server and downloads the updated packages into the appropriate local package repositories.

For SLES, if you installed and configured the SMT tool as described in "SLES Admin Nodes: Update the SLES Package Repository" on page 128, the `sync-repo-updates` script will also download any updates to SLES from the Novell update server. When all package downloads are complete, the script updates the repository metadata.

Once the script completes, the local package repositories on the admin node should contain the latest available package updates and be ready to use with the `cinstallman` command.

The `sync-repo-updates` script operates on all repositories, not just the selected repository.

Note: You can use the `crepo` command to set up custom repositories. If you add packages to these custom repositories later, you need to use the `yume --prepare --repo` command on the custom repository so that the metadata is up to date. Run the `cinstallman --yum-node --node admin clean all` command and then the `yum/yume/cinstallman` command.

SLES Admin Nodes: Update the SLES Package Repository

In 1.8 (or later), SLES updates are mirrored to the admin node using the SUSE Linux Enterprise Subscription Management Tool. The Subscription Management Tool (SMT) is used to mirror and distribute updates from Novell. SMC for SGI ICE software only uses the mirror abilities of this tool. Mechanisms within SMC for SGI ICE are used to deploy updates to installed nodes and images. SMT is described in detail in the SUSELinux Enterprise *Subscription Management Tool Guide*. A copy of this manual is in the `SMT_en.pdf` file located in the `/usr/share/doc/manual/sle-smt_en` directory on the **admin node** of your system. Use the `scp(1)` command to copy the manual to a location where you can view it, as follows:

```
admin :~ # scp /usr/share/doc/manual/sle-smt_en/SMT_en.pdf user@domain_name.mycompany.com:
```

Register with Novell

Register your system with Novell using **Novell Customer Center Configuration**. This is in the **Software** category of YaST. When registering, use the email address that is already on file with Novell. If there is not one on file, use a valid email address that you can associate with your Novell login at a future date.

The SMT will not be able to subscribe to the necessary update channels unless it is configured to work with a properly authorized Novell login. If you have an activation code or if you have entitlements associated with your Novell login, the SMT should be able to access the necessary update channels.

More information on how to register, how to find activation codes, and how to contact Novell with questions about registration can be found in the YaST help for Novell Customer Center Configuration.

Configuring the SMT Using YaST

At this point, your admin node should be registered with Novell. You should also have a Novell login available that is associated with the admin node. This Novell login will be used when configuring the SMT described in this section. If the Novell login does not have proper authorization, you will not be able to register the appropriate update channels. Contact Novell with any questions on how to obtain or properly authorize your Novell login for use with the SMT.

Procedure 2-21 Configuring SMT Using YaST

Note: In step 8, a window pops up asking you for the Database root password. View the file `/etc/odapw`. Enter the contents of that file as the password in the blank box.

To configure SMT using YaST, perform the following steps:

1. Start up the YaST tool, as follows:

```
admin:~ # yast
```

2. Under **Network Services**, find **SMT Configuration**
3. For **Enable Subscription Management Tool Service (SMT)**, check the box.
4. For **NU User**, enter your Novell user name.
5. For **NU Password**, enter your Novell password.

Note: It is the mirror credentials you want. You can have a login that gets updates but cannot mirror the repository.

6. For **NU E-Mail**, use the email with which you registered.
7. For your **SMT Server URL**, just leave the default.

It is a good idea to use the test feature. This will at least confirm basic functionality with your login. However, it does not guarantee that your login has access to all the desired update channels.

Note that **Help** is available within this tool regarding the various fields.

8. When you click **Next**, a window pops up asking for the Database root password. View the file `/etc/odapw`. Enter the contents of that file as the password in the blank box.

A window will likely pop up telling you that you do not have a certificate. You will then be given a chance to create the default certificate. Note that when that tool comes up, you will need to set the password for the certificate by clicking on the certificate settings.

Setting up SMT to Mirror Updates

This section describes how to set up SMT to mirror the appropriate SLES updates.

Procedure 2-22 Setting up SMT to Mirror Updates

To set up SMT to mirror updates, from the admin node, perform the following steps:

1. Refresh the list of available catalogs, as follows:

```
admin:~ # smt-ncc-sync
```

2. Look at the available catalogs, as follows:

```
admin:~ # smt-catalogs
```

In that listing, you should see that the majority of the catalogs matching the admin node distribution (distro) **sles11** have "Yes" in the "**Can be Mirrored**" column.

3. Use the `smt-catalogs -m` command to show you just the ones that you are allowed to mirror.
4. From the **Name** column, choose the entities with the ending of **-Updates** matching channels matching the installed distro. For example, if the base distro is SLES11, you might choose:

```
SLE11-SMT-Updates  
SLE11-SDK-Updates  
SLES11-Updates
```

5. This step shows how you might enable the catalogs. Each time, you will be presented with a menu of choices. Be sure to select only the **x86_64** version and if given a choice between **sles** and **sled**, choose **sles**, as follows:

```
admin:~ # smt-catalogs -e SLE11-SMT-Updates  
admin:~ # smt-catalogs -e SLE11-SDK-Updates
```

```
admin:~ # smt-catalogs -e SLES11-Updates
```

In the example, above, select 7 because it is x86_64 and sles, the others are not.

6. Use the `smt-catalogs -o` command to only show the enabled catalogs. Make sure that it shows the channels you need to be set up for mirroring.



Warning: SMC for SGI ICE does not map the concept of channels on to its repositories. This means that any channel you subscribe to will have its RPMs placed into the distribution repository. Therefore, only subscribe the SMC for SGI ICE admin node to channels related to your SMC for SGI ICE cluster needs.

Downloading the Updates from Novell and SGI

At this time, you should have your update channels registered. From here on, the `sync-repo-updates` script will do the rest of the work. That script will use SMT to download all the updates and position those updates in to the existing repositories so that the various nodes and images can be upgraded.

Run `/opt/sgi/sbin/sync-repo-updates` script.

After this completes, you need to update your nodes and images (see "Installing Updates on Running Admin, Leader, and Service Nodes " on page 132).

Note: Be advised that the first sync with the Novell server will take a very long time.

RHEL Admin Nodes: Update the RHEL Package Repository

This section describes how to keep your packages up to date on RHEL based admin nodes. The general idea is that we download all updates in to the RHEL repository, and then use SGI Management Center for ICE tools to deploy the updates to nodes and images.

Perform the following:

- Register with RHN. This can be done, as follows:

```
# rhn_register
```

- Once registered, the `sync-repo-updates` command will synchronize the latest version of update packages in to the RHEL 6 repository on the system.

```
# sync-repo-updates
```

Update Distros That Do Not Match the Admin Node

In situations where you have distributions (distros) present that do not match the distro on installed on the admin node, you will have to arrange to download the updates on your own.

SLES

The instructions provided earlier show how to set up Novell SMT for the admin node. You could use similar ideas to configure your own SMT server somewhere on your network. Once the RPMs are staged on that server, you can copy them to the admin node using `rsync` or some other similar transport method. Remember to update the repository metadata after you update the packages. For example:

```
# yume --prepare --repo /tftpboot/distro/sles11sp1
```

RHEL

You can register with RHN on a RHEL server on your network. Then, you can look at the `/opt/sgi/sbin/sync-repo-updates` script to see how it stages the packages (search for `RHN` in that file). Following that example, you can set up a server on your house network to stage the files. Then you can then copy the staged packages to the admin node in to the matching distro repository. You need to update the repository metadata after copying packages using `yume` in a way similar to this:

```
# yume --prepare --repo /tftpboot/distro/rhel6.0
```

Note: You can always make a managed service node provide the function of staging the updates.

Installing Updates on Running Admin, Leader, and Service Nodes

This section explains how to update existing nodes and images to the latest packages in the repositories.

To install updates on the admin node, perform the following command from the admin node:

```
admin:~ # cinstallman --update-node --node admin
```

To install updates on all online leader nodes, perform the following command from the admin node:

```
admin:~ # cinstallman --update-node --node r\*lead
```

To install updates on all managed and online service nodes, perform the following from the admin node:

```
admin:~ # cinstallman --update-node --node service\*
```

To install updates on the admin, all online leader nodes, and all online and managed service nodes with one command, perform the following command from the admin node:

```
admin:~ # cinstallman --update-node --node \*
```

Please note the following:

- The `cinstallman` command does not operate on running compute nodes. For compute nodes, it is an image management tool only. You can use it to create and update compute images and use the `cimage` command to push those images out to leader nodes (see "cimage Command" on page 206).

For managed service nodes and leader nodes, you can use the `cinstallman` command to update a running system, as well as, images on that system.

- When using a node aggregation, for example, the asterisk (*), as shown in the examples above, if a node happens to be unreachable, it is skipped. Therefore, you should ensure that all expected nodes get their updated packages.
- For more information on the `crepo` and `cinstallman` commands, see "crepo Command" on page 191 and "cinstallman Command" on page 195, respectively.

Bittorrent Provisioning

The SMC 1.5 release adds support for bittorrent provisioning of service nodes and rack leader controllers (RLCs, also called leader nodes for short).

By default, bittorrent tarballs are still created for compute node images. The bittorrent transport is used to transport the compute images to the RLCs (leader nodes) the first

time the image is pushed to a given RLC. While this is still done by default, you can disable the creation and storing of the compute node bittorrent tarball using the `cinstallman` command.

By default, service and RLC (leader) images will not have a bittorrent tarball created. However, you can enable the tarball using `cinstallman`. For example:

```
# cinstallman --add-to-bt --image service-rhel6.1
```

You can remove the tarball with a command similar to the following:

```
# cinstallman --del-from-bt --image service-rhel6.1
```

If you want to re-install a node or set of nodes and make use of the bittorrent transport, you can use `cinstallman` to use bittorrent installation next time the said nodes reboot, as follows:

```
# cinstallman --next-boot bt --node service\*
```

Consequently, after the service nodes are rebooted, they will install themselves. They will make use of bittorrent during installation.

You can also direct that bittorrent be used when discovering a node. See the `discover` command and the "bt=yes" specifier.

When should I use bittorrent transport?

Note: It does not make sense to use bittorrent, when just a couple nodes are being installed. A bittorrent installation will actually take longer for an individual node because a tarball has to be extracted in the process. However, when you start getting in to more than five nodes, bittorrent may start to make sense to use.

Why not use multicast?

Note: The SGI ICE HEAD network design in SGI ICE 8400 systems, and previous machines, is such; that getting the multicast timing right can be a problem on larger systems. You may know that multicast is used for transporting the root tarball to compute nodes when the compute nodes are set up to use `tmpfs` roots. In this usage case, all the compute nodes are behind a set of CMCs that are stacked together and hence are not susceptible to multicast timing issues.

Updating Packages Within Systemimager Images

You can also use the `cinstallman` command to update `systemimager` images with the latest software packages.

Note: Changes to the kernel package inside the compute image require some additional steps before the new kernel can be used on compute nodes (see "Additional Steps for Compute Image Kernel Updates" on page 136 for more details). This note does **not** apply to leader or managed service nodes.

The following examples show how to upgrade the packages inside the three node images supplied by SGI:

```
admin:~ # cinstallman --update-image --image lead-sles11
admin:~ # cinstallman --update-image --image service-sles11
admin:~ # cinstallman --update-image --image compute-sles11
```

Note: Changes to the compute image on the admin node are not seen by the compute nodes until the updates have been pushed to the leader nodes with the `cimage` command. Updating leader and managed service node images ensure that the next time you add or re-discover or re-image a leader or service node, it will already contain the updated packages.

Before pushing the compute image to the leaders using the `cimage` command, it is good idea to clean the `yum` cache.

Note: The `yum` cache can grow and is in the writable portion of the compute blade image. This means it is replicated 64 times per compute blade image per rack and the space that may be used by compute blades is limited by design to minimize network and load issues on rack leader nodes.

To clean the `yum` cache, from the system admin controller (admin node), perform the following:

```
admin:~ # cinstallman --yum-image --image compute-sles11 clean all
```

Additional Steps for Compute Image Kernel Updates

Any time a compute image is updated with a new kernel, you will need to run some additional steps in order to make the new kernel available. The following example assumes that the compute node image name is `compute-sles11` and that you have already updated the compute node image in the image directory per the instructions in "Creating Compute and Service Node Images Using the `cinstallman` Command" on page 211. If you have named your compute node image something other than `compute-sles11`, replace this in the example that follows:

1. Shut down any compute nodes that are running the `compute-sles11` image (see "Power Management Commands" on page 220).
2. Push out the changes with the `cimage --push-rack` command, as follows:

```
admin:~ # cimage --push-rack compute-sles11 r\*
```

3. Update the database to reflect the new kernel in the `compute-sles11`, as follows:

```
admin:~ # cimage --update-db compute-sles11
```

4. Verify the available kernel versions and select one to associate with the `compute-sles11` image, as follows:

```
admin:~ # cimage --list-images
```

5. Associate the compute nodes with the new kernel/image pairing, as follows:

```
admin:~ # cimage --set compute-sles11 2.6.16.46-0.12-smp "r*i*n"
```

Note: Replace `2.6.16.46-0.12-smp` with the actual kernel version.

6. Reboot the compute nodes with the new kernel/image.

Cascading Dual-Boot

This section describes cascading dual-root (multiple root) support. This adds the notion of a "root slot" that represents a `/`(root directory) and `/boot` directory pair for a certain operating system. The layout and usage is described in the section that follows.

Partition Layout for Admin, Leader, and Service Nodes with Multiroot

Only the leader node has XFS root filesystems. Partition layout for more than one slot is shown in Table 2-1 on page 137.

Table 2-1 Partition Layout for Multiroot

Partition	Filesystem Type	Filesystem Label	Notes
1	swap	sgiswap	Partition Layout: Multiroot
2	ext3	sgidata	SGI Data Partition, MBR boot loader for admin nodes
3	extended	N/A	Extended partition, making logicals out of the rest of the disk
5	ext3	sgiboot	/boot partition for slot 1
6	ext3 or xfs	sgiroot	/partition for slot 1
7	ext3	sgiboot	/boot partition for slot 2 (optional)
8	ext3 or xfs	sgiroot	/ partition for slot 2

Table 2-1 on page 137 shows a partition table with two available slots. SMC for SGI ICE supports up to five available slots. After five slots, partitions are not available to support the slot.

Partition Layout for a Single Root

Partition layout for a single root is shown in Table 2-2 on page 138. Partition layout for single slot is the same layout that leader and service nodes have used previously. Legacy leader/service node layout is used for single slot, in order to generate the correct pxelinux chainload setup. Previously, the MBR bootloader was used. For multiroot, a chainload to a root slot boot partition is used.

Table 2-2 Partition Layout for Single Root

Partition	Filesystem Type	Filesystem Label	Notes
1	ext3	sgiboot	/boot
2	extended	n/a	Extended partition, making logicals out of the rest of the disk
5	swap	sgiswap	Swap partition
6	ext3 or xfs	sgiroot	/

Prior to 1.6 release, admin nodes had a different partition layout than either shown in Table 2-1 on page 137 or Table 2-2 on page 138. It had two partitions: swap and a single root. No separate /boot. Any newly installed admin node will have one of the two partition layouts described in the tables above. However, since admin nodes can be upgraded as opposed to re-installed, you may have one of three different partition layouts for admin nodes.

Admin Node Installation Choices Related to Cascading Dual-Boot

When you boot the admin node installation DVD, you are brought to a `syslinux` boot banner by default with a boot prompt, as in previous releases.

The multiroot feature support adds a few new parameters, as follows:

- `re_partition_with_slots`

The `re_partition_with_slots` boot parameter is used to specify that the admin node system drive should be partitioned for the SMC for SGI ICE multiroot

feature and to specify the number of SMC for SGI ICE slots to create when partitioning the admin node system drive. A maximum of five SMC for SGI ICE slots is supported.

For example, if the admin node system drive is determined to be blank, then the following specifies that the admin node system drive should be partitioned with five SMC for SGI ICE slots:

```
re_partition_with_slots=5
```

The number of slots into which the admin node system drive should be partitioned should be such that $1 \leq x \leq 5$.

- `destructive`

If the admin node root drive is determined to be non-blank, then the `re_partition_with_slots` operation is rejected unless the following boot parameter is also set:

```
destructive=1
```

If `destructive=1`, potentially destructive operations are allowed.

- `install_slot`

The `install_slot` boot parameter specifies the SMC for SGI ICE slot into which the first installation should go. By default, when creating new slots on the admin node system drive, the first installation will go in SMC for SGI ICE slot 1. For example, if the first installation is to go in slot 2, specify:

```
install_slot=2
```

Note: If the boot parameter `install_slot` is used to direct the install into a slot other than default slot 1, then be sure to select that slot during the first boot of the admin node.

After the boot completes, use the `cadmin` command to set the slot in which the initial install occurred to the default slot, for example slot 2:

```
# cadmin --set-default-root --slot 2
```

If the install slot is not selected during the first boot of the admin node, then the node will attempt to boot from empty slot 1, and will fail to boot. Reset the admin node to restart the boot, and select the install slot during the boot.

- **Multiroot installation situations**

Several situations that can be encountered when partitioning the admin node system drive for SMC for SGI ICE multiroot support:

- If an admin node is encountered with exactly one blank/virgin disk, and no multiroot parameters are provided, then the admin node will be partitioned for two SMC for SGI ICE slots and the initial installation will be to slot 1.
- If an admin node is encountered with more than one blank/virgin disk, a protection mechanism triggers and the installer errors out because we are not sure which disk to choose.
- If an admin node is encountered with a disk previously used for SMC for SGI ICE, nothing destructive will happen unless the `destructive=1` parameter is passed.
- If an `install_slot` is specified that appears to have been used for previously, it will not be repartitioned unless `destructive=1` is supplied.

Leader and Service Node Installation

Leader and service nodes are installed, as previously. However, they mimic the admin node in terms of partition layout and which slot is used for what purpose.

Therefore, when a `discover` operation is performed, the slot used for installation is the same slot on which the admin node is currently booted. So you cannot choose what goes where, currently, it all matches the admin node.

If the leader or service node is found to have a slot count that does not match the admin node, the node is re-partitioned. It is assumed if the admin node changes its layout, all partitions on leaders and service nodes are re-initialized as well.

Choosing a Slot to Boot the Admin Node

After the admin node is installed with 1.8 (or later), it will boot one of two ways. If only one root slot is configured, the MBR of the admin node will be used to boot the root as usual.

However, if more than one root slot is selected, then the grub loader in the MBR will direct you to a special `grub` menu that allows you to choose a root slot.

For the multi-root admin node, the `sgidata` partition is used to store some `grub` files and `grub` configuration information. Included is a chainload for each slot. Therefore, the first `grub` to come up on the admin node chooses between a list of slots. When a slot is selected, a chainload is performed and the `grub` representing that slot comes up.

How to Handle Resets, Power Cycles, and BMC `dhcp` Leases When Changing Slots

This section describes how to handle resets, power cycles, and BMC `dhcp` leases when changing slots, as follows:

- Prior to rebooting the admin node to a new root slot, you should shut down the entire cluster including compute blades, leader nodes, and service nodes. If you use the `cpower` with the `--shutdown` option, the managed leader and service nodes will be left in a single user mode state. An example `cpower` command is, as follows:

```
admin:~ # cpower --shutdown --system
```

- After this is complete, reboot the admin node and boot the new slot.
- After the admin node comes up on its new slot, you should use the `cpower` command to reboot all of the leader and service nodes. This ensures that they reboot and become available. An example `cpower` command is, as follows:

```
admin:~ # cpower --reboot --system
```

Note: In some cases, the IP address setup in one slot may be different than another. This problem can potentially affect leader and service node BMCs. After the admin node is booted up in to a new slot, it is possible the BMCs on the leaders and service nodes may have hung on to their old IP addresses. They will eventually time-out and grab new leases. This problem may manifest itself in `cpower` not being able to communicate with the BMCs properly. If you have trouble connecting to leader and service node BMCs after switching slots on the admin node, give the nodes up to 15 minutes to grab a new leases that match the new slot.

Leader and Service Node Booting

The way leader and service nodes boot is dependent on whether the cascading dual-boot feature is in use or not, as explained in this section.

Leader and Service Node Booting on a System Configured with One Root Slot

When a system is configured with only one root slot, it is not using the cascading dual-boot feature. This may be because you want all the disk space on your nodes dedicated to a single installation, or it may be because you have upgraded from previous SMC for SGI ICE releases that did not make use of this feature and you do not want to reinstall at this time.

When not using the cascading dual-boot feature, the admin node creates PXE configuration files that direct the service and leader nodes to do one of the following:

- Boot from their disk
- Boot over the network to reinstall themselves; if set up to re-image themselves by the `cinstallman` command (see "cinstallman Command" on page 195) or by initial discovery with the `discover` command (see "discover Command" on page 80).

Leader and Service Node Booting on a System Configured with Multiple Roots Slots

When a system is configured with two or more root slots, it is using the cascading dual-boot feature.

In this case, the admin node creates leader and service PXE configuration files that direct the managed service and leader nodes to do one of the following:

- Boot from the currently configured slot
- Reinstall the currently configured slot

Which slot is current, is determined by the slot on which the admin node is booted. Therefore, the admin node and all managed service and leader nodes are always booted on the same slot number.

In order to configure a managed service or leader node to boot from a given slot, the admin node creates a PXE configuration file that is configured to load a chainloader. This chainloader is used to boot the appropriate boot partition of the managed service or leader node.

This means that, in a cascading dual-boot situation, the service and leader nodes do not have anything in their master boot record (MBR). However, each `/boot` has `grub` configured to match the associated root slot. A `syslinux` chainload is performed by PXE to start `grub` on the appropriate boot partition.

If, for some reason, a PXE boot fails to work properly, there will be no output at all from that node. This means that cascading dual-boot is heavily dependent on PXE boots working properly for its operation.

Note: Unlike the managed service and leader nodes, the admin node always has an MBR entry. See "Choosing a Slot to Boot the Admin Node" on page 140.

Slot Cloning

A script named `/opt/sgi/sbin/clone-slot` is available. This script allows you to clone a source slot to a destination slot. It then handles synchronizing the data and fixing up `grub` and `fstabs` to make the cloned slot a viable booting choice.

The script sanitizes the input values, then calls a worker script in parallel on all managed nodes and the admin node that does the actual work. The `clone-slot` script waits for all children to complete before exiting.

Important: If the slot you are using as a source is the mounted/active slot, the script will shut down `mysql` on the admin node prior to starting the backup operation and start it when the backup is complete. This ensures there is no data loss.

Admin Node: Managing Which Slot Boots by Default

Use the `cadmin` command to control which slot on the admin node boots by default.

To show the slot that is currently the default, perform the following:

```
admin:~ # cadmin --show-default-root
```

To change it so slot 2 boots by default, perform the following:

```
admin:~ # cadmin --set-default-root --slot 2
```

Admin Node: Managing Grub Labels

You can use the `cadmin` command to control the `grub` labels the various slots have. When a slot is installed, the label is updated to be in this form:

```
slot 1: SMC for SGI ICE 2.0 / sles11: (none)
```

You can adjust the last part (none in the above example). The following are some example commands.

Show the currently configured `grub` root labels, as follows:

```
admin:~ # cadmin --show-root-labels
```

Set the customer-adjustable portion of the root label for slot 1 to say "life is good", as follows:

```
admin:~ # cadmin --show-root-labels
slot 1: SMC for SGI ICE 2.0 / sles11: first sles
slot 2: SMC for SGI ICE 2.0 / sles11: my party
slot 3: SMC for SGI ICE 2.0 / sles11: I can cry if I want to.
slot 4
admin:~ # cadmin --set-root-label --slot 1 --label "life is good"
admin:~ # cadmin --show-root-labels
slot 1: SMC for SGI ICE 2.0 / sles11: life is good
slot 2: SMC for SGI ICE 2.0 / sles11: my party
slot 3: SMC for SGI ICE 2.0 / sles11: I can cry if I want to.
slot 4
```

Admin Node: Which root slot is in use?

You can use the `cadmin` command to show the root slot you are currently booted in to on the admin node, as follows:

```
admin:~ # cadmin --show-current-root
admin node currently booted on slot: 2
```

Installing MPI on a Running SGI ICE System

This section describes how to install MPI on an SGI ICE system that has already been installed. The instructions in this section update existing images instead of creating new ones. It should be noted that integrating MPI before cluster deployment is easier.

SGI supplied media, such as SGI® MPI and SGI® Accelerate™ CDs, have embedded in them suggested package lists for each node type. The `crepo` command, used in the following example, makes use of these lists and indeed recomputes the lists when new media is added and then selected.

File names in this example are just illustrations.

Register SGI MPI and SGI Accelerate with SMC, as follows:

```
# crepo --add accelerate-1.0-cd1-media-rhel6-x86_64.iso
# crepo --add mpi-1.0-cd1-media-rhel6-x86_64.iso
```

Update the `crepo` selected repositories so that all repositories associated with the software distribution (distro) you are installing for are present. For example, if you want MPI to work on RHEL 6, you might do something like this:

Show what is currently selected (the asterisks to the left):

```
# crepo --show
SUSE-Linux-Enterprise-Server-10-SP3 : /tftpboot/distro/sles10sp3
* SGI-2.2-rhel5 : /tftpboot/sgi/SGI-Tempo-2.2-rhel5
  SGI-XFS-XVM-2.2-for-RHEL-rhel6 : /tftpboot/sgi/SGI-XFS-XVM-2.2-for-RHEL-rhel6
* SGI-Foundation-Software-1SP6-rhel5 : /tftpboot/sgi/SGI-Foundation-Software-1SP6-rhel5
  SGI-Foundation-Software-2.2-rhel6 : /tftpboot/sgi/SGI-Foundation-Software-2.2-rhel6
  SGI-Tempo-2.2-sles10 : /tftpboot/sgi/SGI-Tempo-2.2-sles10
  Red-Hat-Enterprise-Linux-6.0 : /tftpboot/distro/rhel6.0
  SGI-MPI-1.0-rhel6 : /tftpboot/sgi/SGI-MPI-1.0-rhel6
  SGI-Foundation-Software-1SP6-sles10 : /tftpboot/sgi/sgifoundationsoftware1sp6-sles10
* Red-Hat-Enterprise-Linux-Server-5.5 : /tftpboot/distro/rhel5.5
  SGI-Management-Center-1.2-rhel6 : /tftpboot/sgi/SGI-Management-Center-1.2-rhel6
  SGI-Tempo-2.2-rhel6 : /tftpboot/sgi/SGI-Tempo-2.2-rhel6
  SGI-Accelerate-1.0-rhel6 : /tftpboot/sgi/SGI-Accelerate-1.0-rhel6
```

Unselect unrelated repositories:

```
# crepo --unselect SGI-Tempo-2.2-rhel5
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel5.5.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel5.5.rpmlist
# crepo --unselect SGI-Foundation-Software-1SP6-rhel5
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel5.5.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel5.5.rpmlist
# crepo --unselect Red-Hat-Enterprise-Linux-Server-5.5
Removing: /etc/opt/sgi/rpmlists/generated-compute-rhel5.5.rpmlist
Removing: /etc/opt/sgi/rpmlists/generated-service-rhel5.5.rpmlist
```

Select RHEL 6 related repositories:

```
# crepo --select Red-Hat-Enterprise-Linux-6.0
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.0.rpmlist
# crepo --select SGI-Foundation-Software-2.2-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.0.rpmlist
# crepo --select SGI-XFS-XVM-2.2-for-RHEL-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.0.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.0.rpmlist
```

After performing the steps, above, the proper repositories are registered and selected so you can operate on them by default. Since you are using an already deployed system, you need to update existing images and potentially existing service nodes themselves. This example uses SGI suggested/ generated rpmlists. If you have custom rpmlists, you need to manually reconcile the two lists for each node type. The list fragments in `/var/opt/sgi/sgi-repodata/` may help you.

For a service node image, perform the following:

```
# cinstallman --refresh-image --image service-rhel6.0 --rpmlist /etc/opt/sgi/rpmlists/generated-service-rhel6.0.rpmlist
```

For a compute node image, perform the following:

```
# cinstallman --refresh-image --image compute-rhel6.0 --rpmlist /etc/opt/sgi/rpmlists/generated-compute-rhel6.0.rpmlist
```

Finally, you need to push the updated compute image to the leaders.

Note: If the compute nodes are booted on the image and are using NFS for roots, you need to shut the compute nodes down before being able to run this command.

```
# cimage --push-rack compute-rhel6.0 r"*"
```

To make sure the compute nodes you are operating on have the associated compute image you just updated, perform a command similar to the following”

```
# cimage --set compute-rhel6.0 2.6.32-71.el6.x86_64 ""
```

You can find the available images and kernels using the `cimage --list-images` command.

If you have booted service/login nodes, you likely want to refresh those running nodes also. (You could also reinstall them, as well). Here is a refresh example:

```
# cinstallman --refresh-node --node service0 --rpmfile /etc/opt/sgi/rpmlists/generated-service-rhel6.0.rpmfile
```

Now reset or bring up the nodes (depends on the state you left them). If you want to bring up all nodes, this command will not disrupt nodes already operating:

```
# cpower --system --up
```

System Admin Controller High Availability Solution

This section describes the system admin controller (admin node) high availability solution.

Overview

The system admin controller (admin node) high availability (HA) solution makes use of qemu-KVM virtual machines managed by `libvirt` virtualization API and the high availability (HAE) components supplied by the SLES HAE to provide a high availability environment.

The two physical machines form an HA framework to host a single virtual machine. This virtual machine acts as the system admin controller (admin node) for the SGI ICE system.

If one physical machine fails, the other physical machine starts up the virtual machine and the admin node is brought back up.

It is important to read the release notes for the latest information, known problems, and workarounds needed to configure the system properly. In a scenario, such as HA, where one node can reset the other, it is all the more important to read the release notes prior to attempting the installation.

Hardware Requirements

The major hardware components that comprise this solution include the following:

- Two system admin controllers (admin nodes) with the following components
 - Add-in card with two or more Ethernet ports
 - Integrated system disks
 - Serial Attached SCSI (SAS) card for external storage
- SGI InfiniteStorage 5000 (IS5000)

For a functional diagram of system admin controller (admin node) high availability (HA) solution, see Figure 2-47 on page 148.

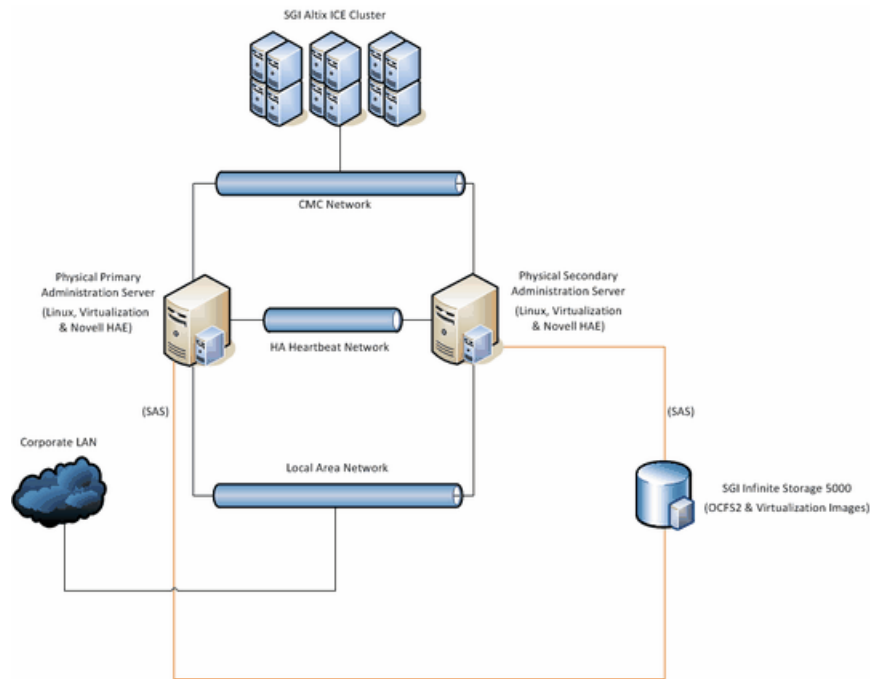


Figure 2-47 Admin Node High Availability Solution Functional Diagram

Physical Connections

The network Ethernet connections are shown in Figure 2-48 on page 149.

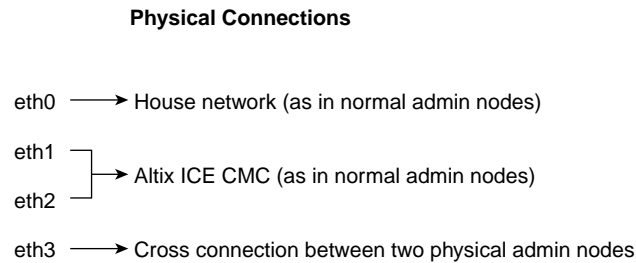


Figure 2-48 HAE Network Ethernet Connections

Note: In some configurations, Ethernet switches may be necessary between the SGI ICE CMCs and the admin nodes due to port availability.

Software Requirements

The major software components that comprise this solution include the following:

- SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP1) base distribution
- SLES 11 SP1 High Availability Extension (HAE)
- The `vm-install` package
- The `sac-ha` package from SGI, available on SGI Management Center media
- SAC Admin Install DVD ISO from SGI

An overview of how to install the system admin controller (admin node) high availability solution is, as follows:

- Perform a default installation of SLES11 SP1 on both physical admin nodes. A later step will enable the firewall in a way that will work with HAE.
- Be sure to disable the firewall on both physical machines.
- Then install HAE and all available SLES and HAE updates from Novell, Inc.

- Install the `vm-install` package on both physical admin nodes.
- Install the `sac-ha` package from SGI on both physical admin nodes.
- Pick an admin node to be primary and configure `sac-ha` to run the helper script.

For a detailed description of this procedure, see "HAE, Virtualization and Updates installation" on page 151.

Physical Node BMC Configuration

This configuration makes use of STONITH technique, that is, "Shoot The Other Node In The Head". This means that the physical nodes need to access each other's BMCs. You need to configure the BMCs to be responsive on your house network including having a host name and knowing the login/password for the admin account in the BMC. You should do this early in the process, it will be required in a later step.

External Storage Configuration

You need to configure your external storage. The external storage must have a SAS connection to both physical admin nodes (see Figure 2-47 on page 148). You must make available at least one LUN and that this LUN needs to appear as a disk to the physical admin nodes.

Base SLES 11 SP1 Installation Considerations

This section provides some general instructions about SLES 11 SP1 when installing the admin nodes, as follows:

- Boot the SLES 11 SP1 installer the way you normally would at your site.
- When asked for a scenario, choose **Physical Machine** (Also for **Fully Virtualized Guests**) for both physical admin nodes.
- Be sure to use the root device inside the chassis, and **not** the external storage for the installation. The external storage will only house virtual machine images in a future step. If you see any storage named SGI-IS400 or similar, that is external storage. If you see storage related to LSILOGIC or Megaraid, that is internal storage. You want to use the root device on internal storage.

- When selecting disks, use the entire hard disk. Default values for root and swap should be sufficient.
- Install just the default SLES11 SP1 packages for now on both physical admin nodes.

Network Configuration

Configure the interface connected to your house network as you normally would. Do not use DHCP but rather a static configuration.

A later step will reconfigure networking and renumber the Ethernet devices to match the necessary layout. It will be told to use the interface you configure now as the reference when it reconfigures networking on the physical admin nodes.

High Availability Extension Documentation

For information on the SUSE Linux Enterprise High Availability Extension (HAE) software package, see the SUSE *High Availability Guide*. This is required reading and the information contained in this guide is not replicated in this section.

On the physical admin nodes, the document can be found at
`/usr/share/doc/manual/sleha-guide_en-pdf`.

It is available in the `sleha-guide_en-pdf` package from Novell, Inc.

HAE, Virtualization and Updates installation

This section describes the High Availability Extension (HAE), virtualization, and updates installation, as follows:

- Add the HAE add-on product using SLES11 SP1 tools.
- Install the HAE defaults with `yast`. If using `zypper`, perform the following:

```
# zypper install -t pattern ha_sles  
# zypper install sleha-guide_en-pdf
```
- Install the necessary QEMU/KVM components. If using `zypper`, perform the following:

```
# zypper install virt-manager libvirt kvm vm-install libvirt-cim virt-viewer bridge-utils
```

- Be sure to register with Novell Customer Configuration Center using YaST if you have not already. This should add update sources to your repository (repo). This is a critical step.
- Confirm that the update repos are configured as well. Use the `zypper -lr` command. It should list the following four repos:

```
sles11sp1
sles11sp1 updates
hae
hae updates
```

- Update the system. This is a critical step because the functionality depends on updates being applied. You can do this with the `zypper` command, as follows:

```
# zypper -n update
```

At this point, both physical admin nodes have the exact same software stack installed including SLES11 SP1, HAE, and all updates applied.

Reboot both admin nodes so that you are running with new kernels, updates, and so on.

Before proceeding to the next section, confirm that both physical admin nodes are now up and running and both with their house network connections configured.

Install SGI Helper Scripts (`sac-ha` Package)

From the SGI Management Center media, copy the RPM named `sac-ha` from the media on to both physical admin nodes.

Install the package on both physical admin nodes. This provides the scripts to configure both systems.

Configure the SGI Helper Scripts

On the physical admin node that you want to use as primary, edit the file `/etc/opt/sgi/sac-hae-initial-setup.conf`.

On the second physical admin node, no configuration is necessary. Configuration is all done from the first admin node.

Make sure you read all of the comments in this file. The comments provide the instructions for filling out the file, which include:

- Instructions for ordering the MAC addresses
- Instructions for locating physical Ethernet ports if you are unsure how they are laid out.
- Instructions for locating and defining your shared external storage
- Instructions for filling out all the network information that is required
- You will need to know the SGI ICE Head Network you intend to use in advance. Most sites do not change from the default so the examples provided should be sufficient.

Once the configuration file is filled out entirely, you can run `/opt/sgi/sbin/sac-hae-initial-setup` command.

This script will reconfigure networking on both physical nodes and then will do the HAE setup, and so on. You will be prompted a few times in the early part of the process. These prompts are related to getting the `ssh` keys and known hosts configuration set up and will involve entering the root password and/or the word **yes** as prompted. After `ssh` is set up, the rest of the scripts should run automated without input.

There are places where this process can fail. It is safe to rerun the scripts if they fail in the network set up phase. However, once HAE components are configured, reconfiguring networking can result in machines resetting themselves or each other for protection.

If you do restart the scripts to configure networking, pay special attention to keeping the `phys1_current_house` and `phys2_current_house` variables up to date in the `/etc/opt/sgi/sac-hae-initial-setup.conf` file. They should always point to the currently configured house interface as it preserves the network settings and applies them to the proper interfaces.

For information about shared disk device selection, see the comments in `/etc/opt/sgi/sac-hae-initial-setup.conf` regarding choosing the shared storage device.

The `/etc/opt/sgi/sac-hae-initial-setup.conf` file describes how to choose MAC addresses. It is important that you work out the MAC address ordering as described as the network will be reconfigured on both physical admin nodes to

match. See the configuration file's notes about using `ethtool` to help you figure out which physical port matches a MAC address.

Accessing `crm_gui` and the `hacluster` Account

By default, the SGI SAC HA scripts set the password for the `hacluster` account to `sgisgi`. You will need this password if you run the `crm_gui` tool and wish to log in as the default `hacluster` user in that tool.

Virtual Machine Installation

After the `sac-hae-initial-setup` script completes, the virtual machine for the admin node should be running on one of the two physical machines. You can identify which one by using the `crm_gui` command and looking at the `virt-clone` resource (or similarly use the `crm` command). If it is not started for some reason, go ahead and start the service.

Next, run the `virt-manager` command on the physical machine that is currently hosting the guest.

It should be at the Admin Node Installation DVD prompt. From this point forward, installation of SGI Management Center ICE is the same as it is on real hardware.

Allocate Physical Node IP Addresses with SGI Management Center

After the admin node virtual machine has been installed and the execution of the `configure-cluster` command is complete, you need to allocate the IP addresses used by the physical admin nodes to avoid an IP address conflict. This should be done prior to discovering any nodes.

In the example below, `phys1` is the HEAD network IP for physical machine 1 and `phys2` is for physical machine 2. The IP addresses provided match the IP address used with the SAC HA configuration scripts you ran earlier:

```
# discover --service 98,other
# cadmin --set-ip --node service98 --net head service98=172.23.254.253
# cadmin --set-hostname --node service98 phys1

# discover --service 99,other
# cadmin --set-ip --node service99 --net head service99=172.23.254.254
# cadmin --set-hostname --node service99 phys2
```

High Availability Customization

The `sac-ha-initial-setup` script sets up a basic default set of high availability (HA) rules for the cluster. They are meant to be a starting point for your site.

SGI will likely come up with additional suggested HA rules and configuration details over time. SGI will update the `sac-hae-initial-setup.conf` file for new releases but since that file is only used once ever, you could miss out on configuration updates.

SGI intends to also release HA configuration suggestions as service bulletins available on SGI SupportFolio.

SMC Licenses

Many of the components in SGI Management Center require a license. The license is associated with the virtual machine, and not the physical machine. For this reason, you will need to request the license based on the virtual machine environment once SGI Management Center ICE is installed.

Virtual Admin Guest Service

The default HAE rules supplied by SGI will ensure a virtual machine named "sac" is always running. If the machine shuts down, it will be restarted on one of the physical machines.

The machine can be accessed with the `virt-manager` command from the physical machine.

In to avoid unexpected results, it is best to use `crm_gui` or `crm` to stop the `virt-clone` service when you wish to shut down the admin node virtual machine. Likewise, use `crm` or `crm_gui` to turn the service back on when desired. If you use `virt-manager` to shut the machine down, HAE will turn it back on for you automatically.

Virtual Admin Node Resource

The default `csync2` configuration set up by SGI will keep your admin node virtual machine configuration up to date on both nodes. However, `csync2` is not automatic.

Therefore, if you make adjustments to the virtual machine configuration, such as CPU or memory use, you should run `csync2 -xv` to synchronize to the other physical admin node.

Some sites may prefer a more targeted approach to the `csync2` configuration as it relates to the `libvirt` configuration files.

One thing to keep in mind is that you need the virtual machine configuration files to be the same on both physical admin nodes because the guest needs to run on both. This also helps facilitate live migration of the running virtual machine (VM) from one physical host to the other.

For more information on `csync2`, see SUSE Linux Enterprise High Availability Extension (HAE) *High Availability Guide*.

Special Virtual Machine Guest Configuration: SGI Emulator Wrapper Script

The SGI configuration helper scripts set up the virtual machine guest for you in the correct manner. There is one thing the script puts in the XML configuration file that is hard to specify in `virt-manager`. This is the `<emulator>` tag.

If you create your own virtual machine with `virt-manager`, it will not use the `/opt/sgi/sbin/sgi-emulator-wrapper` wrapper script but instead have a path to `qemu-kvm`.

It is important that the machine use the SGI `sgi-emulator-wrapper` wrapper script because this script provides proper serial number information to the virtual machine. Currently, this information is not used for license checks but is used by ESP to identify the serial number for your overall system.

System admin controllers (admin nodes) have a special system-wide SGI ICE serial number embedded in them that ESP uses. So the simple wrapper script for the emulator passes on this serial number information to the guest admin node.

Migrating a Virtualized Admin Node Live

Using the `crm_gui` or `crm` command line, it is possible to migrate the virtual machine live from one physical machine to the other. This can be useful, for example, if the system administrator needs to perform maintenance on one physical machine.

Note that this migration method makes use of the `ssh` transport. If your configuration does not allow the root user to `ssh` between physical machines without a password,

you will need to modify the `virt-clone` rules and associated primitives to use a different transport. This will likely involve the set up of certificates for authentication for `libvirt` between the two physical machines. That is not covered in this section.

You can use the command line or GUI to migrate a virtualized admin node. This section describes the GUI method.

Procedure 2-23 Migrating a Virtualized Admin Node Live

To use the `crm_gui` GUI to migrate to a virtualized admin node, perform the following:

1. Start `crm_gui` on one of the two physical machines.
2. Make sure you are in the **Management** section (click **Management** on the left panel).
3. On the right panel, locate **virt-clone**.

It should show that it is running with a green status indicator. The **virt:0** should display with the node on which it is currently running.

4. Right-click on **virt-clone**, then choose **Migrate Resource**.
5. Choose the system to which you want to migrate.
The live migration is initiated.
6. You can confirm it completed successfully by watching `/var/log/messages` on the machine that was previously running the virtual machine.
7. You can use **virt-manager** to connect to the machine and confirm it is still up and running.
8. It is sometimes necessary to use do a resource cleanup to see the resource in `crm` again. If **virt-clone** shows as not running after waiting a couple minutes, then right-click on it and select **cleanup resource**, click on OK. The resource should go back to green.
9. Any time you migrate, a constraint is recorded. If you wish to remove the constraint, right-click on the **virt-clone** entry and select **Clear Migrate Constraints**, then click OK.

Rack Leader Controller High Availability Solution

High Availability Rack Leader Controllers (leader nodes) for SGI ICE systems provide redundant rack leader controllers for SGI ICE racks. A rack leader controller can be taken offline or can fail, and the `tmpfs` booted rack compute nodes will remain fully functional.

This initial implementation of high availability rack leader controllers (HA-RLC) for SGI ICE clusters has two HA-RLCs per rack.

Novell, Inc. high availability software (HAE) running on each rack leader controller (RLC) manages a set of services that are moved from one RLC to the other RLC if the HAE software detects a failure in a monitored service.

Hardware Requirements

This implementation of HA-RLC for SGI ICE requires two lead nodes per rack and an extra network interface card (NIC) per HA-RLC. The extra NIC provides a direct network connection between the HA-RLC for the HAE software.

Software Requirements

HA-RLC is currently implemented only on SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP1) using the SLES 11 SP1 High Availability Extension (HAE). This restriction applies to only the admin and the rack leader nodes.

All current updates to package in the SLES 11 SP1 distribution (distro) and the SLES11 SP1 HAE product must be available to install a HA-RLC SGI ICE cluster.

The major software components that comprise this solution include the following:

- SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP1) base distribution
- SLES 11 SP1 High Availability Extension (HAE)
- SGI Admin Node Autoinstallation DVD for SLES11 SP1
- SLES 11 SP1 distribution updates
- SLES 11 SP1 HAE updates

It is **required** that updates to the SLES11 SP1 distro and to the SLES11 SP1 HAE be used with HA-RLC, in particular, when creating the HA-RLC leader images.

SLES11 SP1 High Availability Extension (HAE) Documentation

For information on the SUSE Linux Enterprise High Availability Extension (HAE) software package, see the *SUSE Linux Enterprise High Availability Extension High Availability Guide*. This is required reading and the information contained in the SLES HAE guide is not replicated in this manual.

The HAE document is available in the `sleha-guide_en-pdf` package from Novell, Inc.

After the repositories are created, updated, and the HA Repository selected, this package can be installed on the admin node, as follows:

```
admin:~ # cinstallman --yum-node --node admin install sleha-guide_en-pdf
```

You can find the HAE guide at the following location:

```
/usr/share/doc/manual/sleha-guide_en-pdf/SLEHA-guide_en.pdf
```

Compute Node `tmpfs` Boot Requirement

It is required that all compute nodes in HA-RLC protected racks use `tmpfs` roots, which implements a memory resident operating system. With the `tmpfs` root, no compute node state information is stored on a RLC, as is the case when the compute nodes use NFS roots.



Warning: It is critical that the HA-RLC compute nodes be booted with a `tmpfs` root. Compute nodes booted with NFS root will crash if a HA-RLC fails.

Since compute nodes using `tmpfs` roots store no state information on a RLC, the HA-RLC can be rebooted, and protected services migrated, without affecting the compute node functionality.

Physical Connections

The connection of the rack leader controller (RLC) to the racks must be via the LOCAL port of the individual rack unit (IRU) chassis management controller (CMC).

In the case of an RLC with redundant management network (RMN) and a separate BMC port, it means that there are three Ethernet connections to the rack, two of which go to LOCAL ports on the CMC and one (the BMC connection) that goes to an LL port. In order to have four LOCAL ports in a rack, there must be four IRUs in that rack.

In addition, there is a direct Ethernet connection between the two HA-RLCs of each rack. This provides a communication path between HA-RLC that does not pass through the stacked CMCs and avoids cluster management traffic (see Figure 2-49 on page 161).

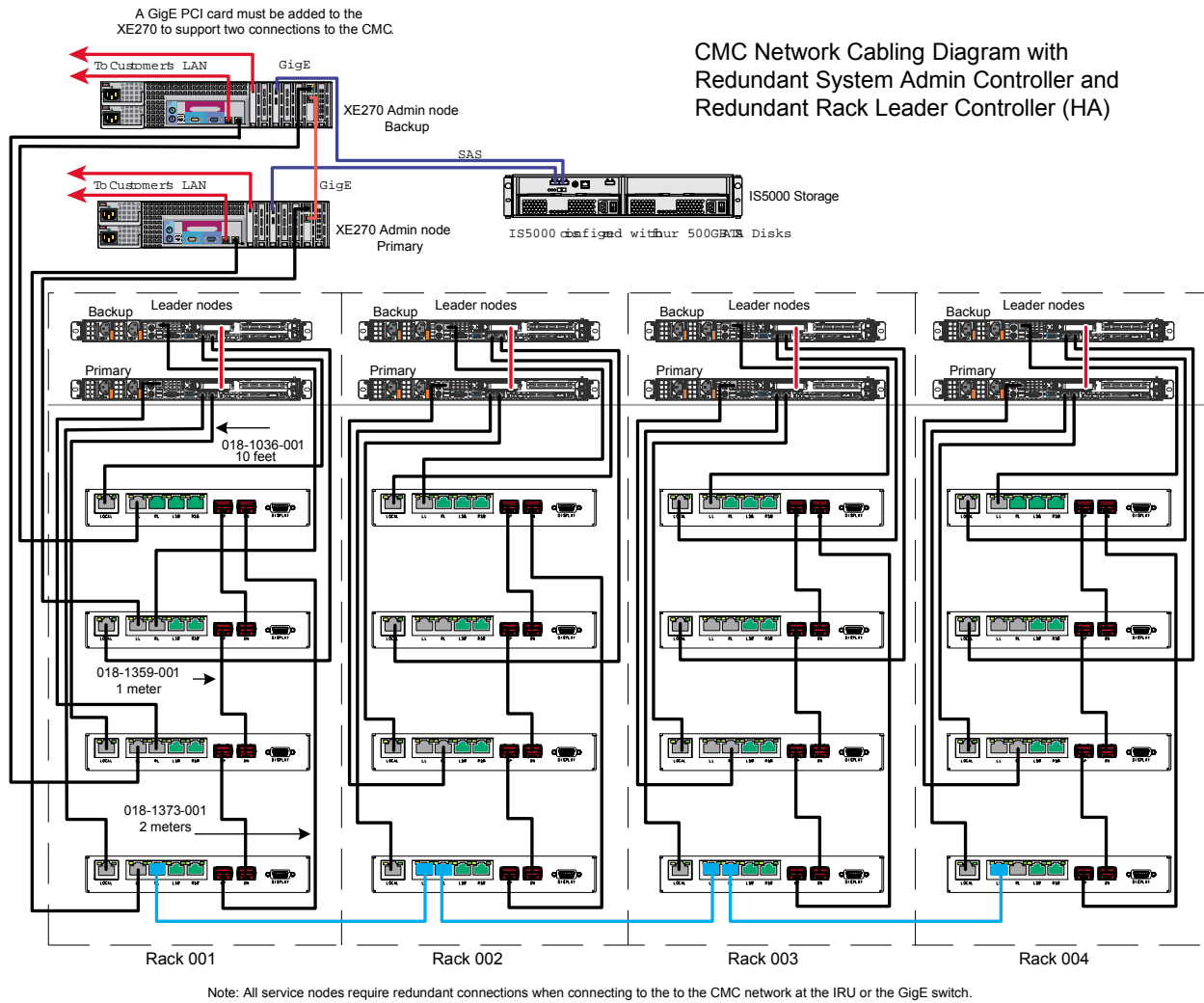


Figure 2-49 CMC Network Cabling Diagram for HA-RLC

High Availability Extension

The SUSE Linux Enterprise High Availability Extension (HAE) software manages *resources*. Examples of managed resources include `blademond`, `dhcpd`, and `conserver`. The HAE software stops, starts and monitors these resources. Only a single instance of these resources runs per rack. These three resources also happen to be Linux Standards Base (LSB) services.

With HA-RLC, all resources are co-located on one or the other HA-RLCs. The HA-RLC on which the resources are running can be said to "host" the resources and can be referred to as the "active" HA-RLC.

The resources can be migrated between the two HA-RLCs for a rack, either in response to administrative commands or in response to a failure detected with one of the managed resources.

For detailed information on HAE software, see the *SUSE Linux Enterprise High Availability Extension High Availability Guide* (see "SLES11 SP1 High Availability Extension (HAE) Documentation" on page 159).

HA-RLC Naming

The HA-RLC hostnames and interfaces for a rack are distinguished from each other with the addition of a "1" or a "2" to the standard RLC interface names.

For example, here are the names of the HA-RLC interfaces for racks 1 and 2. The names of the interfaces for racks 3 to n follow this same naming scheme.

The Rack 1 HA-RLC hostnames and interfaces are named, as follows:

- `r1lead`
Logical interface to the rack 1 RLC. Migrates.
- `r1lead1`
Host interface to the first of the two RLCs for rack1. Associated with the physical device. Does not migrate.
- `r1lead1-bmc`
BMC interface to the first of the two RLCs for rack1. Associated with the physical device. Does not migrate.
- `r1lead2`

Host interface to the second of the two RLCs for rack1. Associated with the physical device. Does not migrate.

- `r1lead2-bmc`

BMC interface to the second of the two RLCs for rack1. Associated with the physical device. Does not migrate.

The Rack 2 HA-RLC interfaces are named, as follows:

- `r2lead`

Logical interface to the rack 2 RLC. Migrates.

- `r2lead1`

Host interface to the first of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- `r2lead1-bmc`

BMC interface to the first of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- `r2lead2`

Host interface to the second of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- `r2lead2-bmc`

BMC interface to the second of the two RLCs for rack2. Associated with the physical device. Does not migrate.

HA-RLC Networking Changes

Changes to default SGI ICE networking were needed to implement HA-RLC. In particular, HA-RLC addresses on the `VLAN_HEAD`, `VLAN_BMC`, and `VLAN_GBE` networks have been implemented as IP aliases that are managed as HAE resources.

Each HA-RLC also has a physical address via which the node is accessible from the admin node, regardless of where the HAE resources are hosted.

The IP alias address associated with `r1lead` is a logical rack alias and requests to `r1lead` will be directed to whichever HA-RLC is currently hosting the resources.

In contrast, the IP physical addresses associated with `r1lead1` and `r1lead2` always connect to that node. The BMC `console` connect is via the physical address.

```
admin:~ # console r1lead1
```

The console connections to the physical address are not disrupted by migration of the resources. The `ssh` connections to the physical addresses are also not disrupted by migration of resources. The `ssh` connections to the logical addresses are disrupted by migration.

Figure 2-49 on page 161 shows the CMC network cabling diagram with a redundant system admin controller and redundant rack leader controller.

HAE with HA-RLC

Each pair of HA-RLC comprise a separate instance of a HAE cluster. There is no communication between the separate HAE instances, that is, HAE instances running on different racks are independent.

The HAE software controls resources. LSB `init` scripts such as `blademon`, `dhcpd`, and `conserver` map directly to HAE resources.

HAE supplied scripts can be used for multiple HAE resources. For example, the script `ocf::heartbeat:IPaddr2` is used to implement the three IP aliases used by HA-RLC.

All HA-RLC HAE resources are *co-located*, which means that all resources run simultaneously on one or the other of each HA-RLC pair.

A `stonith-ipmi` resource runs on each of the HA-RLCs. These `stonith` resources do not migrate with failure. The `stonith-ipmi` resources monitor peer BMC, but `stonith` action is not enabled.

If there is a failure of one of the HAE managed resources, the resources are all migrated to the other HA-RLC. Resources can also be administratively migrated between the HA-RLC of a rack.

As the HAE instance running on a rack is independent of HAE instances running on other racks, resources cannot be migrated to a HA-RLC of a different rack.

The HAE configuration is loaded when the HA-RLC boot for the first time. It is not intended that configuration be changed except via SGI update.



Warning: Making changes to the SGI supplied HAE resources is not supported. Changes to HAE resources are expected to be done only with SGI release upgrades.

Installing the ICE Admin Node for a HA-RLC Cluster

Install the ICE Admin Autoinstall SLES 11 SP1 DVD as described in "Installing SLES11 on the Admin Node" on page 42.

Do **NOT** proceed to the steps described in the "`configure-cluster` Command Cluster Configuration Tool" on page 56 section until you have finished creating repositories, as described in "Create SLES11 SP1 and HAE Repositories" on page 165.

Create SLES11 SP1 and HAE Repositories

Before running the `configure-cluster` command for a cluster that will include HA-RLC, you **must** create updated repositories for the SLES 11 SP1 distro and for SLES 11 SP1 HAE.



Warning: Updated repositories are required to build working HA-RLC images

Since the default leader image is created by `configure-cluster`, it is necessary that the updated repositories exist before running `configure-cluster`.

Create updated repositories for the distro and HAE using the method of your choice. The method of obtaining updates with which to create update repositories will vary from site to site. An example for creating these repositories from local update sources follows.

Use the `crepo --select` command to select the repositories so that they are used when building the default images.

Update Repositories Example

In the example in this section four separate repositories are created, as follows:

- SLES 11 SP1 distro
- SLES 11 SP1 HAE
- SLES 11 SP1 distro updates
- SLES 11 SP1 HAE updates

After these repositories are created, the four repositories must be marked as selected using the `crepo` command.

The SLES 11 SP1 distro and SLES 11 SP1 HAE GA release packages are obtained from the respective ISOs in a local or NFS mounted directory.

The SLES 11 SP1 distro update packages and the SLES 11 SP1 HAE update packages are obtained (from <http://www.novell.com/home/>, a support contract is needed, see "SLES Admin Nodes: Update the SLES Package Repository" on page 128) and are expected to be in separate directories, either local or NFS mounted.

The bash shell is used in the following examples.

Procedure 2-24 Create Software Repositories for HA-RLC

To create software repositories for installing the HA-RLC solution, perform the following steps:

1. Create SLES 11 SP1 distro and SLES 11 SP1 HAE repositories from ISOs in the current local directory (for reference, see `crepo --help`), as follows:

```
admin:~ # crepo --add SLES-11-SP1-DVD-x86_64-GM-DVD1.iso
admin:~ # crepo --add SLE-11-SP1-HA-x86_64-GM-Media1.iso
```

2. Create a SLES11 SP1 distro updates repository from the updates packages located in the local SLES11-SP1-Updates/sle-11-x86_64/rpm, as follows:

```
admin# cd SLES11-SP1-Updates/sle-11-x86_64/rpm
admin# find -name "*.rpm" | egrep -v '\.delta\.rpm$|\.patch\.rpm$|\.src\.rpm$|\.nosrc\.rpm$' | \
    sort > /tmp/distro-updates-list-$$

admin# DISTRO_UPDATES_DIR=/tftpboot/updates/sles11sp1-updates
admin# mkdir -p $DISTRO_UPDATES_DIR

admin# for f in `cat /tmp/distro-updates-list-$$`; do cp -a $f $DISTRO_UPDATES_DIR; done
admin# crepo --add $DISTRO_UPDATES_DIR --custom "sles11sp1-updates"
```

3. Create a SLES11 SP1 HAE updates repository from the updates packages located in the local SLES11-HAE-SP1-Updates/sle-11-x86_64/rpm, as follows:

```
admin# cd SLES11-HAE-SP1-Updates/sle-11-x86_64/rpm

admin# find -name "*.rpm" | egrep -v '\.delta\.rpm$|\.patch\.rpm$|\.src\.rpm$|\.nosrc\.rpm$' | \
    sort > /tmp/hae-updates-list-$$

admin# HAE_UPDATES_DIR=/tftpboot/updates/SUSE-Linux-Enterprise-High-Availability-Extension-11-SP1-updates
admin# mkdir -p $HAE_UPDATES_DIR

admin# for f in `cat /tmp/hae-updates-list-$$`; do cp -a $f $HAE_UPDATES_DIR; done
admin# crepo --add $HAE_UPDATES_DIR --custom "SLES-HAE-updates"
```

4. The newly created repositories must be selected so that they are used when building the default images (for reference, see `crepo --help`), as follows:

```
admin:~ # crepo --select SUSE-Linux-Enterprise-Server-11-SP1
admin:~ # crepo --select sles11sp1-updates
admin:~ # crepo --select SUSE-Linux-Enterprise-High-Availability-Extension-11-SP1
admin:~ # crepo --select SLES-HAE-updates
```

5. To display the currently selected repositories, perform the following:

```
admin:~ # crepo --show
* SLES-HAE-updates : /tftpboot/updates/SUSE-Linux-Enterprise-High-Availability-Extension-11-SP1-updates
* SUSE-Linux-Enterprise-High-Availability-Extension-11-SP1 : /tftpboot/other/SUSE-Linux-Enterprise-
  High-Availability-Extension-11-SP1
* SUSE-Linux-Enterprise-Server-11-SP1 : /tftpboot/distro/sles11sp1
* sles11sp1-updates : /tftpboot/updates/sles11sp1-updates
```

Note: The leading "*" indicates that the repository is currently "selected".

Initial Cluster Configuration with HA-RLC

The Cluster Configuration Tool (`configure-cluster`) is used to perform initial cluster configuration (see "`configure-cluster` Command Cluster Configuration Tool" on page 56).

It is required when installing a cluster that will include HA-RLC racks that the SLES11 SP1 distro and SLES11 SP1 HAE repositories, and update repositories, be created before running the `configure-cluster` command.

Although it is possible to recover from the failure to create the updates repositories before running `configure-cluster`, this is not the supported method of installing HA-RLC clusters.

The `configure-cluster` instructions in this section are a supplement to the installation instructions provided in "`configure-cluster` Command Cluster Configuration Tool" on page 56.

Procedure 2-25 Initial Cluster Configuration with HA-RLC

To configure a HA-RLC cluster, perform the following steps:

1. Use the `configure-cluster` command to install the following embedded ISOs:
 - SGI-Foundation-Software
 - SGI-Management-Center
 - SGI-Tempo

SLES 11 SP1 distro and SLES11 HAE repositories, plus updates repositories for these products, should have already been created (see "Create SLES11 SP1 and HAE Repositories" on page 165). You do not need to recreate them.

2. Install the Admin cluster software (see "Installing SLES11 on the Admin Node" on page 42).
3. Using `configure-cluster`, confirm default network settings or make local modifications. These settings cannot be changed after the cluster is initialized.
4. Configure the locally accessible NTP server.
5. Do not proceed unless the update repositories were created, as described "Create SLES11 SP1 and HAE Repositories" on page 165. Using `configure-cluster`, perform the infrastructure setup.

The admin node log file of particular interest during the image creation process is `/var/log/cinstallman`.

Note: This step can take 30 minutes or more to complete

6. Using `configure-cluster`, configure the local DNS resolvers

Discover HA—RLC Racks

After completing the **Initial Cluster Setup Tasks** with `configure-cluster` described in "Initial Cluster Configuration with HA-RLC" on page 168, the HA-RLC racks can be installed.

Discover HA-RLC Rack Overview

During the `discover rack` operations, the following events occur:

- HA-RLCs are installed.

- HAE software is initialized on the HA-RLCs.
- Default compute node images are downloaded to the HA-RLCs and are installed.
- Compute nodes are booted.

The `discover` command discovers the MAC addresses of the HA-RLC and configures the admin node for the installation of the HA-RLC. The `discover` process exits, leaving the installation of the HA-RLC to continue. The remainder of the `discover` rack operations are started when the HA-RLC boot.

The admin log file of particular interest for HA-RLC discovery and installation operations is, as follow:

```
admin:~ # tail -f /var/log/messages
```

In particular, monitor the messages related to DHCP and `tftp`.

Each pair of HA-RLC will boot into HA mode, and one (and only one) of the HA-RLC will start hosting the resources for each rack.

When the HAE software comes on-line, `blademon` (one of the HAE managed resources) discovers the rack hardware (CMCs, compute nodes) and pushes the rack information to the admin node. The admin node then configures the HA-RLC and pushes compute node images to the HA-RLC.

After the HA-RLCs are fully configured, the compute nodes boot from the HA-RLC.

Discover Both HA-RLCs in a Rack

SGI recommends that whenever both HA-RLCs of a rack are to be discovered, the HA-RLC be discovered together.

Either of the following commands will install both rack HA-RLCs.

Discover two HA-RLCs for rack 1, as follows:

```
admin:~ # discover --rack 1,ha=all
```

Discover two HA-RLCs per rack for racks 1, 2, and 3, as follows:

```
admin:~ # discover --rack 1,3,ha=all
```

The HA-RLCs are discovered sequentially by the `discover` command. For a single rack, the `discover --rack 1,ha=all` command requests that power be applied

first to `r1lead1` and then will request that power be applied to `r21lead2`. "Apply power" means to plug-in the HA-RLC, but **not** to push the power button.

For more information on the `discover` command, see "discover Command" on page 80.

Monitor HA-RLC Installation

You can follow the installation of the HA-RLCs by opening BMC consoles for the two HA-RLC from different admin node windows or from different window tabs. For example, to follow the installation of the rack 1 HA-RLC, run these commands in different windows:

```
admin:~ # console r1lead1
admin:~ # console r1lead2
```

Monitor the admin node messages file for HA-RLC install messages, in particular look for the initial DHCP and tftp messages, as follows:

```
admin:~ # tail -f /var/log/messages
```

Confirm HAE After HA-RLC Boot

When the rack leader nodes boot for the first time, the HAE servers will initialize and one of the HA-RLC pair will start the HAE resources. Generally, this will be the first leader node discovered for the rack.

Use the `crm_mon` command to determine which HA-RLC is hosting the resources. This command can be run via `ssh` from the admin node or from a login to either of the HA-RLCs, as follows:

```
admin:~ # ssh r1lead crm_mon -l
=====
Last updated: Mon Apr 18 14:14:22 2011
Stack: openais
Current DC: r1lead1 - partition with quorum
Version: 1.1.5-5ce2879aa0d5f43d01629bc20edc6868a9352002
2 Nodes configured, 2 expected votes
7 Resources configured.
```

```
=====
```

```
Online: [ r1lead1 r1lead2 ]
```

```
blademond      (lsb:blademond):      Started r1lead1
conserver      (lsb:conserver):      Started r1lead1
dhcpd (lsb:dhcpd):      Started r1lead1
Resource Group: alias-group
  head-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
  vlan1-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
  vlan2-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
stonith-ipmi-1 (stonith:external/ipmi):      Started r1lead1
stonith-ipmi-2 (stonith:external/ipmi):      Started r1lead2
```

It takes several minutes for HAE to initialize and for the resources to be started.

Until `blademond` initializes the `dhcpd.conf` file (actually, `ice.conf`), `crm_mon` reports DHCP resource errors.

Until `blademond` probes the rack CMC the slot maps and the compute nodes and the `conserver` configuration file has been initialized, `crm_mon` reports errors for the `conserver` resource.

There should be no `stonith-ipmi` errors. There should be no `stonith-ipmi` resources if only one HA-RLC is installed. Errors are reported for the `stonith-ipmi` resource if an HA-RLC has been discovered and is installed but has not booted.

Monitor Compute Node Set up

On the active HA-RLC (indicated by the resource being `Started r1lead1` in the output of `crm_mon -l`), the blade monitoring daemon `blademond` starts, interrogates the rack CMC for blade (slot) information, configures `dhcpd` and starts the `discover-rack` process on the admin node.

The `discover-rack` process running on the admin node controls the configuration of the HA-RLC for booting the compute nodes.

Progress can be monitored from the following log files on the admin node and on the active HA-RLC (in this example, assumed to be `r1lead1`):

```
r1lead1:~ # tail -f /var/log/blademond
admin:~ # tail -f /var/log/discover-rack
```

Completion of HA-RLC configuration for booting the compute nodes occurs when the following message is logged in the `/var/log/discover-rack` file:

```
setup-new-blades: booting blades finished
```

Completion messages will also be logged on the "active" HA-RLC in `/var/log/blademon`d file and the log will indicate that the blademon is, as follows:

```
Jun 17 21:39:25 r1lead1 blademon: calling discover-rack on admin node
Jun 17 21:45:41 r1lead1 blademon: moving /var/opt/sgi/lib/blademon/slot_map.new to
/var/opt/sgi/lib/blademon/slot_map
Jun 17 21:45:42 r1lead1 blademon: INFO: pushing /var/opt/sgi/lib/blademon/slot_map to
r1lead2:/var/opt/sgi/lib/blademon
Jun 17 21:45:43 r1lead1 blademon: INFO: pushing /etc/dhcpd.conf.d/ice.conf to
r1lead2:/etc/dhcpd.conf.d/ice.conf
Jun 17 21:45:43 r1lead1 blademon: sleeping ...
```

Confirm Boot of Compute Nodes

To confirm the boot of the compute nodes in rack 1, use a command similar to the following:

```
admin:~ # cexec rack_1: uptime
```

You can also use the `cpower` command to confirm that the compute nodes have booted. For example, for rack 1, perform the following:

```
admin:~ # cpower --status r1i*n*
```

Confirm Initial HA-RLC Setup

The two HA-RLCs for a rack must always be configured to boot the same images. This can be confirmed by comparing the compute node boot links in `/tftpboot/pxelinux.cfg` of the two HA-RLCs.

With the initial install, all links will point to the `tmpfs` version of the default compute image (`compute-sles11sp1`) boot file.

For example:

```
admin:~ # pdsh -w r1lead1,r1lead2 ls -l /tftpboot/pxelinux.cfg/ | \
awk '{print $1,$9,$10,$11}' | dshbak -c
-----
r1lead[1-2]
-----

C0A89F0A -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F19 -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1A -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1B -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1C -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1D -> /tftpboot/compute-sles11sp1/pxelinux.config-2.6.32.29-0.3-default-tmpfs
```

Note that the hex number is the compute node IP address.

The links will only be present when a HA-RLC has been fully configured to boot the compute nodes.

If compute nodes are later set to boot different images, the links for those compute nodes will point to the different boot files. The links are the same on both HA-RLCs, the boot files will be `tmpfs` and not `nfs`.

Sequential Discovery of HA-RLC

The discovery of a single HA-RLC makes it possible to add a replacement HA-RLC to the cluster. The feature also makes it possible to first discover one in a pair of HA_RLCs and later discover the other HA-RLC pair.

SGI recommends that the initial HA-RLC discovery be of both of the rack HA-RLCs, that is, using `ha=all`.

For rack 1, discover first one of the HA-RLC and then discover the second HA-RLC. For example, first discover `r1lead1` and after `r1lead1` is fully configured, HAE is running, and the compute nodes have booted, then discover the second HA-RLC, `r1lead2`.

To discover HA-RLC r1lead1, perform the following:

```
admin:~ # discover --rack 1,ha=1
```

After the configuration of r1lead1 completes, discover HA-RLC r1lead2, as follows:

```
admin:~ # discover --rack 1,ha=2
```



Warning: Before starting the discovery of the second HA-RLC, all configuration steps **must** have completed for the first HA-RLC. This means that the rack compute nodes will have booted. Overlapping individual discovery of HA-RLCs is **NOT** supported.

For additional information about installing a single HA-RLC, see the section below, "Discover replacement HA-RLC".

Monitor Rack HAE Status

After the HA-RLC installation completes, there are multiple methods of monitoring the HAE status of a rack as described in this section.

crm_mon Command Line Utility

The command line utility `crm_mon` can be used to check the HAE status of a rack, as follows:

```
admin:~ # ssh r1lead crm_mon -1
=====
Last updated: Tue May 17 15:27:23 2011
Stack: openais
Current DC: r1lead1 - partition with quorum
Version: 1.1.5-5ce2879aa0d5f43d01629bc20edc6868a9352002
2 Nodes configured, 2 expected votes
7 Resources configured.
=====

Online: [ r1lead1 r1lead2 ]

blademond      (lsb:blademond):      Started r1lead1
conserver      (lsb:conserver):      Started r1lead1
dhcpd          (lsb:dhcpd):          Started r1lead1
```

```
Resource Group: alias-group
  head-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
  vlan1-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
  vlan2-alias (ocf::heartbeat:IPaddr2):      Started r1lead1
  stonith-ipmi-1 (stonith:external/ipmi):    Started r1lead1
  stonith-ipmi-2 (stonith:external/ipmi):    Started r1lead2
```

The list and order of resources may differ with release version.

One stonith resource runs on each HA-RLC and only functions to monitor the BMC of the peer HA-RLC. If only a single HA-RLC is installed, there will be no stonith-impd resources configured.

You can also use an abbreviated version of `crm_mon` to check HA-RLC HAE status, as follows:

```
sys-name:~ # pdsh -g leader crm_mon -s
r2lead2: Ok: 2 nodes online, 6 resources configured
r1lead2: Ok: 2 nodes online, 6 resources configured
r1lead1: Ok: 2 nodes online, 6 resources configured
r2lead1: Ok: 2 nodes online, 6 resources configured
```

The number of resources configured for HA-RLC may differ with release version.

`crm_gui` Utility

The `crm_gui` runs from a login to a HA-RLC. This tool provides significant functionality for working with HAE.



Warning: Making changes to the HAE configuration is not supported.

From the `r1lead1` leader node, launch `crm_gui`, as follows:

```
r1lead1# crm_gui &
```

The CRM GUI appears, as shown in Figure 2-50 on page 177.

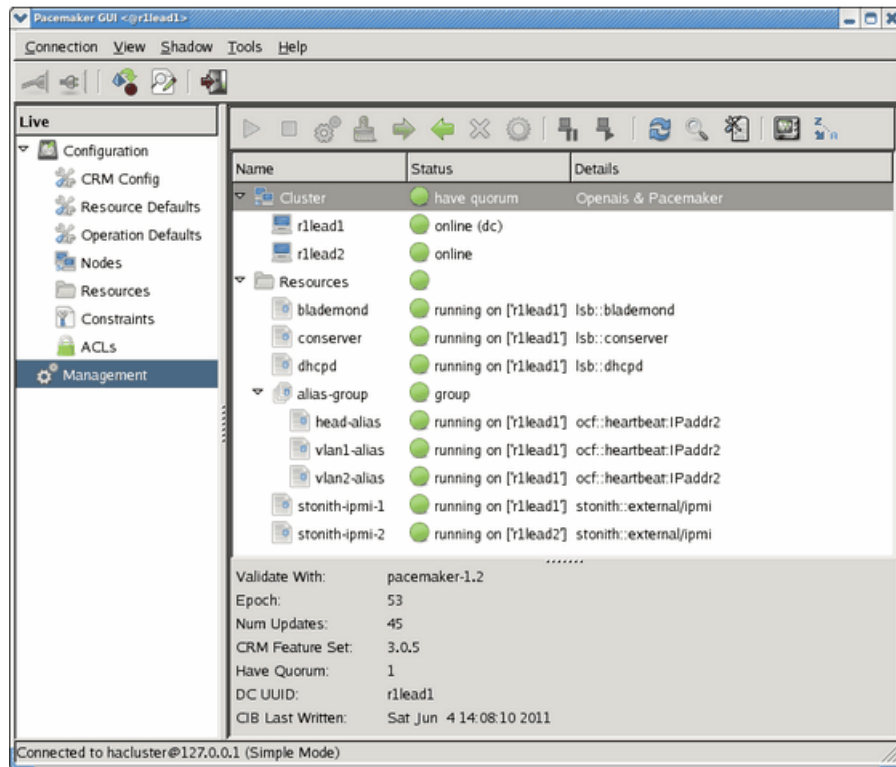


Figure 2-50 CRM Cluster Status GUI

Hawk Browser Interface

The Hawk browser interface provides status and some basic commands for an individual HAE instance.

Start the Firefox browser on the admin node and point to the Hawk server (port 7630) at a HA-RLC physical interface. If pointed at a HA-RLC alias, the connection is dropped when the resources are migrated.

For example, to connect to the rack 1 HAE instance via the `r1lead1` leader node, enter the following in your Firefox browser: `https://r1lead1:7630/`.

Login as **hacluster**, using the default password **sgisgi**. Confirm the security exception. The default user **hacluster** is configured for HA-RLC in `/etc/opt/sgi/conf.d/80-corosync`.

By left-clicking on the wrench icon of a resource, the commands: **Start**, **Stop**, **Move**, **Drop Relocation Rule** and **Clean Up** run on a resource, as shown in Figure 2-51 on page 178.

By left-clicking on the wrench icon of a HA-RLC, **Online**, **Offline**, and **Fence Node** functions are available, as shown in Figure 2-56 on page 184. Fencing is not implemented.

For more information, see "Starting the HA Web Console and Logging In" in the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

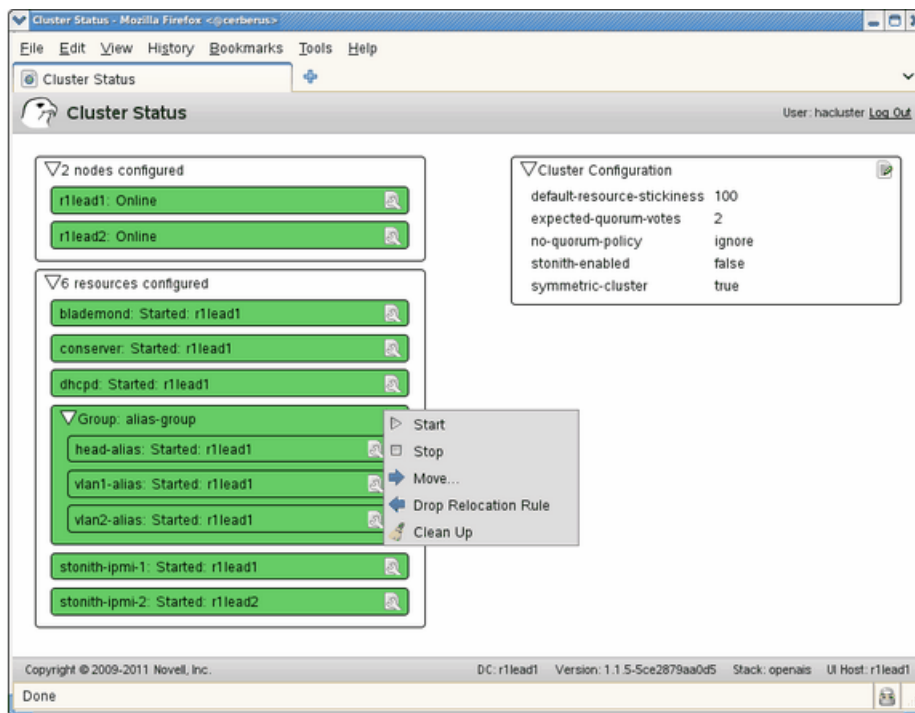


Figure 2-51 Hawk Interface with Resource Control Pop-up Menu

Administrative Resource Migration

Migrating resources is very useful for testing and is also useful for moving resources from a HA-RLC if maintenance is needed on that HA-RLC.

Three tools that can be used to migrate resources are , as follows:

- `crm` command line interface
- `crm_gui` run on a HA-RLC
- The Hawk browser interface to the HAE running on the HA-RLC

Note: In response to an administrative migration using `crm` or using `crm_gui`, HAE creates a location constraint that forces the resources to migrate. This location constraint **must** be deleted for it to be possible for the resources to migrate back.

`crm` - Resource Migration

Login to a HA-RLC via a physical interface, or from the admin node send commands to the rack HA alias.

To migrate all resources from `r1lead1` to `r2lead2`, perform the following:

```
admin:~ # ssh r1lead crm resource migrate alias-group r1lead2
admin:~ # ssh r1lead crm_mon -1
admin:~ # ssh r1lead crm configure show | grep prefer
admin:~ # ssh r1lead crm configure delete cli-prefer-alias-group
admin:~ # ssh r1lead crm configure show | grep prefer
admin:~ # ssh r1lead crm_mon -1
```

The `cli-prefer-alias-group` must be deleted to migrate the resources back.

It is possible that this constraint will also prevent failure driven resource migration.

`crm_gui` - Resource Migrate

For running the `crm_gui` from a HA-RLC, it is best to login via a physical interface. With the migration, a login via the rack HA alias is dropped.

For example, to login to `r1lead1` and start `crm_gui`, perform the following:

```
admin:~ # ssh r1lead1
r1lead1# crm_gui &
```

For migrating all of the resources (all that migrate), migrate the **alias-group**, as shown in Figure 2-52 on page 180 and Figure 2-53 on page 181.

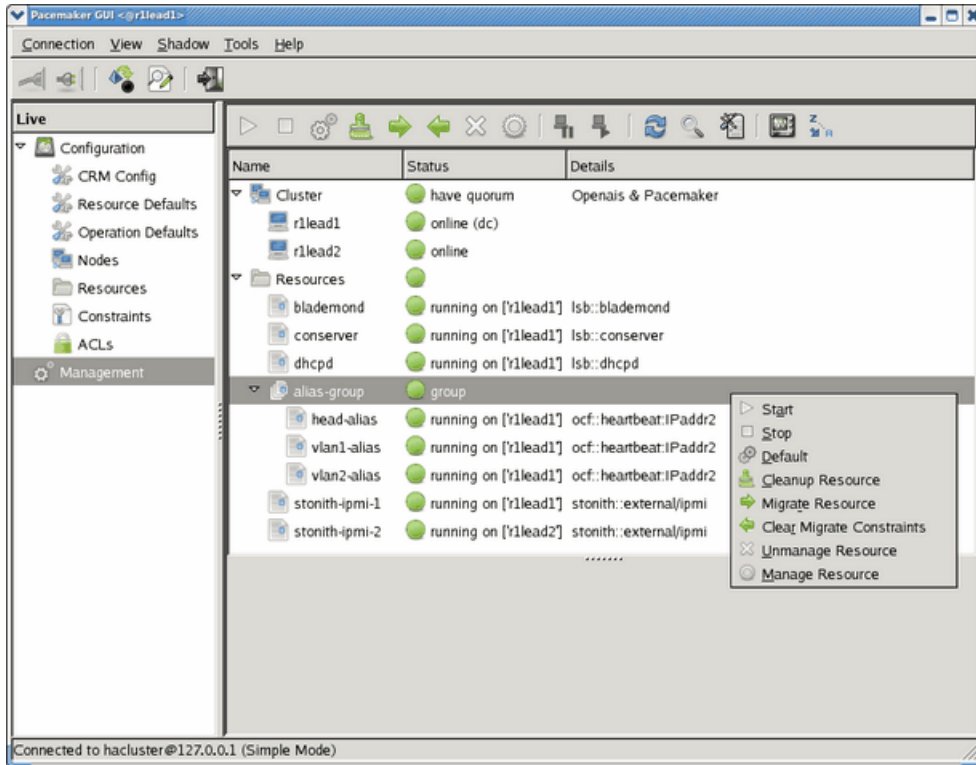


Figure 2-52 crm_gui Migrate alias-group Resource Example

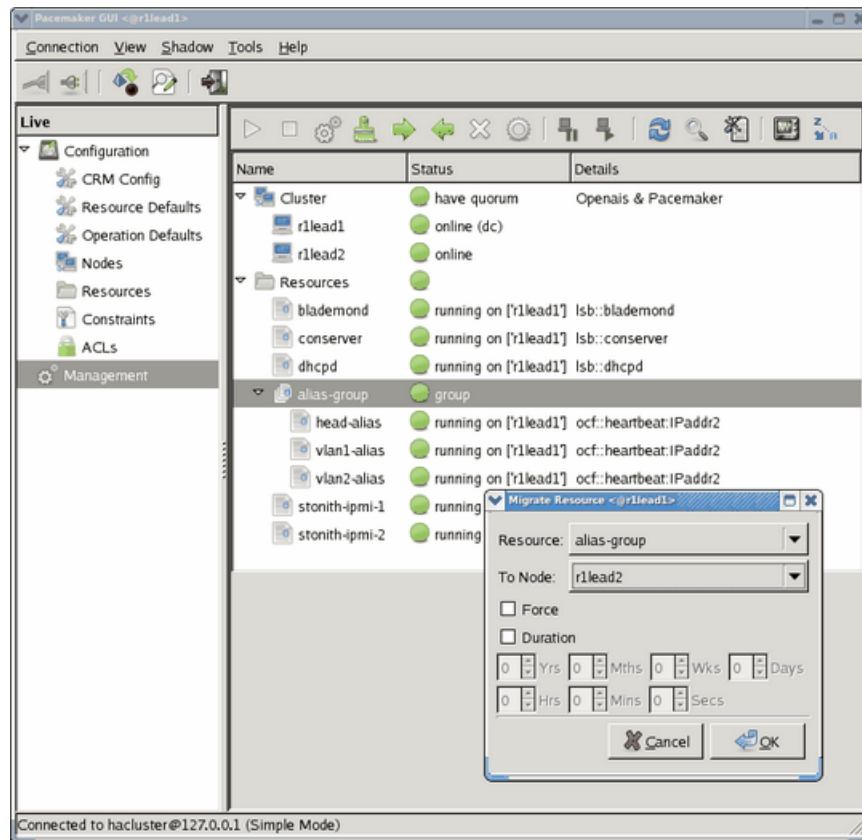


Figure 2-53 crm_gui Migrate alias-group Resource Example

As with `crm resource migrate`, manual migration via the `crm_gui` creates a location constraint. This preferred location constraint **must** be deleted after the migration, either via the **Constraints** screen or via the button at the top of the **Management** screen, as shown in Figure 2-54 on page 182.

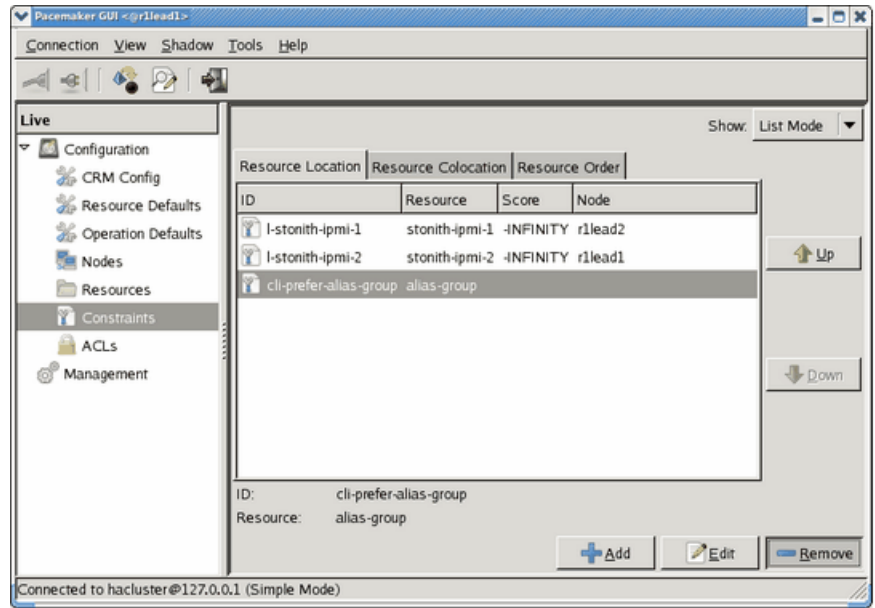


Figure 2-54 CRM GUI Configuration Constraints Selection Example

Hawk Browser Interface — Resource Migration

The Hawk browser interface provides status and some basic commands for an individual HAE instance.

Start Firefox on the admin node and point to the Hawk server (port 7630) at a HA-RLC physical interface. If pointed at a HA-RLC alias, the connection is dropped when the resources are migrated.

For example, to connect to the rack 1 HAE instance via r1lead1, enter the following:
`https://r1lead1:7630/.`

Login as **hacluster** and with the default password **sgisgi**. Confirm the security exception. The default **hacluster** user is configured for HA-RLC in `/etc/opt/sgi/conf.d/80-corosync`.

By left-clicking on the wrench icon of a resource, **Start**, **Stop**, **Move**, **Drop Relocation Rule**, and **Clean Up** run on a resource, as shown in Figure 2-55 on page 183.

By clicking on the wrench icon of a HA-RLC, **Online**, **Offline**, and **Fence Node** functions are available, as shown in Figure 2-56 on page 184. Fencing is not implemented.

For more information, see "Starting the HA Web Console and Logging In" in the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

This section describes two methods of resource migration using the Hawk browser interface.

Migrate Using Hawk - Method 1

On the Hawk interface, left-click on the **alias-group** wrench icon and select **move**, and then select the leader node to which to migrate the resources, as shown in Figure 2-55 on page 183.

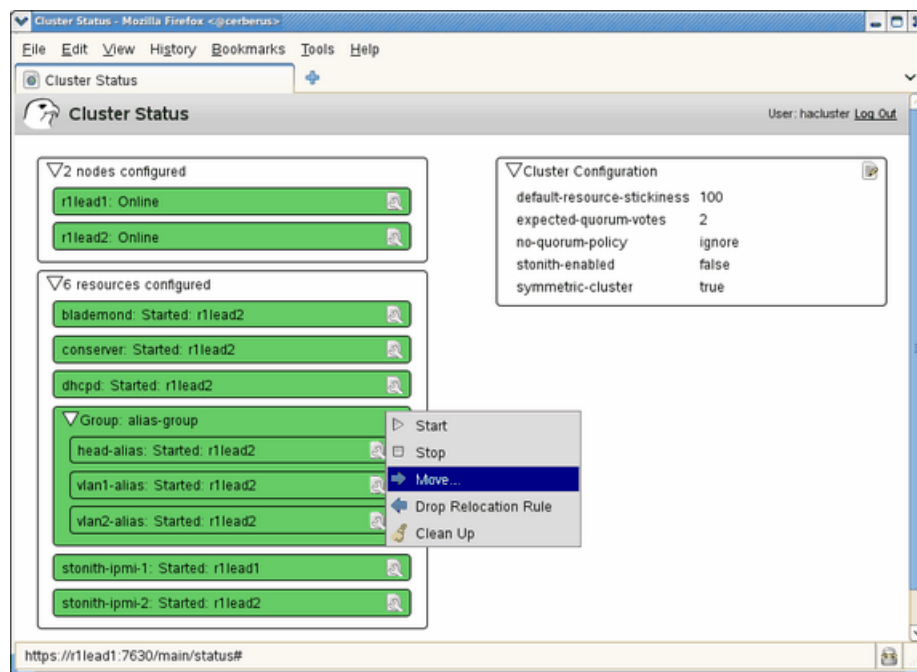


Figure 2-55 Using Hawk Browser Interface to Migrate Resources (Method 1 of 2)

Monitor that the resources do migrate. If they do not migrate, it is probably necessary to clear the migration constraint.

Clear the migration constraint with **Drop Relocation Rule**, which is accessible with a left-click on the **alias-group** wrench.

Migrate Using Hawk - Method 2

The Hawk interface commands **Standby** and **Online** can be used to migrate resources between HA-RLC, as shown in Figure 2-56 on page 184.

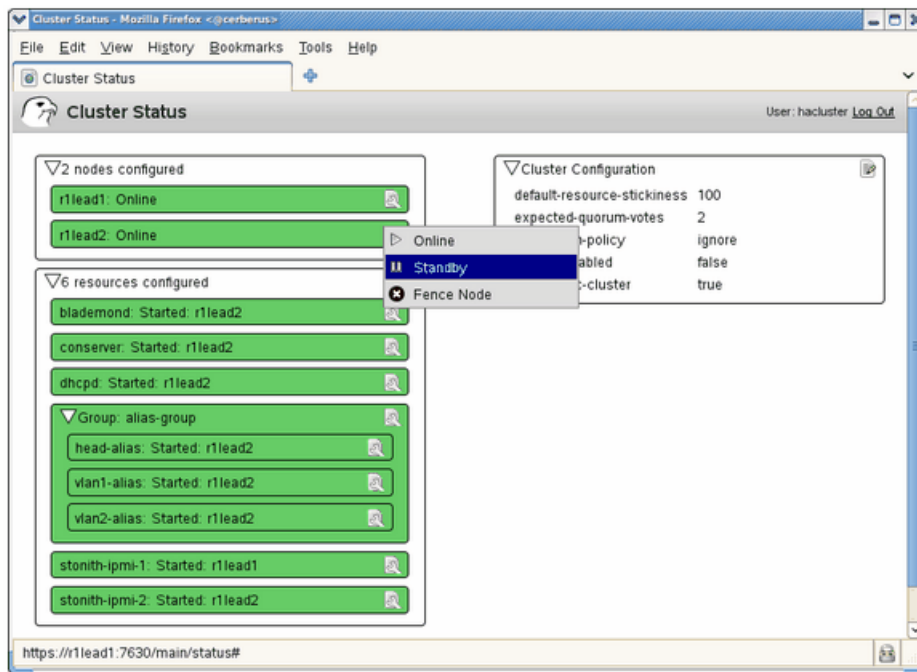


Figure 2-56 Using Hawk Browser Interface to Migrate Resources (Method 2 of 2)

With both HA-RLCs online, and all of the resources started, select the wrench icon associated with the HA-RLC which is currently hosting the resources. Select **Standby** from the pop-up menu. Placing this HA-RLC in **standby** mode causes the resources to be migrated to the **online** HA-RLC.

This action does not create a migration constraint.

To migrate the resources back to the first HA-RLC, select **online** from the pop-up menu associated with the first HA-RLC (which is in standby mode) to bring it back **online**.

Then force the HA-RLC hosting the resources into **standby** mode.

After the resources have migrated, bring the HA-RLC in **standby** mode back **online**.

Replace a HA-RLC

It is possible to take a HA-RLC offline, remove it from the cluster, and to then discover a replacement HA-RLC.

In this context "take offline" can include failure of the HA-RLC.

Remove an HA-RLC from a Cluster

There are two commands associated with removing a HA-RLC from a cluster:

- `discover --delrack`

The purpose of the `discover --delrack` command is to keep place holders in the database for the replacement HA-RLC, so that the original IP addresses are reused when the replacement HA-RLC is discovered.

In this example, the command sets the database state of HA-RLC `r1lead1` to `NOT_EXIST` and does some cleanup.

To delete the first HA-RLC of rack 1, perform the following:

```
admin:~ # discover --delrack 1,ha=1
```

Note: If the remaining HA-RLC (in this example, that would be `r1lead2`) is also deleted, that is, with `ha=2`, then the state of `r1lead2` and rack compute nodes are also set to `NOT_EXIST` and discovery of the whole rack is necessary.

- To purge the node from the database, perform the following:

```
admin:~ # cadmin --db_purge --node r1lead1
```

In this example, the command purges all information referring to `r1lead1` from the cluster and the cluster database. A replacement HA-RLC will be assigned the next (higher) addresses in the database tables.

Note: If the remaining HA-RLC (in this example, that would be `r1lead2`) is also purged, then the rack and compute nodes are purged from the database, and discover of the whole rack will be necessary.

Discover Replacement HA-RLC

After physically replacing the HA-RLC, the `discover` command is used to discover the single replacement HA-RLC. It is assumed that the peer HA-RLC is currently hosting the HAE resources and is fully configured.

For example, to discover the HA-RLC `r1lead1`, perform the following:

```
admin:~ # discover --rack 1,ha=1
```

The `discover` command prompts for power to be applied to the replacement HA-RLC, that is, for the HA-RLC to be plugged in.

If the BMC and IP MAC addresses for the replacement node are known, then that information could be entered into a `macfile` and the "apply power to discover" prompt can be avoided.

The discovery of the replacement HA-RLC initiates the installation of the default lead node image, the HA-RLC boots and joins the rack HAE cluster. The HAE software then synchronizes the replacement node with the existing HAE information.

The compute node images that are **in-use** on the running HA-RLC will be loaded on the replacement HA-RLC and that HA-RLC will be configured to boot compute nodes with the appropriate image.

Compute node images that are installed but not "in-use" are not installed on the replacement HA-RLC. However, it is not possible to set a compute node to use one of those images without again pushing the image to the rack.

System Operation

This chapter describes how to use the SMC for SGI ICE systems management software to operate your SGI ICE system and covers the following topics:

- "Software Image Management" on page 187
- "Power Management Commands" on page 220
- "C3 Commands" on page 227
- "cadmin: SMC for SGI ICE Administrative Interface" on page 232
- "Console Management" on page 238
- "Keeping System Time Synchronized" on page 240
- "Changing the Size of Per-node Swap Space" on page 245
- "Switching Compute Nodes to a `tmpfs` Root" on page 247
- "Setting up Local Storage Space for Swap and Scratch Disk Space" on page 248
- "Changing the Size of `/tmp` on Compute Nodes" on page 242
- "RAID Utility" on page 254
- "Restoring the `grub` Boot-loader on a Node" on page 257
- "Backing up and Restoring the System Database" on page 257
- "Enabling EDNS" on page 259
- "Firmware Management" on page 259

Software Image Management

This section describes image management operations.

This section describes Linux services turned off on compute nodes by default, how you can customize the software running on compute nodes or service nodes, create a simple clone image of compute node or service node software, how to use the `cimage` command to push images to compute nodes, how to use `crepo` command to

manage software image repositories, and how to use the `cinstallman` command to create compute and service node images. It covers these topics:

- "Finding Which Distributions (distros) Are Supported" on page 188
- "Operating Systems Supported per Node Type" on page 189
- "Compute Node Services Turned Off by Default" on page 190
- "`crepo` Command" on page 191
- "`cinstallman` Command" on page 195
- "Customizing Software On Your SGI ICE System" on page 198
- "`cimage` Command" on page 206
- "Using `cinstallman` to Install Packages into Software Images" on page 210
- "Using `yum` to Install Packages on Running Service or Leader Nodes" on page 211
- "Creating Compute and Service Node Images Using the `cinstallman` Command" on page 211
- "Installing a Service Node with a Non-default Image" on page 213
- "Retrieving a Service Node Image from a Running Service Node" on page 214
- "Using a Custom Repository for Site Packages" on page 215
- "SGI ICE System Configuration Framework" on page 216
- "Cluster Configuration Repository: Updates on Demand" on page 219

Finding Which Distributions (distros) Are Supported

To find a list of distributions supported on SGI ICE nodes, perform the following commands from the system admin controller (admin node), as follows:

```
sys-admin:~ # cd /opt/sgi/share/rpmlists/distro/

sys-admin:/opt/sgi/share/rpmlists/distro # ls
compute-distro-centos5.4.rpmlist  lead-distro-rhel6.1.rpmlist
compute-distro-centos5.5.rpmlist  lead-distro-rhel6.2.rpmlist
compute-distro-centos6.0.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist    service-distro-centos5.5.rpmlist
```

compute-distro-rhel5.5.rpmlist	service-distro-centos6.0.rpmlist
compute-distro-rhel5.6.rpmlist	service-distro-rhel5.4.rpmlist
compute-distro-rhel5.7.rpmlist	service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist	service-distro-rhel5.6.rpmlist
compute-distro-rhel6.1.rpmlist	service-distro-rhel5.7.rpmlist
compute-distro-rhel6.2.rpmlist	service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist	service-distro-rhel6.1.rpmlist
compute-distro-sles10sp4.rpmlist	service-distro-rhel6.2.rpmlist
compute-distro-sles11sp1.rpmlist	service-distro-sles10sp3.rpmlist
lead-distro-centos6.0.rpmlist	service-distro-sles10sp4.rpmlist
lead-distro-rhel6.0.rpmlist	service-distro-sles11sp1.rpmlist

Operating Systems Supported per Node Type

This section describes what operating systems are supported for various SGI ICE nodes.

System Admin Controller

The system admin controller (admin node) supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Rack Leader Controller

The system admin controller (admin node) supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Service Nodes

Service nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5

Compute Nodes

Compute nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4

Compute Node Services Turned Off by Default

To improve the performance of applications running MPI jobs on compute nodes, most services are disabled by default in compute node images. To see what adjustments are being made, view the `/etc/opt/sgi/conf.d/80-compute-distro-services` script.

If you wish to change anything in this script, SGI suggests that you copy the existing script to `.local` and adjust it there. Perform the following commands:

```
# cd /var/lib/systemimager/images/compute-image-name
# cp etc/opt/sgi/conf.d/80-compute-distro-services 80-compute-distro-services.local
# vi etc/opt/sgi/conf.d/80-compute-distro-services.local
```

At this point, the configuration framework will execute the `.local` version, and skip the other. For more information on making adjustments to configuration framework files, see "SGI ICE System Configuration Framework" on page 216.

Use the `cimage` command to push the changed image out to the leader nodes.

crepo Command

You can use the `crepo` command to manage software repositories, such as, SGI Foundation, SMC for SGI ICE, SGI Performance Suite, and the Linux distribution(s) you are using on your system. You also use the `crepo` command to manage any custom repositories you create yourself.

The `configure-cluster` command calls the `crepo` command when it prompts you for media and then makes it available. You can also use the `crepo` command to add additional media.

Each repository has associated with it a name, directory, update URL, selection status, and suggested package lists. The update URL is used by the `sync-repo-updates` command. The directory is where the actual `yum` repository exists, and is located in one of these locations, as follows:

Repository	Description
<code>/tftpboot/sgi/*</code>	For SGI media
<code>/tftpboot/other/*</code>	For any media that is not from SGI
<code>/tftpboot/distro/*</code>	For Linux distribution repositories such as SLES or RHEL
<code>/tftpboot/x</code>	Customer-supplied repositories

The repository information is determined from the media itself when adding media supplied by SGI, Linux distribution media (SLES, RHEL, and so on.), and any other YaST-compatible media. For customer-supplied repositories, the information must be provided to the `crepo` command when adding the repository.

Repositories can be selected and unselected. Usually, SMC for SGI ICE commands ignore unselected repositories. One notable exception is that `sync-repo-updates` always operates on all repositories.

The `crepo` command constructs default RPM lists based on the suggested package lists. The RPM lists can be used by the `cininstallman` command when creating a new image. These RPM lists are only generated if a single distribution is selected and can be found in `/etc/opt/sgi/rpmlists`; they match the form `generated-*.rpmlist`. The `crepo` command will tell you when it updates or removes generated rpmlists. For example:

```
# crepo --select SUSE-Linux-Enterprise-Server-10-SP3
Updating: /etc/opt/sgi/rpmlists/generated-compute-sles10sp3.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-sles10sp3.rpmlist
```

When generating the RPM lists, the `crepo` command combines the a list of distribution RPMs with suggested RPMs from every other selected repository. The distribution RPM lists are usually read from the `/opt/sgi/share/rpmlists/distro` directory. For example, the compute node RPM list for `sles11sp1` is `/opt/sgi/share/rpmlists/distro/compute-distro-sles11sp1.rpmlist`. The suggested RPMs for non-distribution repositories are read from the `/var/opt/sgi/sgi-repodata` directory. For example, the rpmlist for SLES 11 SP1 compute nodes is read from `/var/opt/sgi/sgi-repodata/SMC-for-ICE 1.0-for-Linux-sles11/smc-ice-compute.rpmlist`.

The suggested rpmlists can be overridden by creating an override rpmlist in the `/etc/opt/sgi/rpmlists/override/` directory. For example, to change the default SMC for ICE 1.0 suggested rpmlist, a file `/etc/opt/sgi/rpmlists/override/SMC-for-ICE-1.0-for-Linux-sles11/smc-ice-compute.rpmlist` can be created.

The following example shows the contents of the `/etc/opt/sgi/rpmlists` directory after the `crepo` command has created the suggested RPM lists. Change directory (`cd`) to the `/etc/opt/sgi/rpmlists` directory. Use the `ls` command to see a list of rpms, as follows:

```
admin distro]# ls
compute-distro-centos5.4.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist   service-distro-rhel5.4.rpmlist
compute-distro-rhel5.5.rpmlist   service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist   service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist  service-distro-sles10sp3.rpmlist
compute-distro-sles11sp1.rpmlist  service-distro-sles11sp1.rpmlist
lead-distro-rhel6.0.rpmlist
```

Specifically, SMC for SGI ICE software looks for `/etc/opt/sgi/rpmlists/generate-*.rpm`list and creates an image for each rpmfile that matches.

It also determines the default image to use for each node type by hard-coding "`$nodeType-$distro`" as the type, where `distro` is the admin node's distro and `nodeType` is `compute`, `service`, `leader`, and so on. The default image can be overridden by specifying a global `cattr` attribute named `image_default_$nodeType`; for example, `image_default_service`. Use `cattr --h`, for information about the `cattr` command.

The following example shows the contents of the `/etc/opt/sgi/rpmlists` directory after the `crepo` command has created the suggested RPM lists. The files with `-distro-` in the name are the base Linux distro RPMs that SGI recommends.

Change directory (`cd`) to `/etc/opt/sgi/rpmlists`. Use the `ls` command to see a list of rpms, as follows:

```
admin:/etc/opt/sgi/rpmlists # ls
compute-minimal-sles11sp1.rpmfile  generated-lead-rhel6.0.rpmfile
generated-compute-rhel6.0.rpmfile  generated-service-rhel6.0.rpmfile
```

For more information on rpmfile customization information, see "Creating Compute and Service Node Images Using the `cininstallman` Command" on page 211.

You can use the `crepo --show` command to show the available repositories on the admin node, as follows:

```
sys-admin:~ # crepo --show
* SGI-Foundation-Software-2.4-sles11 : /tftpboot/sgi/SGI-Foundation-Software-2.4-sles11
* SGI-Tempo-2.4-sles11 : /tftpboot/sgi/SGI-Tempo-2.4-sles11
  SGI-Foundation-Software-2SP1-sles11 : /tftpboot/sgi/SGI-Foundation-Software-2SP1-sles11
* SUSE-Linux-Enterprise-Server-11-SP1 : /tftpboot/distro/sles11sp1
* SGI-Management-Center-1.4-sles11 : /tftpboot/sgi/SGI-Management-Center-1.4-sles11
  SGI-Tempo-2.1-for-Linux-sles11 : /tftpboot/sgi/SGI-Tempo-2.1-for-Linux-sles11
```

For a `crepo` command usage statement, perform the following:

```
admin:~ # crepo --h
crepo Usage:
Operations:
--help           : print his usage message

--add {path/URL} : add SGI/SMC media to the system repositories
```

3: System Operation

`--custom {name}`: Optional. Use with `--add` to add custom repo under `/tftpboot`. Repo must pre-exist for this case.

`--del {product}` : delete an add-on product and associated `/tftpboot` repo

`--select {product}` : mark the product as selected

`--show` : show available add-on products

`--show-distro` : like show, but only reports distro media like `sles10sp2`

`--show-updateurls` : Show the update sources associated add-on products

`--reexport` : re-export all repositories with yume. Use if there was a yume export problem previously.

`--unselect {product}` : mark the product as not selected

Flags:

Note for `--add`: If the pathname is local to the machine, it can be an ISO file or mounted media. If a network path is used -- such as an nfs path or a URL -- the path must point to an ISO file. The argument to `--add` may be a comma delimited list to specify multiple source media.

Use `--add` for SGI/SMC media, to make the repos and rpms available. If the supplied SGI/SMC media has suggested rpms from SMC node types, those suggested rpms will be integrated with the default rpmlists for leader, service, and compute nodes. You can use `create-default-sgi-images` to re-create the default images including new suggested packages or you can just browse the updated versions in `/etc/opt/sgi/rpmlists`.

Use `--add` with `--custom` to register your own custom repository. This will ensure that, by default, the custom repository is available to yume and `mksiiimage` commands. It is assumed you will maintain your own default package lists, perhaps using the sgi default package lists in `/etc/opt/sgi/rpmlists` or `/opt/sgi/share/rpmlists` as a starting point. The directory and rpms within must pre-exist. This script will create the yum metadata for it.

Example:

```
crepo --add /tftpboot/myrepo --custom my-custom-name
```

`cinstallman` Command

The `cinstallman` command is a wrapper tool for several SMC for SGI ICE operations that previously ran separately. You can use the `cinstallman` command to perform the following:

- Create an image from scratch
- Clone an existing image
- Recreate an image (so that any nodes associated with said image prior to the command are also associated after)
- Use existing images that may have been created by some other means
- Delete images
- Show available images
- Update or manage images (via `yume`)
- Update or manage nodes (via `yume`)
- Assign images to nodes
- Choose what a node should do next time it reboots (image itself or boot from its disk)
- Refresh the `bittorrent` `tarball` and `torrent` file for a compute node image after making changes to the expanded image

When compute images are created for the first time, a `bittorrent` `tarball` is also created. When images are pushed to rack leaders for the first time, `bittorrent` is used to transport the `tarball` snapshot of the image. However, as you make adjustments to your compute image, those changes do not automatically generate a new `bittorrent` `tarball`. We handle that situation by always doing a follow-up `rsync` of the compute image after transporting the `tarball`. However, as your compute image begins to diverge from the `bittorrent` `tarball` snapshot, it becomes less and less efficient to transport a given compute node image that is new to a given rack leader.

You no longer need to use `yum`, `yume`, or `mksiimage` commands directly for most common operations. Compute images are automatically configured in such a way as to make them available to the `cimage` command.

For a cinstallman command usage statement, perform the following:

```
admin:~ # cinstallman --help
```

```
cinstallman Usage:
```

```
cinstallman is a tool that manages:
```

- image creation (as a wrapper to mksiimage)
- node package updates (as a wrapper to yume)
- image package updates (yume within a chroot to the image)

This is a convenience tool and not all operations for the commands that are wrapped are provided. The most common operations are collected here for ease of use.

For operations that take the `--node` parameter, the node can be an aggregation of nodes like `cimage` and `cpower` can take. Depending on the situation, non-managed or offline nodes are skipped.

The tool retrieves the registered repositories from `crepo` so that they need not be specified on the command line.

Operations:

```
--help           : print his usage message
--create-image   : create a new systemimager image
                  By default, requires --rpmfile and --image
                  Optional flags below:
--clone          : Clone existing image, requires --source, --image.
                  Doesn't require --rpmfile.
--recreate       : Like --del-image then --add-image, but preserves any
                  node associations.
                  Requires --image and --rpmfile
--repos {list}   : A comma-separated list of repositories to use.
--use-existing   : register an already existing image, doesn't
                  require --rpmfile
--image {image}  : Specify the image to operate on
--rpmfile {path} : Provide the rpmfile to use when creating images
--source {image} : Specify a source image to operate on (for clone)

--del-image      : delete the image, may use with --del-nodes
--image {image}  : Specify the image to operate on
```

```
--show-images          : List available images (similar to mksiimage -L)

--show-nodes           : Show non-compute nodes (similar to mksimachine -L)

--update-image         : update packages in image to latest packages available
                        : in repos, Requires --image
  --image {image}      : Specify the image to operate on

--refresh-image        : Refresh the given image to include all packages
                        : in the supplied rpmlist. Use after registering
                        : new media with crepo that has new suggested rpms.
  --image {image}      : Specify the node or nodes to operate on
  --rpmlist {path}     : rpmlist containing packages to be sure are included

--yum-image            : Perform yum operations to supplied image, via yume
                        : Requires --image, trailing arguments passed to yume
  --image {image}      : Specify the image to operate on

--update-node          : Update supplied node to latest pkgs avail in
                        : repos, requires --node
  --node {node}        : Specify the node or nodes to operate on

--refresh-node         : Refresh the given node to include all packages
                        : in the supplied rpmlist. Use after registering
                        : new media with crepo that has new suggested rpms.
  --node {node}        : Specify the node or nodes to operate on
  --rpmlist {path}     : rpmlist containing packages to be sure are included

--yum-node             : Perform yum operations to nodes, via yume. Requires
                        : --node. Trailing arguments passed to yume
  --node {node}        : Specify the node or nodes to operate on

--assign-image         : Assign image to node. Requires --node, --image
  --node {node}        : Specify the node or nodes to operate on
  --image {image}      : Specify the image to operate on

--next-boot {image|disk} : node action next boot: boot from disk or
                        : reinstall/reimage? Requires --node

--refresh-bt           : Refresh the bittorrent tarball and torrent file
                        : Requires --image
```

`--image {image}` : Specify the image to operate on

In the following example, the `--refresh-node` operation is used to ensure the online managed service nodes include all the packages in the list. You could use this if you updated your `rpmlist` to include new packages or if you recently added new media with the `crepo` command and want running nodes to have the newly updated packages. A similar `--refresh-image` operation exists for images.

```
# cinstallman --refresh-node --node service\* --rpmlist
/etc/opt/sgi/rpmlists/service-sles11.rpmlist
```

Customizing Software On Your SGI ICE System

This section discusses how to manage various nodes on your SGI ICE system. It describes how to configure the various nodes, including the compute and service nodes. It describes how to augment software packages. Many tasks having to do with package management have multiple valid methods to use.

For information on installing patches and updates, see "Installing SMC for SGI ICE Patches and Updating SGI ICE Systems " on page 125.

Creating Compute Node Custom Images

You can add per-host compute node customization to the compute node images. You do this by adding scripts either to the `/opt/sgi/share/per-host-customization/global/` directory or the `/opt/sgi/share/per-host-customization/mynewimage/` directory on the system admin controller.

Note: When creating custom images for compute nodes, make sure you clone the original SGI images. This provides the original images intact that you can fall back to if necessary. The following example is based on SLES.

Scripts in the global directory apply to all compute nodes images. Scripts under the image name apply only to the image in question. The scripts are cycled through once per host when being installed on the rack leader controllers. They receive one input argument, which is the full path (on the rack leader controller) to the per-host base directory, for example, `/var/lib/sgi/mynewimage/i2n11`. There is a `README` file

at `/opt/sgi/share/per-host-customization/README` on the system admin controller, as follows:

This directory contains compute node image customization scripts which are executed as part of the install-image operations on the leader nodes when pulling over a new compute node image.

After the image has been pulled over, and the `per-host-customization` dir has been rsynced, the `per-host /etc` and `/var` directories are populated, then the scripts in this directory are cycled through once per-host. This allows the scripts to source the node specific network and cluster management settings, and set node specific settings.

Scripts in the global directory are iterated through first, then if a directory exists that matches the image name, those scripts are iterated through next.

You can use the scripts in the global directory as examples.

An example global script,

`/opt/sgi/share/per-host-customization/global/sgi-fstab` is, as follows:

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
#
# Set up the compute node's /etc/fstab file.
```

3: System Operation

```
#
# Modify per your sites requirements.
#
# This script is executed once per-host as part of the install-image
operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#
# sanity checks
. /opt/sgi/share/per-host-customization/global/sanity.sh

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n '/^compat
/s/^compat.*: //p' )
compatdistro=${os[0]}${os[1]}

if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then

#
# SLES 10 compatible
#
cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
tmpfs /tmp tmpfs size=150m 0 0
EOF

elif [ ${compatdistro} = "rhel5" ]; then

#
# RHEL 5 compatible
#
#
# RHEL expects several subsys directories to be present under
/var/run
# and /var/lock, hence no tmpfs mounts for them
#
cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
```

```

tmpfs          /tmp          tmpfs   size=150m     0      0
devpts         /dev/pts     devpts  gid=5,mode=620 0      0
EOF

else

    echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi

```

Modify Compute Image Kernel Boot Options

You can use the `cattr` command to set extra kernel boot parameters for compute nodes on a per-image basis. For example to append "cgroup_disable=memory" to kernel boot parameters for any node booting the "compute-sles11sp1" image, perform a command similar to the following:

```

% cattr set kernel_extra_params-compute-sles11sp1 cgroup_disable=memory
    Push the image, as follows:

    # cimage --push-rack mynewimage r1

```

Compute Node Per-Host Customization for Additional Network Interfaces

Note: The following example is only for systems running SLES.

Per compute-node customization may be useful for configuring additional network interfaces that are on some, but not all, compute nodes. An example of how to configure network interfaces on individual compute nodes is the `/opt/sgi/share/per-host-customization/mynewimage/mycustomization` script, that follows:

```

Copyright (c) 2008 Silicon Graphics, Inc.
# All rights reserved.
#
# do node specific setup
#
# This script is executed once per-host as part of the install-image operation

```

3: System Operation

```
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $ARGV[0],
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

use lib "/usr/lib/systemconfig", "/opt/sgi/share/per-host-customization/global";
use sanity;

sanity_checks();

$blade_path = $node = $ARGV[0];
$node =~ s/.*\///;

sub i0n4 {
    my $ifcfg="etc/sysconfig/network/ifcfg-eth2";
    open(IFCFG, ">$blade_path/$ifcfg") or
        die "$0: can't open $blade_path/$ifcfg";
    print IFCFG<<EOF
BOOTPROTO='static'
IPADDR='10.20.0.1'
NETMASK='255.255.0.0'
STARTMODE='onboot'
WIRELESS='no'
EOF
        ;
    close(IFCFG);
}

@nodes = ("i0n4");

foreach $n (@nodes) {
    if ( $n eq $node ) {
        eval $n;
    }
}
```

Pushing `mynewimage` to rack 1 causes the `eth2` interface of compute node `r1i0n4` to be configured with IP address `10.20.0.1` when the node is brought up with `mynewimage`. Push the image, as follows:

```
# cimage --push-rack mynewimage r1
```

Customizing Software Images

Note: Procedures in this section describe how to work with service node and compute node images. Always use a cloned image. If you are adjusting an RPM list, use your own copy of the RPM list.

The service and compute node images are created during the `configure-cluster` operation (or during your upgrade from a prior release). This process uses an RPM list to generate a root on the fly.

You can clone a compute node image, or create a new one based on an RPM list. For service nodes, SGI does not support a clone operation. For compute images, you can either clone the image and work on a copy or you can always make a new compute node image from the SGI supplied default RPM list.

Procedure 3-1 Creating a Simple Compute Node Image Clone

Note: Always work from a clone image, see "Customizing Software Images" on page 203.

To create a simple compute node image clone from the system admin controller, perform the following steps:

1. To clone the compute node image, perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image compute-sles11-new
```

2. To see the images and kernels in the list, perform the following:

```
# cimage --list-images
image: compute-sles11
      kernel: 2.6.27.19-5-smp

image: compute-sles11-new
      kernel: 2.6.27.19-5-smp
```

3. To push the compute node image out to the rack, perform the following:

```
# cimage --push-rack compute-sles11-new r\*
```

4. To change the compute nodes to use the cloned image/kernel pair, perform the following:

```
# cimage --set compute-sles11-new 2.6.27.19-5-smp "r*i*n"
```

Procedure 3-2 Manually Adding a Package to a Compute Node Image

To manually add a package to a compute node image, perform the steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "cinstallman Command" on page 195.

1. Make a clone of the compute node image, as described in "Customizing Software Images" on page 203.
- 2.

Note: This example shows SLES11.

Determine what images and kernels you have available now, as follows:

```
# cimage --list-images
image: compute-sles11
      kernel: 2.6.27.19-5-smp

image: compute-sles11-new
      kernel: 2.6.27.19-5-smp
```

3. From the system admin controller, change directory to the images directory, as follows:

```
# cd /var/lib/systemimager/images/
```

4. From the system admin controller, copy the RPMs you wish to add, as follows, where `compute-sles11-new` is your own compute node image, as follows:

```
# cp /tmp/newrpm.rpm compute-sles11-new/tmp
```

5. The new RPMs now reside in `/tmp` directory in the image named `compute-sles11-new`. To install them into your new compute node image, perform the following commands:

```
# chroot compute-sles11-new bash
```

And then perform the following:

```
# rpm -Uvh /tmp/newrpm.rpm
```

At this point, the image has been updated with the RPM.

6. The image on the system admin controller is updated. However, you still need to push the changes out. Ensure there are no nodes currently using the image and then run this command:

```
# cimage --push-rack compute-sles11-new r\*
```

This will push the updates to the rack lead controllers and the changes will be seen by the compute nodes the next time they start up. For information on how to ensure the image is associated with a given node, see the `cimage --set` command and the example in Procedure 3-1, page 203.

Procedure 3-3 Manually Adding a Package to the Service Node Image

To manually add a package to the service node image, perform the following steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "cinstallman Command" on page 195.

1. Use the `cinstallman` command to create your own version of the service node image. See "cinstallman Command" on page 195.
2. Change directory to the `images` directory, as follows:

```
# cd /var/lib/systemimager/images/
```

3. From the system admin controller, copy the RPMs you wish to add, as follows, where `my-service-image` is your own service node image:

```
# cp /tmp/newrpm.rpm my-service-image/tmp
```

4. The new RPMs now reside in `/tmp` directory in the image named `my-service-image`. To install them into your new service node image, perform the following commands:

```
# chroot my-service-image bash
```

And then perform the following:

```
# rpm -Uvh /tmp/newrpm.rpm
```

At this point, the image has been updated with the RPM. Please note, that unlike compute node images, changes made to a service node image will not be seen by service nodes until they are reinstalled with the image. If you wish to install the package on running systems, you can copy the RPM to the running system and use the RPM from there.

cimage Command

The `cimage` command allows you to list, modify, and set software images on the compute nodes in your system.

For a help statement, perform the following command:

```
admin:~ # cimage --help
cimage is a program for managing compute node root images in SMC for ICE.
```

```
Usage: cimage OPTION ...
```

Options

<code>--help</code>	Usage and help text.
<code>--debug</code>	Output additional debug information.
<code>--list-images</code>	List images and their kernels.
<code>--list-nodes NODE</code>	List node(s) and what they are set to.
<code>--set [OPTION] IMAGE KERNEL NODE</code>	Set node(s) to image and kernel.
<code>--nfs</code>	Use NFS roots (default).
<code>--tmpfs</code>	Use tmpfs roots.
<code>--set-default [OPTION] IMAGE KERNEL</code>	Set default image, kernel, rootfs type.
<code>--nfs</code>	Use NFS roots (default).

<code>--tmpfs</code>	Use tmpfs roots.
<code>--show-default</code>	Show default image, kernel, rootfs type.
<code>--add-db IMAGE</code>	Add image and its kernels to the db.
<code>--del-db IMAGE</code>	Delete image and its kernels from db.
<code>--update-db IMAGE</code>	Short-cut for <code>--del-db</code> , then <code>--add-db</code> .
<code>--push-rack [OPTIONS] IMAGE RACK</code>	Push or update image on rack(s).
<code>--force</code>	Bypass the booted nodes check, deletes.
<code>--update-only</code>	Skip files newer in dest, no delete.
<code>--quiet</code>	Turn off diagnostic information.
<code>--del-rack IMAGE RACK</code>	Delete an image from rack(s).
<code>--clone-image OIMAGE NIMAGE</code>	Clone an existing image to a new image.
<code>--del-image [OPTIONS] IMAGE</code>	Delete an existing image entirely.
<code>--quiet</code>	Turn off diagnostic information.

RACK arguments take the format 'rX'

NODE arguments take the format 'rXiYnZ'

ROOTFS argument can be either 'nfs' or 'tmpfs'

X, Y, Z can be single digits, a [start-end] range, or * for all matches.

EXAMPLES

Example 3-1 cimage Command Examples

The following examples walk you through some typical `cimage` command operations.

To list the available images and their associated kernels, perform the following:

```
# cimage --list-images
image: compute-sles11
      kernel: 2.6.27.19-5-carlsbad
      kernel: 2.6.27.19-5-default
image: compute-sles11-1_7
      kernel: 2.6.27.19-5-default
```

To list the compute nodes in rack 1 and the image and kernel they are set to boot, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-default nfs
r1i0n8: compute-sles11 2.6.27.19-5-default nfs
```

The `cimage` command also shows the root filesystem type (nfs or tmpfs)

To set the `r1i0n0` compute node to boot the `2.6.27.19-5-smp` kernel from the `compute-sles11` image, perform the following: :

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i0n0
```

To list the nodes in rack 1 to see the changes set in the example above, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-smp
r1i0n1: compute-sles11 2.6.27.19-5-smp
r1i0n2: compute-sles11 2.6.27.19-5-smp
[...snip...]
```

To set all nodes in all racks to boot the `2.6.27.19-5-smp` kernel from the `compute-sles11` image, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r*i*n*
```

To set two ranges of nodes to boot the `2.6.27.19-5-smp` kernel, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the `compute-sles11` image to a new image (so that you can modify it) , perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image mynewimage
Cloning compute-sles11 to mynewimage ... done
```

The clone process adds the image and its kernels to the database.

Note: If you have made changes to the compute node image and are pushing that image out to leader nodes, it is a good practice to use the `cinstallman --refresh-bt --image {image}` command to refresh the bittorrent tarball and torrent file for a compute node image. This avoids duplication by `rsync` when the image is pushed out to the leader nodes. For more information, see the `cinstallman --h usage statement` or "cinstallman Command" on page 195.

To change to the cloned image created in the example, above, copy the needed rpms into the `/var/lib/systemimager/images/mynewimage/tmp` directory, use the

`chroot` command to enter the directory and then install the rpms, perform the following:

```
# cp *.rpm /var/lib/systemimager/images/mynewimage/tmp
# chroot /var/lib/systemimager/images/mynewimage/ bash
# rpm -Uvh /tmp/*.rpm
```

If you make changes to the kernels in the image, you need to refresh the kernel database entries for your image. To do this, perform the following:

```
# cimage --update-db mynewimage
```

If you did not make changes to the kernels in the cloned image created in the example above, you can omit this step.

To push new software images out to the compute blades in a rack or set of racks, perform the following:

```
# cimage --push-rack mynewimage r*
r1lead: install-image: mynewimage
r1lead: install-image: mynewimage done.
```

To list images in the database the kernels they contain, perform the following:

```
# cimage --list-images

image: compute-sles11
      kernel: 2.6.16.60-0.7-carlsbad
      kernel: 2.6.16.60-0.7-smp

image: mynewimage
      kernel: 2.6.16.60-0.7-carlsbad
      kernel: 2.6.16.60-0.7-smp
```

To set some compute nodes to boot an image, perform the following:

```
# cimage --set mynewimage 2.6.16.60-0.7-smp r1i3n*
```

You need to reboot the compute nodes to run the new images.

Completely remove an image you no longer use, both from system admin controller and all compute nodes in all racks, perform the following:

```
# cimage --del-image mynewimage
r1lead: delete-image: mynewimage
```

```
rllead: delete-image: mynewimage done.
```

Using `cinstallman` to Install Packages into Software Images

The packages that make up SMC for SGI ICE, SGI Foundation, and the Linux distribution media, and any other media or custom repositories you have added reside in repositories. The `cinstallman` command looks up the list of all repositories and provides that list to the commands it calls out for its operation such as `yume`.

Note: Always work with copies of software images.

The `cinstallman` command can update packages within `systemimager` images. You may also use `cinstallman` to install a single package within an image.

However, `cinstallman` and the commands it calls only works with the configured repositories. So if you are installing your own RPM, you will need that package to be part of an existing repository. You may use the `crepo` command to create a custom repository into which you can collect custom packages.

Note: The `yum` command maintains a cache of the package metadata. If you just recently changed the repositories, `yum` caches for the nodes or images you are working with may be out of date. In that case, you can issue the `yum` command "clean all" with `--yum-node` and `--yum-image`. The `cinstallman` command `--update-node` and `--update-image` options do this for you.

The following example shows how to install the `zlib-devel` package in to the service node image so that the next time you image or install a service node, it will have this new package.

```
# cinstallman --yum-image --image my-service-sles11 install zlib-devel
```

You can perform a similar operation for compute node images. Note the following:

- If you update a compute node image on the system admin controller (admin node), you have to use the `cimage` command to push the changes. For more information on the `cimage` command, see "cimage Command" on page 206.
- If you update a service node image on the admin node, that service node needs to be reinstalled and/or reimaged to get the change. The `discover` command can be given an alternate image or you may use the `cinstallman --assign-image`

command followed by the `cinstallman --next-boot` command to direct the service node to reimage itself with a specified image the next time it boots.

Using `yum` to Install Packages on Running Service or Leader Nodes

Note: These instructions only apply to managed service nodes and leader nodes. They do not apply to compute nodes.

You can use the `yum` command to install a package on a service node. From the admin node, you can issue a command similar to the following:

```
# cinstallman --yum-node --node service0 install zlib-devel
```

Note: To get all service nodes, replace `service0` with `service*`.

For more information on the `cinstallman` command, see "cinstallman Command" on page 195.

Creating Compute and Service Node Images Using the `cinstallman` Command

You can create service node and compute node images using the `cinstallman` command. This generates a root directory for images, automatically.

Fresh installations of SMC for SGI ICE create these images during the `configure-cluster` installation step (see "Installing SMC for SGI ICE Admin Node Software").

The RPM lists that drive which packages get installed in the images are listed in files located in `/etc/opt/sgi/rpmlists`. For example, `/etc/opt/sgi/rpmlists/compute-sles11.rpmlist` (see "crepo Command" on page 191). You should **NOT** edit the default lists. These default files are recreated by the `crepo` command when repositories are added or removed. Therefore, you should only use the default RPM lists as a model for your own.

Note: The procedure below uses SLES.

Procedure 3-4 Using the `cinstallman` Command to Create a Service Node Image:

To create a service node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the example service node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/service-sles11.rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.
3. Use the `cinstallman` command with the `--create-image` option to create the images root directory, as follows:

```
# cinstallman --create-image --image my-service-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

This example uses `my-service-node-image` as the home/name of the image.

Output is logged to `/var/log/cinstallman` on the admin node.

4. After the `cinstallman` command finishes, the image is ready to be used with service nodes. You can supply this image as an optional image name to the `discover` command, or you may assign an existing service node to this image using the `cinstallman --assign-image` command. You can tell a service node to image itself next reboot by using the `cinstallman --next-boot` option.

Procedure 3-5 Use the `cinstallman` Command to Create a Compute Node Image

To create a compute node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the compute node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/compute-sles11.rpmlist
/etc/opt/sgi/rpmlists/my-compute-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.

3. Run the `cinstallman` command to create the root, as follows:

```
# cinstallman --create-image --image my-compute-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-compute-node.rpmlist
```

This example uses the name `my-compute-node-image` as the name.

Output is logged to `/var/log/cinstallman` on the admin node.

The `cinstallman` command makes the new image available to the `cimage` command.

4. For information on how to use the `cimage` command to push this new image to rack leader controllers (leader nodes), see "cimage Command" on page 206.

Installing a Service Node with a Non-default Image

If you have a non-default service node image you wish to install on a service node, you have two choices, as follows:

- Specify the image name when you first discover the node with the `discover` command.
- Use the `cinstallman` command to associate an image with a service node, then set up the node to reinstall itself the next time it boots.

The following example shows how to associate a custom image at discover time:

```
# discover --service 2,image=my-service-node-image
```

The next example shows how to reinstall an already discovered service node with a new image:

```
# cinstallman --assign-image --node service2 --image my-service-node-image
# cinstallman --next-boot image --node service2
```

When you reboot the node, it will reinstall itself.

For more information on the `discover` command, see "discover Command" on page 80. For more information on the `cinstallman` command, see "cinstallman Command" on page 195.

Retrieving a Service Node Image from a Running Service Node

To retrieve a service node image from a running service node, perform the following steps:

1. As **root user**, log into the service node from which you wish to retrieve an image. You can use the `si_prepareclient(8)` program to extract an image. Start the program, as follows:

```
service0:~ # si_prepareclient --server admin
```

```
Welcome to the SystemImager si_prepareclient command. This command may modify the following files to prepare your golden client for having its image retrieved by the imageserver. It will also create the /etc/systemimager directory and fill it with information about your golden client. All modified files will be backed up with the .before_systemimager-3.8.0 extension.
```

```
/etc/services:
```

```
This file defines the port numbers used by certain software on your system. Entries for rsync will be added if necessary.
```

```
/tmp/filet10eP5:
```

```
This is a temporary configuration file that rsync needs on your golden client in order to make your filesystem available to your SystemImager server.
```

```
inetd configuration:
```

```
SystemImager needs to run rsync as a standalone daemon on your golden client until its image is retrieved by your SystemImager server. If rsyncd is configured to run as a service started by inetd, it will be temporarily disabled, and any running rsync daemons or commands will be stopped. Then, an rsync daemon will be started using the temporary configuration file mentioned above.
```

See "si_prepareclient --help" for command line options.

```
Continue? (y/[n]):
```

Enter **y** to continue. After a few moments, you are returned to the command prompt. You are now ready to retrieve the image from the admin node.

2. Exit the **service0** node, and as **root user** on the admin node, perform the following command: (Replace the image name and service node name, as needed.)

```
admin # mksiimage --Get --client service0 --name myimage
```

It now retrieves the image. No progress information is provided. It takes several minutes depending on the size of the image on the service node.

3. Use the `cinstallman` command to register the newly collected image:

```
admin # cinstallman --create --use-existing --image myimage
```

4. If you want to discover a node using this image directly, you can use the `discover` command, as follows:

```
admin # discover --service 0,image=myimage
```

5. If you want to re-image an already discovered node with your new image, run the following commands:

```
# cinstallman --assign-image --node service0 --image myimag  
# cinstallman --next-boot image --node service0
```

6. Reboot the service node.

Using a Custom Repository for Site Packages

This section describes how to maintain packages specific to your site and have them available to the `crepo` command (see "crepo Command" on page 191).

SGI suggests putting site-specific packages in a separate location. They should not reside in the same location as SGI or Novell supplied packages.

Procedure 3-6 Setting Up a Custom Repository for Site Packages

To set up a custom repository for your custom packages, perform the following steps:

1. Create directory for your site-specific packages on the system admin controller (admin node), as follows:

```
# mkdir -p /tftpboot/site-local/sles-10-x86_64
```

2. Copy your site packages in to the new directory, as follows:

```
# cp my-package-1.0.x86_64.rpm /tftpboot/site-local/sles-10-x86_64
```

3. Register your custom repository using the `crepo` command. This command will ensure your repository is consulted when the `installman` command performs its operations. This command also creates the necessary `yum/repomd` metadata.

```
# crepo --add /tftpboot/site-local/sles-10-x86_64 --custom my-repo
```

Your new repository may be consulted by `installman` command operations going forward including updating images, nodes, and creating images.

4. If you wish this repository to be used by `installman` by default, you need to select it. Use the following command:

```
# crepo --select my-repo
```

5. If you use `installman` to create an image, you will want to add your custom package to the `rpmlist` you use with the `installman` command (see "Using `installman` to Install Packages into Software Images" on page 210).

SGI ICE System Configuration Framework

All node types that are part of an SGI ICE system can have configuration settings adjusted by the configuration framework. There is some overlap between the per-host customization instructions and the configuration framework instructions. Each approach plays a role in configuring your system. The major differences between the two methods are, as follows:

- Per-host customization runs at the time an image is pushed to the rack leader controllers.
- Per-host customization only applies to compute node images.
- The SGI ICE system configuration framework can be used with all node types.
- The system configuration framework is run when a new root is created, when `SUSEconfig` command is run for some other reason, as part of a `yum` operation, or when new compute images are pushed with the `cimage` command.

This framework exists to make it easy to adjust configuration items. There are SGI-supplied scripts already present. You can add more scripts as you wish. You can also exclude scripts from running without purging the script if you decide a certain script should not be run. The following set of questions in bold and bulleted answers describes how to use the system configuration framework.

How does the system configuration framework operate?

These files could be added, for example, to a running service node, or to an already created service or compute image. Remember that images destined for compute nodes need to be pushed with the `cimage` command after being altered. For more information, see "cimage Command" on page 206.

- A `/opt/sgi/lib/cluster-configuration` script is called, from where it is called is described below.
- That script iterates through scripts residing in `/etc/opt/sgi/conf.d`.
- Any scripts listed in `/etc/opt/sgi/conf.d/exclude` are skipped, as are scripts, that are not executable.
- Scripts in system configuration framework **must** be tolerant of files that do not exist yet, as described below. For example, check that a `syslog` configuration file exists before trying to adjust it.
- Scripts ending in a distro name, or a distro name with a specific distro version are only run if the node in question is running that distro. For example, `/etc/opt/sgi/conf.d/99-foo.sles` would only run if the node was running `sles`. This example shows the precedence of operations: If you had `88-myscript.sles10`, `88-myscript.sles`, and `88-myscript`
 - On a `sles10` system, `88-myscript.sles10` would execute
 - On a `sles` system that is not `sles10`, `88-myscript.sles` would execute
 - On all other distros, `88-myscript` would execute
- If you wish to make a custom version of a script supplied by SGI, you may simply name it with `.local` and the local version will run in place of the one supplied by SGI. This allows for customization without modifying scripts supplied by SGI. Scripts ending in `.local` have the highest precedence. In other words, if you had `88-myscript.sles`, and `88-myscript.local`, then `88-myscript.local` would execute in all cases and the other `88-myscript` scripts would never execute.

From where is the framework called?

- The callout for `/opt/sgi/lib/cluster-configuration` is implemented as a `yum` plugin that executes after packages have been installed and cleaned.
- On SLES only, there is also a SUSE configuration script in the `/sbin/conf.d` directory, called `SuSEconfig.00cluster-configuration`, that calls the framework. This is in case of you are using YaST to install or upgrade packages.

- On SLES only, one of the scripts called by the framework calls `SuSEconfig`. A check is made to avoid a callout loop.
- The framework is also called when the admin, leader, or service nodes start up. The call is made just after networking is configured. As a site administrator, you could create custom scripts here that check on or perform certain configuration operations.
- When using the `cimage` command to push a compute node root image to rack leaders, the configuration framework executes within the `chroot` of the compute node image after it is pulled from the admin node to the rack leader node.

How do I adjust my system configuration?

- Create a small script in `/etc/opt/sgi/conf.d` to do the adjustment.

Be sure that you test for existence of files and do not assume they are there (see "Why do scripts need to tolerate files that do not exist but should?" below).

Why do scripts need to tolerate files that do not exist but should?

- This is because the `mksiimage` command runs `yume` and `yum` in two steps. The first step only installs 40 or so RPMs but our framework is called then too. The second pass installs the other "hundreds" of RPMs. So the framework is called once before many packages are installed, and again after everything is in place. So not all files you expect might be available when your small script is called.

How does the yum plugin work?

- In order for the `yum` plugin to work, the `/etc/yum.conf` file has to have `plugins=1` set in its configuration file. SMC for SGI ICE software ensures that is in place by way of a trigger in the `sgi-cluster` package. Any time `yum` is installed or updated, it verify `plugins=1` is set.

How does yume work?

- `yume`, an oscar wrapper for `yum`, works by creating a temporary `yum` configuration file in `/tmp` and then points `yum` at it. This temporary configuration file needs to have plugins enabled. A tiny patch to `yume` makes this happen. This fixes it for `yume` and also `mksiimage`, which calls `yume` as part of its operation.

Cluster Configuration Repository: Updates on Demand

SMC for ICE contains a cluster configuration repository/update framework. This framework generates and distributes configuration updates to admin, service, and leader nodes in the cluster. Some of the configuration files managed by this framework include C3 conserver, DNS, Ganglia, hosts files, and NTP.

When an event occurs that requires these files to be updated, the framework executes on the admin node. The admin node stores the updated configuration framework in a special cached location and updates the appropriate nodes with their new configuration files.

In addition to the updates happening as required, the configuration file repository is consulted when a admin, service, or leader node boots. This happens shortly after networking is started. Any configuration files that are new or updated are transferred at this early stage so that the node is fully configured by the time the node is fully operational.

There are no hooks for customer configuration in the configuration repository at this time.

This update framework is tied in with the `/etc/opt/sgi/conf.d` configuration framework to provide a full configuration solution. As mentioned earlier, customers are encouraged to create `/etc/opt/sgi/conf.d` scripts to do cluster configuration.

`cnodes` Command

The `cnodes` command provides information about the types of nodes in your system. For help information, perform the following:

```
[admin ~]# cnodes --help
Options:
--all                all compute, leader and service nodes, and switches
--compute           all compute nodes
--leader            all leader nodes
--service           all service nodes
--switch            all switch nodes
--online            modifier: nodes marked online
--offline           modifier: nodes marked offline
--managed           modifier: managed nodes
--unmanaged         modifier: unmanaged nodes
--smc-for-ice-names modifier: return SMC-for-ICE node names instead of hostnames
```

Note: default modifiers are 'online' and 'managed' unless otherwise specified.

EXAMPLES

Example 3-2 `cnodes` Example

The following examples walk you through some typical `cnodes` command operations.

To see a list of all nodes in your system, perform the following:

```
[admin ~]# cnodes --all
rli0n0
rli0n1
r1lead
service0
```

To see a list of all compute nodes, perform the following:

```
[admin ~]# cnodes --compute
rli0n0
rli0n1
```

To see a list of service nodes, perform the following:

```
[admin ~]# cnodes --service
service0
```

Power Management Commands

The `cpower` command allows you to power up, power down, reset, and show the power status of system components.

`cpower` Command

The `cpower` command is, as follows:

```
cpower [<option> ...] [<target_type>] [<action>] <target>
```

The *<option>* argument can be one or more of the following:

Option	Description
<code>--noleader</code>	Do not include leader nodes (valid with rack and system domains only).

<code>--noservice</code>	Do not include service nodes (valid with system domain only).
<code>--force</code>	When using wildcards in the target, disable all “safety” checks. Make sure you really want to use this command.
<code>-n, --noexec</code>	Displays, but does not execute, commands that affect power.
<code>-v, --verbose</code>	Print additional information on command progress

The *<target>* argument is one of the following:

<code>--node</code>	Applies the action to nodes. Nodes are compute nodes, rack leader controllers (leader nodes), system admin controller (admin node), and service nodes. [default]
<code>--iru</code>	Applies the action at the IRU level.
<code>--rack</code>	Applies the action at the rack level.
<code>--system</code>	Applies the action to the system. You must not specify a target with this type.

The *<action>* argument is one of the following:

<code>--status</code>	Show the power status of the target, including whether it is booted or not. [default]
<code>--up --on</code>	Powers up the target.
<code>--down --off</code>	Powers down the target.
<code>--reset</code>	Performs a hard reset on the target.
<code>--cycle</code>	Power cycles the target.
<code>--boot</code>	Boots up the target, unless it is already booted. Waits for all targets to boot.
<code>--reboot</code>	Reboots the target, even if already booted. Wait for all targets to boot.
<code>--halt</code>	Halts and then powers off the target.
<code>--shutdown</code>	Shuts down the target, but does not power it off. Waits for targets to shut down.

<code>--identify</code> <code><interval></code>	Turns on the identifying LED for the specified interval in seconds. Uses an interval of 0 to turn off immediately.
<code>-h, --help</code>	Shows help usage statement.

The target must always be specified except when the `--system` option is used. Wildcards may be used, but be careful **not** to accidentally power off or reboot the leader nodes. If wildcard use affects any leader node, the command fails with an error.

Operations on Nodes

The default for the `cpower` command is to operate on system nodes, such as compute nodes, leader nodes, or service nodes. If you do not specify `--iru`, `--rack`, or `--system`, the command defaulted to operating as if you had specified `--node`.

Here are examples of node target names:

- `r1i3n10`
Compute node at rack 1, IRU 3, slot 10
- `service0`
Service node 0
- `r3lead`
Rack leader controller (leader node) for rack 3
- `r1i*n*`
Wildcards let you specify ranges of nodes, for example, `r1i*n*` all compute nodes in all IRUs on rack 1

IPMI-style Commands

The default operation for the `cpower` command is to operate on nodes and to provide you the status of these nodes, as follows:

```
# cpower r1i*n*
```

This command is equivalent to the following:

```
# cpower --node --status rli*n*
```

This command issues an `ipmitool power off` command to all of the nodes specified by the wildcard, as follows:

```
# cpower --off r2i*n*
```

The default is to apply to a node.

The following commands behave exactly as you would expect as if you were using `ipmitool`, and have no special extra logic for ordering:

```
# cpower --up rli*n*
```

```
# cpower --reset rli*n*
```

```
# cpower --cycle rli*n*
```

```
# cpower --identify 5 rli*n*
```

Note: `--up` is a synonym for `--on` and `--down` is a synonym for `--off`.

IRU, Rack, and System Domains

The `cpower` command contains more logic when you go up to higher levels of abstraction, for example, using `--iru`, `--rack`, and `--system`. These higher level domain specifiers tell the command to be smart about how to order various of the actions that you give on the command line.

The `--iru` option tells the command to use correct ordering with IRU power commands. In this case, it firsts connect to the CMC on each IRU in rack 1 to issue the `power on` command, which turns on power to the IRU chassis (this is not the equivalent `ipmitool` command). Then it powers up the compute nodes in the IRU. Powering things down is the opposite, with the power to the IRU being turned off after power to the blades. IRU targets are specified as follows: `r3i2` for rack 3, IRU 2.

```
# cpower --iru --up rli*
```

The `--rack` option ensures power commands to the leader node are down in the correct order relative to compute nodes within a rack. First, it powers up the leader

node and waits for it to boot up (if it is not already up). Then it will do the functional equivalent of a `cpower --irun --up r4i*` on each of the IRUs contained in the rack, including applying power to each IRU chassis. Using the `--down` option is the opposite, and also turns off the leader node (after doing a shutdown) after all the IRUs are powered down. To avoid including leader nodes in a power command for a rack, use the `--noleader` option. Rack targets are specified, as follows: `r4` for rack 4. Here is an example:

```
# cpower --rack --up r4
```

Commands with the `--system` option ensures that power up commands are applied first to service nodes, then to leader nodes, then to IRUs and compute blades, in just the same way. Likewise, compute blades are powered down before IRUs, leader nodes, and service nodes, in that order. To avoid including service nodes in a system-domain command, use the `--noservice` option. Note that you must not specify a target with `--system` option, since it applies to the SGI ICE system.

Shutting Down and Booting

Note: The `--shutdown --off` combination of actions were deprecated in a previous release. Use the `--halt` option in its place.

It is useful to be able to shutdown a machine before turning off the power, in most cases. The following `cpower` options to enable you to do this: `--halt`, `--boot`, and `--reboot`. The `--halt` option allows you to shut down a node. The `--reboot` option ensures that a system is always rebooted, whereas `--boot` will only boot up a system if it is not already booted. Thus, `--boot` is useful for booting up compute blades that have failed to start.

You need to configure the order in which service nodes are booted up and shut down as part of the overall system power management process. This is done by setting a `boot_order` for each service node. Use the `cadmin` command to set the boot order for a service node, for example:

```
# cadmin --set-boot-order --node service0 2
```

The `cpower --system --boot` command boots up service nodes with a lower boot order, first. It then boots up service nodes with a higher boot order. The reverse is true when shutting down the system with `cpower`. For example, if `service1` has a boot order of 3 and `service2` has a boot order of 5, `service1` is booted completely, and

then service2 is booted, afterwards. During shutdown, service2 is shut down completely before service1 is shutdown.

There is a special meaning to a service node having a boot order of zero. This value causes the `cpower --system` command to skip that service node completely for both start up and shutdown (although not for status queries). Negative values for the service node boot order setting are not permitted.

Note: The IPMI power commands necessary to enable a system to boot (either with a power reset, or a power on) may be sent to a node. The `--halt` option, halts the target node and then powers it off.

The `--halt` options works on node, IRU, or rack domain levels. It will shut down nodes (in the correct order if you use the `--iru` or `--rack` options), and then just leave them as they are, power still applied. Using both these actions results in nodes being halted, then powered off. This is particularly useful when powering off a rack, since otherwise, the leaders may be shutdown before there is a chance to power off the compute blades. Here is an example:

```
# cpower --halt --rack r1
```

To boot up systems that have not already been booted, perform the following:

```
# cpower --boot r1i2n*
```

Again, the command boots up nodes in the right orders if you specify the `--iru` or `--rack` options and the appropriate target. Otherwise, there is no guarantee that, for example, the command will attempt to power on the leader node before compute nodes in the same rack.

To reboot all of the nodes specified, or boot them if they are already shut down, perform the following:

```
# cpower --reboot --iru r3i3
```

The `--iru` or `--rack` options ensure proper ordering if you use them. In this case, the command will make sure that power is supplied to the chassis for rack 3, IRU 3, and then the all the compute nodes in that IRU will be rebooted.

EXAMPLES

Example 3-3 `cpower` Command Examples

To boot compute blade `r1i0n8`, perform the following:

```
# cpower --boot r1i0n8
```

To boot a number of compute blades at the same time, perform the following:

```
# cpower --boot --rack r1
```

Note: The `--boot` option will only boot those nodes that have not already booted.

To shut down service node 0, perform the following:

```
# cpower --halt service0
```

To shutdown and switch off everything in rack 3, perform the following:

```
# cpower --halt --rack r3
```

Note: This command will shutdown and then power off all of the computer nodes in parallel, then shutdown and power off the leader node. Use the `--noleader` option if you want the leader node to remain booted up.

To shutdown the entire system, including all service nodes and all leader nodes, but not the admin node, and not turn the power off to anything, perform the following:

```
# cpower --halt --system
```

To shutdown all the compute nodes, but not the service nodes, leader nodes, perform the following:

```
# cpower --halt --system --noleader --noservice
```

Note: The only way to shut down the system admin controller (admin node) is to perform the operation manually.

C3 Commands

Note: For legacy SGI ICE systems, this section remains intact. However, SGI recommends you use the `pdsh` and `pdcp` utilities described in "pdsh and pdcp Utilities" on page 231.

This section describes the cluster command and control (C3) tool suite for cluster administration and application support.

Note: The SMC for SGI ICE version of C3 does not include the `cshutdown` and `cpushimage` commands.

The C3 commands used on the the SGI Alitx ICE 8200 system are, as follows:

C3 Utilities	Description
<code>cexec(s)</code>	Executes a given command string on each node of a cluster
<code>cget</code>	Retrieves a specified file from each node of a cluster and places it into the specified target directory
<code>ckill</code>	Runs <code>kill</code> on each node of a cluster for a specified process name
<code>clist</code>	Lists the names and types of clusters in the cluster configuration file
<code>cnum</code>	Returns the node names specified by the range specified on the command line
<code>cname</code>	Returns the node positions specified by the node name given on the command line
<code>cpush</code>	Pushes files from the local machine to the nodes in your cluster

`cexec` is the most useful C3 utility. Use the `cpower` command rather than `cshutdown` (see "Power Management Commands" on page 220).

EXAMPLES

Example 3-4 C3 Command General Examples

The following examples walk you through some typical C3 command operations.

You can use the `cname` and `cnum` commands to map names to locations and vice versa, as follows:

```
# cname rack_1:0-2
local name for cluster: rack_1
nodes from cluster: rack_1
cluster: rack_1 ; node name: r1i0n0
cluster: rack_1 ; node name: r1i0n1
cluster: rack_1 ; node name: r1i0n10
```

```
# cnum rack_1: r1i0n0
local name for cluster: rack_1
nodes from cluster: rack_1
r1i0n0 is at index 0 in cluster rack_1
```

```
# cnum rack_1: r1i0n1
local name for cluster: rack_1
nodes from cluster: rack_1
```

You can use the `clist` command to retrieve the number of racks, as follows:

```
# clist
cluster rack_1 is an indirect remote cluster
cluster rack_2 is an indirect remote cluster
cluster rack_3 is an indirect remote cluster
cluster rack_4 is an indirect remote cluster
```

You can use the `cexec` command to view the addressing scheme of the C3 utility, as follows:

```
# cexec rack_1:1 hostname
***** rack_1 *****
***** rack_1 *****
----- r1i0n1-----
r1i0n1

# cexec rack_1:2-3 rack_4:0-3,10 hostname
***** rack_1 *****
***** rack_1 *****
```

```

----- r1i0n10-----
r1i0n10
----- r1i0n11-----
r1i0n11
***** rack_4 *****
***** rack_4 *****
----- r4i0n0-----
r4i0n0
----- r4i0n1-----
r4i0n1
----- r4i0n10-----
r4i0n10
----- r4i0n11-----
r4i0n11
----- r4i0n4-----
r4i0n4

```

The following set of command shows how to use the C3 commands to transverse the different levels of hierarchy in your SGI ICE system (for information on the hierarchical design of your SGI ICE system see "Basic System Building Blocks" on page 1).

To execute a C3 command on all blades within the default SGI ICE system, for example, rack 1, perform the following:

```

# cexec hostname
***** rack_1 *****
***** rack_1 *****
----- r1i0n0-----
r1i0n0
----- r1i0n1-----
r1i0n1
----- r1i0n10-----
r1i0n10
----- r1i0n11-----
r1i0n11
...

```

To run a C3 command on all compute nodes across an SGI ICE system, perform the following:

```
# cexec --all hostname
***** rack_1 *****
***** rack_1 *****
----- r1i0n0-----
r1i0n0
----- r1i0n1-----
r1i0n1
...
----- r2i0n10-----
r2i0n10
...
----- r3i0n11-----
r3i0n11
...
```

To run a C3 command against the first rack leader controller, in the first rack, perform the following:

```
# cexec --head hostname
***** rack_1 *****
----- rack_1-----
r1lead
```

To run a C3 command against all rack leader controllers across all racks, perform the following:

```
# cexec --head --all hostname
***** rack_1 *****
----- rack_1-----
r1lead
***** rack_2 *****
----- rack_2-----
r2lead
***** rack_3 *****
----- rack_3-----
r3lead
***** rack_4 *****
----- rack_4-----
```

```
r4lead
```

The following set of examples shows some specific case uses for the C3 commands that you are likely to employ.

Example 3-5 C3 Command Specific Use Examples

From the **system admin controller**, run command on rack 1 without including the rack leader controller, as follows:

```
# cexec rack_1: <cmd>
```

Run a command on all service nodes only, as follows:

```
# cexec -f /etc/c3svc.conf <cmd>
```

Run a command on all compute nodes in the system, as follows:

```
# cexec --all <cmd>
```

Run a command on all rack leader controllers, as follows:

```
# cexec --all --head <cmd>
```

Run a command on blade 42 (compute node 42) in rack 2, as follows:

```
# cexec rack_2:42 <cmd>
```

From a **service node** over the InfiniBand Fabric, run a command on all blades (compute nodes) in the system, as follows:

```
# cexec --all <cmd>
```

Run a command on blade 42 (compute node 42), as follows:

```
# cexec blades:42 <cmd>
```

pdsh and pdcp Utilities

The `pdsh(1)` command is the parallel shell utility. The `pdcp(1)` command is the parallel copy/fetch utility. The SMC for SGI ICE software populates some `dshgroups` files for the various node types. On the admin node, SMC for SGI ICE software populates the `leader` and `service` groups files, which contain the list of online nodes in each of those groups.

On the leader node, software populates the `compute` group for all the online compute nodes in that group.

On the service node, software populates the `compute` group which contains all the online compute nodes in the whole system.

For more information, see the `pdsh(1)` and `pdcp(1)` man pages.

EXAMPLES

From the admin node, to run the `hostname` command on all the leader nodes, perform the following:

```
# pdsh -g leader hostname
```

To run the `hostname` command on all the compute nodes in the system, via the leader nodes, perform the following:

```
# pdsh -g leader pdsh -g compute hostname
```

To run the `hostname` command on just `r1lead` and `r2lead`, perform the following:

```
# pdsh -w r1lead,r2lead hostname
```

cadmin: SMC for SGI ICE Administrative Interface

The `cadmin` command allows you to change certain administrative parameters in the cluster such as the boot order of service nodes, the administrative status of nodes, and the adding, changing, and removal of IP addresses associated with service nodes.

To get the `cadmin` usage statement, perform the following:

```
admin ~]# cadmin --h
cadmin: SGI Tempo Administrative Interface
Help:
```

```
In general, these commands operate on {node}. {node} is the Tempo style
node name. For example, service0, r1lead, r1i0n0. Even when the host name
for a service node is changed, the Tempo name for that node may still be used
for {node} below. The node name can either be the tempo unique node name
or a customer-supplied host name associated with a tempo unique node name.
```

```
--version : Display current release information
```

```

--set-admin-status --node {node} {value} : Set Administrative Status
--show-admin-status --node {node} : Show Administrative Status
--set-boot-order --node {node} [value] : Set boot order [1]
--show-boot-order --node {node} : Show boot order [1]
--set-ip --node {node} --net {net} {hostname}={ip} : Change an allocated ip [1]
--del-ip --node {node} --net {net} {hostname}={ip} : Delete an ip [1]
--add-ip --node {node} --net {net} {hostname}={ip} : allocate a new ip [1]
--show-ips --node {node} : Show all allocated IPs associated with node
--set-hostname --node {node} {new-hostname} : change the host name [5]
--show-hostname --node {node} : show the current host name for ice node {node}
--set-subdomain {domain} : Set the cluster subdomain [3]
--show-subdomain : Show the cluster subdomain
--set-admin-domain {domain} : Set the admin node house network domain
--show-admin-domain : Show the admin node house network domain
--db-purge --node {node} : Purge service or lead node (incl entire rack) from DB
--set-external-dns --ip {ip} : Set IP addr(s) of external DNS master(s) [4]
--show-external-dns : Show the IP addr(s) of the external DNS master(s)
--del-external-dns : Delete the configuration of external DNS master(s)
--show-root-labels : Show grub root labels if multiple roots are in use
--set-root-label --slot {#} --label {label} : Set changeable part of root label
--show-default-root : Show default root if multiple roots are in use
--set-default-root --slot {#} : Set the default slot if multiple roots in use
--show-current-root : Show current root slot
--enable-auto-recovery : Enable ability for nodes to recover themselves [6]
--disable-auto-recovery : Disable auto recovery [6]
--show-auto-recovery : Show the current state of node auto recovery [6]
--enable-redundant-mgmt-network --node {node}: Enable network
management redundancy
--disable-redundant-mgmt-network --node {node}: Disable management network
redundancy
--show-redundant-mgmt-network --node {node}: Show current value.
--show-dhcp-option: Show admin dhcp option code used to distinguish mgmt network
--set-dhcp-option {value}: Set admin dhcp option code
--enable-switch-mgmt-network --node {node}: Enable switch management network
for a node that is connected to managed top level switches. Not for ICE
8200/8400 nodes. [7]
--disable-switch-mgmt-network --node {node}: Disable switch management network
for the specified node.
--show-switch-mgmt-network --node {node}: Show current value.
--enable-replication --node {node}: Enable MySQL Replication on the specified
admin, leader, or service node. [8]

```

3: System Operation

```
--disable-replication --node {node}: Disable MySQL Replication on the specified
admin, leader, or service node.
--show-replication-status --node {node}: Show current value.
--set-mgmt-bonding --node {node} {value}: "802.3ad" or "active-backup"
    Must be "active-backup" ICE 8200/8400 systems. Depends on mgmt switch.
--show-mgmt-bonding --node {node}: Show current value.
--set-max-rack-irus --node {node} {value}: Max number of IRUs for this node.
    Saves push-rack time by not doing expansions for nodes that do not exist.
--show-max-rack-irus --node {node}: Show current value
```

Node-attribute options:

```
--add-attribute [--string-data "{string}"] [--int-data {int}] {attribute-name}
--is-attribute {attribute-name}
--delete-attribute {attribute-name}
--set-attribute-data [--string-data "{string}"] [--int-data {int}]
    {attribute-name}
--get-attribute-data {attribute-name}
--search-attributes [--string-data "{string|regex}"] [--int-data {int}]
--add-node-attribute [--string-data "{string}"] [--int-data {int}]
    --node {node} --attribute {attribute-name}
--is-node-attribute --node {node} --attribute {attribute-name}
--delete-node-attribute --node {node} --attribute {attribute-name}
--set-node-attribute-data [--string-data "{string}"] [--int-data {int}]
    --node {node} --attribute {attribute-name}
--get-node-attribute-data --node {node} --attribute {attribute-name}
--search-node-attributes [--node {node}] [--attribute {attribute-name}]
    [--string-data "{string|regex}"] [--int-data {int}]
```

Descriptions of Selected Values:

{hostname}={ip} means specify the host name associated with the specified ip address.

{net} is the tempo network to change such as ib-0, ib-1, head, gbe, bmc, etc

{node} is a tempo-style node name such as rlead, service0, or rli0n0.

[1] Only applies to service/ibswitch/mgmtswitch nodes

[2] This operation may require the cluster to be fully shut down and AC power to be removed. IPs will have to be re-allocated to fit in the new range.

[3] All cluster nodes will have to be reset. Compute images need to be pushed again with the cimage command.

[4] Use quoted, semi-colon separated list if more than one master

[5] Only applies to admin and service/ibswitch/mgmtswitch nodes

[6] Auto recovery will allow service and leader nodes to boot in to a special

recovery mode if the cluster doesn't recognize them. This is enabled by default and would be used, for example, if a node's main board was replaced but the original system disks were imported from the original system.

- [7] The global value for this is automatically detected and adjustable using `configure-cluster`. This per-node value is used for systems that mix the older cascaded CMC management network found in ICE 8200/8400 with the newer top level switch management network. In a mixed system, all leader and managed service nodes in the 8200/8400 part of the system should have this set to `no`. This can also be specified using the `"discover"` command.
- [8] Node should be `admin`, `leader`, or a `service` node. Use `configure-cluster` to globally configure MySQL Replication on the whole system. If MySQL Replication is disabled on the `admin` node, replication will be off for the whole system, even if some nodes may have the replication attribute set to `"yes"`. Use this feature with caution, if replication is enabled on a `leader` or `service` node, but disabled on the `admin` node, the `leader` or `service` node will not be able to get the information from the `admin` node, and the database calls will fail or return out of date information. To enable MySQL Replication on a limited number of node, disable it globally from within `configure-cluster`, then use `cadmin` to enable it on the `admin` node, and then on the other nodes. Replication is disabled by default.

EXAMPLES

Example 3-6 SMC for SGI ICE Administrative Interface (`cadmin`) Command

Set a node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

Set a node online, as follows:

```
# cadmin --set-admin-status --node r1i0n0 online
```

Set the boot order for a service node, as follows:

```
# cadmin --set-boot-order --node service0 2
```

Add an IP to an existing service node, as follows:

```
# cadmin --add-ip --node service0 --net ib-0 my-new-ib0-ip=10.148.0.200
```

Change the SMC for SGI ICE needed `service0-ib0` IP address, as follows:

```
# cadmin --set-ip --node service0 --net head service0=172.23.0.199
```

Show currently allocated IP addresses for service0, as follows:

```
# cadmin --show-ips --node service0  
IP Address Information for SMC for SGI ICE node: service0
```

ifname	ip	Network
myservice-bmc	172.24.0.3	head-bmc
myservice	172.23.0.3	head
myservice-ib0	10.148.0.254	ib-0
myservice-ib1	10.149.0.67	ib-1
myhost	172.24.0.55	head-bmc
myhost2	172.24.0.56	head-bmc
myhost3	172.24.0.57	head-bmc

Delete a site-added IP address (you cannot delete SMC for SGI ICE needed IP addresses), as follows:

```
admin:~ # cadmin --del-ip --node service0 --net ib-0 my-new-ib0-2-ip=10.148.0.201
```

Change the hostname associated with service0 to be myservice, as follows:

```
admin:~ # cadmin --set-hostname --node service0 myservice
```

Change the hostname associated with admin to be newname, as follows:

```
admin:~ # cadmin --set-hostname --node admin newname
```

Set and show the cluster subdomain, as follows:

```
admin:~ # cadmin --set-subdomain mysubdomain.domain.mycompany.com  
admin:~ # cadmin --show-subdomain  
The cluster subdomain is: mysubdomain
```

Show the admin node house network domain, as follows:

```
admin:~ # cadmin --show-admin-domain  
The admin node house network domain is: domain.mycompany.com
```

Show the SMC for SGI ICE systems DHCP option identifier, as follows:

```
admin:~ # cadmin --show-dhcp-option  
149
```

Show the current switch management value for a specified node, as follows:

```
admin:~ # cadmin --show-switch-mgmt-network --node admin  
no
```

Enable the switch management network for a specified node that is connected to managed top level switches, as follows:

```
admin:~ # cadmin --enable-switch-mgmt-network --node admin
```

Note: This option cannot be used for SGI ICE 8200 or SGI ICE 8400 system nodes.

Disable the switch management network for a specified node that is connected to managed top level switches, as follows:

```
admin:~ # cadmin --disable-switch-mgmt-network --node admin
```

Note: This option cannot be used for SGI ICE 8200 or SGI ICE 8400 system nodes.

Show MySQL replication status for a specified admin, leader, or service node, as follows:

```
admin:~ # cadmin --show-replication --node r2lead  
yes
```

Note: MySQL replication is disabled by default.

Enable MySQL replication on a specified admin, leader, or service node, as follows:

```
admin:~ # cadmin --enable-replication --node r2lead  
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...  
mysql          0:off 1:off 2:on 3:on 4:off 5:on 6:off  
Restarting service MySQL  
Shutting down service MySQL ..done  
Starting service MySQL ..done
```

Disable MySQL replication on a specified admin, leader, or service node, as follows:

```
admin:~ # cadmin --disable-replication --node r2lead  
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...  
Shutting down service MySQL ..done
```

```
mysql                                0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Console Management

SMC for SGI ICE management systems software uses the open-source console management package called `conserver`. For detailed information on `conserver`, see <http://www.conserver.com/>

An overview of the `conserver` package is, as follows:

- Manages the console devices of all managed nodes in an SGI ICE system
- A `conserver` daemon runs on the system admin controller (admin node) and the rack leader controllers (leader nodes). The system admin controller manages leader and service node consoles. The rack leader controllers manage blade consoles.
- The `conserver` daemon connects to the consoles using `ipmitool`. Users connect to the daemon to access them. Multiple users can connect but non-primary users are read-only.
- The `conserver` package is configured to allow all consoles to be accessed from the system admin controller.
- All consoles are logged. These logs can be found at `/var/log/consoles` on the system admin controller and rack leader controllers. An `autofs` configuration file is created to allow you to access rack leader controller managed console logs from the system admin controller, as follows:

```
admin # cd /net/r1lead/var/log/consoles/
```

The `/etc/conserver.cf` file is the configuration file for the `conserver` daemon. This file is generated for both the system admin controller and rack leader controllers from the `/opt/sgi/sbin/generate-conserver-files` script on the system admin controller. This script is called from `discover-rack` command as part of rack discovery or rediscovery and generates both the `conserver.cf` file for the rack in question and regenerates the `conserver.cf` for the system admin controller.

Note: The `conserver` package replaces `cconsole` for access to all consoles (blades, leader nodes, managed service nodes)

You may find the following `conserver` man pages useful:

Man Page	Description
console(1)	Console server client program
conserver(8)	Console server daemon
conserver.cf(5)	Console configuration file for <code>conserver(8)</code>
conserver.passwd(5)	User access information for <code>conserver(8)</code>

Procedure 3-7 Using `conserver` Console Manager

To use the `conserver` console manager, perform the following steps:

1. To see the list of available consoles, perform the following:

```
admin:~ # console -x
service0          on /dev/pts/2          at Local
r2lead            on /dev/pts/1          at Local
r1lead            on /dev/pts/0          at Local
r1i0n8            on /dev/pts/0          at Local
r1i0n0            on /dev/pts/1          at Local
```

2. To connect to the service console, perform the following:

```
admin:~ # console service0
[Enter '^Ec?' for help]
```

```
Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64) - Kernel 2.6.16.60-0.12-smp (ttyS1).
```

```
service0 login:
```

3. To connect to the rack leader controller console, perform the following:

```
admin:~ # console r1lead
[Enter '^Ec?' for help]
```

```
Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64)
- Kernel 2.6.16.60-0.12-smp (ttyS1).
```

rllead login:

4. To trigger system request commands `sysrq` (once connected to a console), perform the following:

```
Ctrl-e c l 1 8           # set log level to 8
Ctrl-e c l 1 <sysrq cmd> # send sysrq command
```

5. To see the list of `conserver` escape keys, perform the following:

```
Ctrl-e c ?
```

Keeping System Time Synchronized

The SMC for SGI ICE systems management software uses network time protocol (NTP) as the primary mechanism to keep the nodes in your SGI ICE system synchronized. This section describes this mechanism operates on the various SGI ICE components and covers these topics:

- "System Admin Controller NTP" on page 240
- "Rack Leader Controller NTP" on page 241
- "Managed Service, Compute, and Leader BMC Setup with NTP" on page 241
- "Service Node NTP" on page 241
- "Compute Node NTP" on page 241
- "NTP Work Arounds" on page 241

System Admin Controller NTP

When you used the `configure-cluster` command, it guided you through setting up NTP on the admin node. The NTP client on the system admin controller should point to the house network time server. The NTP server provides NTP service to system components so that nodes can consult it when they are booted. The system admin controller sends NTP broadcasts to some networks to keep the nodes in sync after they have booted.

Rack Leader Controller NTP

NTP client on the rack leader controller gets time from the system admin controller when it is booted and then stays in sync by connecting to the admin node for time. The NTP server on the leader node provides NTP service to SGI ICE components so that compute nodes can sync their time when they are booted. The rack leader controller sends NTP broadcasts to some networks to keep the compute nodes in sync after they have booted.

Managed Service, Compute, and Leader BMC Setup with NTP

The BMC controllers on managed service nodes, compute nodes, and leader nodes are also kept in sync with NTP. Note that you may need the latest BMC firmware for the BMCs to sync with NTP properly. The NTP server information for BMCs is provided by special options stored in the DHCP server configuration file.

Service Node NTP

The NTP client on *managed* service nodes (for a definition of managed, see "discover Command" on page 80) sets its time at initial booting from the system admin controller. It listens to NTP broadcasts from the system admin controller to stay in sync. It does not provide any NTP service.

Compute Node NTP

The NTP Client on the compute node sets its time at initial booting from the rack leader controller. It listens to NTP broadcasts from the rack leader controller to stay in sync.

NTP Work Arounds

Sometime, especially during initial deployment of an SGI ICE system when system components are being installed and configured for the first time, NTP is not available to serve time to system components.

A non-modified NTP server, running for the first time, takes quite some time before it offers service. This means the leader and service nodes may fail to get time from the system admin controller as they come on-line. Compute nodes may also fail to get time from the leader when they first come up. This situation usually only happens at

first deployment. After the `ntp` servers have a chance to create their drift files, `ntp` servers offer time with far less delay on subsequent reboots.

The following work arounds are in place for situations when NTP can not serve the time:

- The admin and rack leader controllers have the `time` service enabled (`xinetd`).
- All system node types have the `netdate` command.
- A special startup script is on leader, service, and compute nodes that runs before the NTP startup script.

This script attempts to get the time using the `ntpdate` command. If the `ntpdate` command fails because the NTP server it is using is not ready yet to offer time service, it uses the `netdate` command to get the clock "close".

The `ntp` startup script starts the NTP service as normal. Since the clock is known to be "close", NTP will fix the time when the NTP servers start offering time service.

Changing the Size of `/tmp` on Compute Nodes

This section describes how to change the size of `/tmp` on SGI ICE compute nodes.

Procedure 3-8 Increasing the `/tmp` Size

To change the size of `/tmp` on your system compute nodes, perform the following steps:

1. From the admin node, change directory (`cd`) to `/opt/sgi/share/per-host-customization/global`.
2. Open the `sgi-fstab` file and change the `size=` parameter for the `/tmp` mount in both locations that it appears.

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
```

```

# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
#
# Set up the compute node's /etc/fstab file.
#
# Modify per your sites requirements.
#
# This script is executed once per-host as part of the install-image operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

# sanity checks
. /opt/sgi/share/per-host-customization/global/sanity.sh

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n '/^compat /s/^compat.*: //p' ) )

compatdistro=${os[0]}${os[1]}

if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then

    #
    # SLES 10 compatible
    #
    cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
tmpfs /tmp tmpfs size=150m 0 0
EOF

elif [ ${compatdistro} = "rhel5" ]; then

```

```
#
# RHEL 5 compatible
#
#
# RHEL expects several subsys directories to be present under /var/run
# and /var/lock, hence no tmpfs mounts for them
#
cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
tmpfs          /tmp          tmpfs        size=150m    0           0
devpts         /dev/pts      devpts       gid=5,mode=620 0           0
EOF

else

    echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi
```

3. Push the image out to the racks to pick up the change, as follows:

```
# cimage --push-rack mynewimage r\*
```

For more information on using the `cimage` command, see "[cimage Command](#)" on page 206.

Enabling or Disabling the Compute Node iSCSI Swap Device

This section describes how to enable or disable the Internet small computer system interface (iSCSI) compute node swap device. The iSCSI compute node swap device is turned off by default for new installations. It can cause problems during rack-wide out of memory (OOM) conditions, with both compute nodes and the rack leader controller (RLC) becoming unresponsive during the heavy write-out to the per-node iSCSI swap devices.

Procedure 3-9 Enabling the iSCSI Swap Device

If you wish to enable the iSCSI swap device in a given compute node image, perform the following steps:

1. Change root (chroot) into the compute node image on the admin node and enable the `iscsiswap` service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap on
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Procedure 3-10 Disabling the iSCSI Swap Device

To disable the iSCSI swap device in a compute node image where it is currently enabled, perform the following steps:

1. Disable the service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap off
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Changing the Size of Per-node Swap Space

This section describes how to change per-node swap space on your SGI ICE system.

Procedure 3-11 Increasing Per-node Swap Space

To increase the default size of the per-blade swap space on your system, perform the following:

1. Shutdown all blades in the affected rack (see "Shutting Down and Booting" on page 224).
2. Log into the leader node for the rack in question. (Note that you need to do this on each rack leader).
3. Change directory (cd) to the `/var/lib/sgi/swapfiles` directory.

- To adjust the swap space size appropriate for your site, run a script similar to the following:

```
#!/bin/bash

size=262144    # size in KB

for i in $(seq 0 3); do
    for n in $(seq 0 15); do
        dd if=/dev/zero of=i${i}n${n} bs=1k count=${size}
        mkswap i${i}n${n}
    done
done
```

- Reboot the all blades in the affected rack (see "Shutting Down and Booting" on page 224).
- From the rack leader node, use the `cexec --all` command to run the `free(1)` command on the compute blades to view the new swap sizes, as follows:

```
rllead:~ # cexec --all free
***** rack_1 *****
----- rli0n0-----
      total      used      free      shared    buffers    cached
Mem:      2060140  206768  1853372         0         4      46256
-/+ buffers/cache:  160508  1899632
Swap:      49144         0      49144
----- rli0n1-----
      total      used      free      shared    buffers    cached
Mem:      2060140  137848  1922292         0         4      44200
-/+ buffers/cache:   93644  1966496
Swap:      49144         0      49144
----- rli0n8-----
      total      used      free      shared    buffers    cached
Mem:      2060140  138076  1922064         0         4      43172
-/+ buffers/cache:   94900  1965240
Swap:      49144         0      49144
```

If you want change per-node swap space across your entire system, all (new) leaders nodes as part of discovery, you can edit the `/etc/opt/sgi/conf.d/35-compute-swapfiles` "inside" the `lead-sles11` image on the admin node. The images are in the

`/var/lib/systemimager/images` directory. For more information on customizing these images, see "Customizing Software Images" on page 203.

Switching Compute Nodes to a `tmpfs` Root

This section describes how to switch your system compute nodes to a `tmpfs` root.

Procedure 3-12 Switching Compute Nodes to a `tmpfs` Root

To switch your compute nodes to a `tmpfs` root, from the system admin controller (admin node) perform the following steps:

1. To switch compute nodes to a `tmpfs` root, use the optional `--tmpfs` flag to the `cimage --set` command, for example:

```
adminadmin:~ # cimage --set --tmpfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

Note: To use a `/tmpfs` root with the standard compute node image, the compute node needs to have 4GB of memory or above. A standard `/tmpfs` mount has access to half the system memory, and the standard compute node image is just over 1 GB in size.

2. You can view the current setting of a compute node, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp tmpfs
```

3. To set it back to an NFS root, use the `--nfs` flag to the `cimage --set` command, as follows:

```
admin:~ # cimage --set --nfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

4. You can change the view the change back to NFS root, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp nfs
```

For help information, use the `cimage --h` option.

Setting up Local Storage Space for Swap and Scratch Disk Space

The SGI ICE system has the option to support local storage space on compute nodes (also known as blades). Solid-state drive (SSD) devices and 2.5" disks are available for this purpose. You can define the size and status for both swap and scratch partitions. The values can be set globally or per node or group of nodes. By default, the disks are partitioned only if blank, the swap is off, and the scratch is set to occupy the whole disk space and be mounted in `/tmp/scratch`.

The `/etc/init.d/set-swap-scratch` script is responsible for auto-configuring the swap and scratch space based on the settings retrieved via the `cattr` command. You can use the `cadmin` to configure settings globally or you can use the `cattr` command to set custom values for specific nodes.

The `/etc/opt/sgi/conf.d/30-set-swap-scratch` script makes sure `/etc/init.d/swapscratch` service is on so that swap/scratch partitions are configured directly after booting. The `swapscratch` service calls the `/opt/sgi/lib/set-swap-scratch` script when the service is started and then it exits.

You can customize the following settings:

- `blade_disk_allow_partitioning`

The default value is "on" which means that the `set-swap-scratch` script will repartition and format the local storage disk if needed.

Note: To protect user data, the script will not re-partition the disk if it is already partitioned. In this case, you need a blank disk before it can be used for `swap/scratch`.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_allow_partitioning` value for the node on which it is running:

```
# cattr get blade_disk_allow_partitioning -N $compute_node_name --default on
```

You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_allow_partitioning
```

You can globally turn it off, as follows:

```
# cadmin --add-attribute --string-data off blade_disk_allow_partitioning
```

- blade_disk_swap_status

The default value is "off" which means that the set-swap-scratch script will not enable a swap partition on the local storage disk.

The set-swap-scratch script uses the following command to retrieve the blade_disk_swap_status value for the node on which it is running:

```
# cattr get blade_disk_swap_status -N $compute_node_name --default off
```

You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_swap_status
```

You can globally turn it off, as follows:

```
# cadmin --add-attribute --string-data off blade_disk_swap_status
```

The set-swap-scratch script uses SGI_SWAP label when partitioning the disk. It enables the swap only if it finds a partition labeled SGI_SWAP.

- blade_disk_swap_size

The default value is 0 which means that the set-swap-scratch script will not create a swap partition on the local storage disk.

The set-swap-scratch script uses the following command to retrieve the blade_disk_swap_size value for the node on which it is running:

```
attr get blade_disk_swap_size -N $compute_node_name --default 0
```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data 1024 blade_disk_swap_size
```

The size is specified in megabytes. Allowed values are, as follows: 0, -0 (use all free space when partitioning), 1, 2, ...

- blade_disk_scratch_status

The default value is "off" which means that the set-swap-scratch script will not enable the scratch partition on the local storage disk.

The set-swap-scratch script uses the following command to retrieve the blade_disk_scratch_status value for the node on which it is running:

```
cattr get blade_disk_scratch_status -N $compute_node_name --default off
```

You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_scratch_status
```

You can globally turn it off, as follows:

```
cadmin --add-attribute --string-data off blade_disk_scratch_status
```

Note: The `set-swap-scratch` script uses the `SGI_SCRATCH` label when partitioning the disk. It mounts the scratch only on the partition labeled as `SGI_SCRATCH`.

- `blade_disk_scratch_size`

The default value is `-0` which means that the `set-swap-scratch` script will use all remaining free space when creating the scratch partition.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

```
catr get blade_disk_scratch_size -N $compute_node_name --default -0
```

You can globally set the value, as follows:

```
cadmin --add-attribute --string-data 10240 blade_disk_scratch_size
```

The size is specified in megabytes. Allowed values are, as follows: `0`, `-0` (use all free space when partitioning), `1`, `2`, ...

- `blade_disk_scratch_mount_point`

The default value is `/tmp/scratch` which means that the `set-swap-scratch` script will mount the scratch partition in `/tmp/scratch`.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

```
# catr get blade_disk_scratch_mount_point -N $compute_node_name --default /tmp/scratch
```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data /tmp/scratch blade_disk_scratch_mount_point
```

You can mount the disk to any mount point you desire. The `set-swap-scratch` script will create that folder if it does not exist (as long as the script has the permission to create it at that path). The root mount point (`/`) is not writable on the compute nodes. You need to create that folder as part of the compute node image if you want to mount something like `/scratch`.

- `blade_disk_raid_level`

The default value is `off`. When set to 0, it allows you to setup RAID0, if you have two disks for `swap/scratch`. Values of 1, . . . are currently ignored.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_raid_level` value for the node on which it is running:

```
attr get blade_disk_raid_level -N $compute_node_name --default 0
```

You can globally set the value, as follows:

```
# cadadmin --add-attribute --string-data blade_disk_raid_level
```

- `blade_disk_reformat_swap_at_boot`

The default value is `off`. When set to 0, it allows you to format the swap every time the node reboots. Values of 1, . . . are currently ignored.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_swap_at_boot` value for the node on which it is running:

```
attr get blade_disk_reformat_swap_at_boot -N $compute_node_name --default 0
```

You can globally set the value, as follows:

```
# cadadmin --add-attribute --string-data blade_disk_reformat_swap_at_boot
```

- `blade_disk_reformat_scratch_at_boot`

The default value is `off`. When set to 0, it allows you to format the swap every time the node reboots. Values of 1, . . . are currently ignored.

The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_scratch_at_boot` value for the node on which it is running:

```
attr get blade_disk_reformat_scratch_at_boot -N $compute_node_name --default 0
```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data blade_disk_reformat_scratch_at_boot
```

For a `cattr` command help statement, perform the following command:

```
# cattr -h
Usage:
  cattr [--help] COMMAND [ARG]...

Commands:
  exists  check for the existence of an attribute
  get     print the value of an attribute
  list    print a list of attribute values
  set     set the value of an attribute
  unset   delete the value of an attribute
```

For more detailed help, use `'cattr COMMAND --help'`.

Viewing the Compute Node Read-Write Quotas

This section describes how to view the per compute node read and write quota.

Procedure 3-13 Viewing the Compute Node Read-Write Quotas

To view the per compute node read and write quota, log onto the leader node and perform the following:

```
rllead:~ # xfs_quota -x -c 'quota -ph 1'
Disk quotas for Project #1 (1)
Filesystem  Blocks  Quota  Limit Warn/Time  Mounted on
/dev/disk/by-label/sgiroot
           64.6M    0    1G  00 [-----] /
```

Map the XFS project ID to the quota you are interested in by looking it up in `/etc/projects` file.

If you decided to change the `xfs_quota` values, log back onto the admin node and edit the `/etc/opt/sgi/cminfo` file **inside** the compute image where you want to change the value, for example, `/var/lib/systemimager/images/image_name`.

Change the value of the `PER_BLADE_QUOTA` variable and then repush the image with the following command:

```
# cimage --push-rack image_name racks
```

For help information, perform the following:

```
xfs_quota> help
df [-bir] [-hn] [-f file] -- show free and used counts for blocks and inodes
help [command] -- help for one or all commands
print -- list known mount points and projects
quit -- exit the program
quota [-bir] [-gpu] [-hmv] [-f file] [id|name]... -- show usage and limits
```

Use 'help commandname' for extended help

Use help *commandname* for extended help, such as the following:

```
xfs_quota> help quota
```

```
quota [-bir] [-gpu] [-hmv] [-f file] [id|name]... -- show usage and limits
```

```
display usage and quota information
```

```
-g -- display group quota information
-p -- display project quota information
-u -- display user quota information
-b -- display number of blocks used
-i -- display number of inodes used
-r -- display number of realtime blocks used
-h -- report in a human-readable format
-n -- skip identifier-to-name translations, just report IDs
-N -- suppress the initial header
-v -- increase verbosity in reporting (also dumps zero values)
-f -- send output to a file
```

The (optional) user/group/project can be specified either by name or by number (i.e. uid/gid/projid).

```
xfs_quota>
```

RAID Utility

The infrastructure nodes on your SGI ICE system have LSI RAID enabled by default from the factory. A `lsiutil` command-line utility is included with the installation for the admin node, the leader node, and the service node (when installed from the SGI service node image). This tool allows you to look at the devices connected to the RAID controller and manage them. Some functions, such as, setting up mirrored or striped volumes, can be handled either by the LSI BIOS configuration tool or the `lsiutil` utility.

Note: These instructions only apply to SGI XE250 or SGI XE270 systems with the 1068-based controller. They do not apply to SGI XE250 or SGI XE270 systems that have the LSI Megaraid controller.

Example 3-7 Using the `lsiutil` Utility

The following `lsiutil` command-line utility example shows a sample session, as follows:

Start the `lsiutil` tool, as follows:

```
admin:~ # lsiutil

LSI Logic MPT Configuration Utility, Version 1.54, January 22, 2008

1 MPT Port found

      Port Name          Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
1.  /proc/mpt/ioc0      LSI Logic SAS1068E B2    105      01140100      0

Select a device: [1-1 or 0 to quit]

Select 1 to show the MPT Port, as follows:

1 MPT Port found

      Port Name          Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
1.  /proc/mpt/ioc0      LSI Logic SAS1068E B2    105      01140100      0

Select a device: [1-1 or 0 to quit] 1

1.  Identify firmware, BIOS, and/or FCode
```

- 2. Download firmware (update the FLASH)
- 4. Download/erase BIOS and/or FCode (update the FLASH)
- 8. Scan for devices
- 10. Change IOC settings (interrupt coalescing)
- 13. Change SAS IO Unit settings
- 16. Display attached devices
- 20. Diagnostics
- 21. RAID actions
- 22. Reset bus
- 23. Reset target
- 42. Display operating system names for devices
- 45. Concatenate SAS firmware and NVDATA files
- 60. Show non-default settings
- 61. Restore default settings
- 69. Show board manufacturing information
- 97. Reset SAS link, HARD RESET
- 98. Reset SAS link
- 99. Reset port
 - e Enable expert mode in menus
 - p Enable paged mode in menus
 - w Enable logging

Main menu, select an option: [1-99 or e/p/w or 0 to quit]

Choose 21. RAID actions, as follows:

Main menu, select an option: [1-99 or e/p/w or 0 to quit] **21**

- 1. Show volumes
- 2. Show physical disks
- 3. Get volume state
- 4. Wait for volume resync to complete
- 23. Replace physical disk
- 26. Disable drive firmware update mode
- 27. Enable drive firmware update mode
- 30. Create volume
- 31. Delete volume
- 32. Change volume settings
- 50. Create hot spare
- 99. Reset port
 - e Enable expert mode in menus
 - p Enable paged mode in menus

w Enable logging

RAID actions menu, select an option: [1-99 or e/p/w or 0 to quit]

Choose 2. Show physical disks, to show the status of the disks making up the volume, as follows:

RAID actions menu, select an option: [1-99 or e/p/w or 0 to quit] **2**

1 volume is active, 2 physical disks are active

PhysDisk 0 is Bus 0 Target 1

PhysDisk State: online

PhysDisk Size 238475 MB, Inquiry Data: ATA Hitachi HDT72502 A73A

PhysDisk 1 is Bus 0 Target 2

PhysDisk State: online

PhysDisk Size 238475 MB, Inquiry Data: ATA Hitachi HDT72502 A73A

RAID actions menu, select an option: [1-99 or e/p/w or 0 to quit]

Choose 1. Show volumes, to show information about the volume including its health, as follows:

RAID actions menu, select an option: [1-99 or e/p/w or 0 to quit] **1**

1 volume is active, 2 physical disks are active

Volume 0 is Bus 0 Target 0, Type IM (Integrated Mirroring)

Volume Name:

Volume WWID: 09195c6d31688623

Volume State: optimal, enabled

Volume Settings: write caching disabled, auto configure

Volume draws from Hot Spare Pools: 0

Volume Size 237464 MB, 2 Members

Primary is PhysDisk 1 (Bus 0 Target 2)

Secondary is PhysDisk 0 (Bus 0 Target 1)

RAID actions menu, select an option: [1-99 or e/p/w or 0 to quit]

Restoring the grub Boot-loader on a Node

When `grub(8)` boot-loader is not written to the rack leader controllers (leader nodes) or any of the system service nodes or is not functioning correctly, the `grub` boot-loader will have to be re-installed on the master boot record (MBR) of the root drive for the node.

To rewrite `grub` to the MBR of the root drive on a system that is booted, issue the following `grub` commands:

```
# grub
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
```

If you cannot boot your system (and it is hanging on `grub`), you need to boot the node in rescue mode and then issue the following commands:

```
# mount /dev/ /system
# mount -o bind /dev /system/dev
# mount -t proc proc /system/proc # optional
# mount -t sysfs sysfs /system/sys # optional
# chroot /system
# grub
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
# reboot
```

Backing up and Restoring the System Database

The SMC for SGI ICE systems management software captures the relevant data for the managed objects in an SGI SGI ICE system. Managed objects are the hierarchy of nodes described in "Basic System Building Blocks" on page 1. The system database is critical to the operation of your SGI SGI ICE system and you need to back up the database on a regular basis.

Managed objects on an SGI SGI ICE include the following

- SGI ICE system

One ICE system is modeled as a meta-cluster. This meta-cluster contains the racks each modeled as a sub-cluster.

- Nodes

System admin controller (admin node), rack leader controllers (leader nodes), service nodes, compute nodes (blades) and chassis management control blades (CMCs) are modeled as nodes.

- Networks

The preconfigured and potentially customized IP networks

- Nics

The network interfaces for Ethernet and InfiniBand adapters.

- The network interfaces for Ethernet and InfiniBand adapter.

The node images installed on each particular node.

SGI recommends that you keep three backups of your system database at any given time. You should implement a rotating backup procedure following the son-father-grandfather principle.

Procedure 3-14 Backing up and Restoring the System Database

To back up and restore the system database, perform the following steps:

- 1.

Note: A password is required to use the `mysqldump` command. The password file is located in the `/etc/odapw` file.

From the system admin controller, to back up the system database perform a command similar to the following:

```
# mysqldump --opt oscar > backup-file.sql
```

2. To read the dump file back into the system admin controller, perform a command similar to the following:

```
# mysql oscar < backup-file.sql
```

For more information, see the `mysqldump(1)` man page.

Enabling EDNS

Extension mechanisms for DNS (EDNS) can cause excessive logging activity when not working properly. SMC on SGI ICE contains code to limit EDNS logging. This section describes how to delete this code and allow EDNS to work unrestricted and log messages.

Procedure 3-15 Enabling EDNS

To enable EDNS on your SGI ICE system, perform the following steps:

1. Open the `/opt/sgi/lib/Tempo/Named.pm` file with your favorite editing tool.
2. To remove the limit on the `edns_udp_size` parameter, comment out or remove the following line:

```
$limit_edns_udp_size = "edns-udp-size 512;";"
```

3. Remove the following lines so that EDNS logging is no longer disabled:

```
logging {  
  category lame-servers {null; };  
  category edns-disabled { null; }; };
```

Firmware Management

The `fwmgr` tool and its associated libraries form a firmware update framework. This framework makes managing the various firmware types in a cluster easier.

A given cluster may have several types of firmware including mainboard BIOS, BMC, disk controllers, InfiniBand (`ib`) interfaces, Ethernet NICs, network switches, and many other types.

The firmware management tools allow the firmware to be stored in a central location (firmware bundle library) to be accessed by command line or graphical tools. The tools allow you to add firmware to the library, remove firmware from the library, install firmware on a given set of nodes, and other related operations.

License Requirement

This framework is licensed. It cannot be used without the appropriate license.

Terminology

This section describes some terminology associated with the firmware management, as follows:

- Raw firmware file

These are files that you download, likely from SGI, that include the firmware and option tools to flash said firmware. For example, a raw firmware file for an SGI ICE compute node BIOS update might be downloaded as, `sgi-ice-blade-bios-2009.12.14-1.x86_64.rpm`.

- Firmware bundle

A firmware bundle is a file that contains the firmware to be flashed in a way that the integrated tools understand. Normally, firmware bundles are stored in the firmware bundle library (see below). However, these bundles can also be checked out of the library and accessed directly in some cases. In most situations, a firmware bundle is a sort of wrapper around the raw firmware file(s) and various attributes and tools. A firmware bundle can contain more than one type of firmware. This is the case when the underlying flash tool supports more than one firmware type. An example of this is the SGI ICE compute node firmware, that contains several different BIOS files for different mainboards and multiple BMC firmware revisions. Another example might be a raw file that includes both the BIOS and BMC firmware for a given mainboard/server.

- Firmware bundle library

This is a storage repository for firmware bundles. The management tools allow you to query the library for available bundles and associated attributes.

- Update environment

Some raw firmware types, like the various SGI ICE firmware released as RPMs, run "live" on the admin node to facilitate flashing. The underlying tool may indeed set nodes up to network boot a low level flash tool, but there are many other methods used by the underlying tools. Some firmware types, like BIOS ROMs with associated flash executables, require an update environment to be constructed. One type of update environment is a DOS Update Environment. This update environment may be used, for example, to construct a DOS boot image for the BIOS ROM and associated flash tool. A firmware bundle calls for a specific update environment. In this way, a firmware bundle with an associated update environment form the necessary pieces to facilitate booting of a DOS update

environment over the network that flashes the target nodes with the specified BIOS ROM (as an example).

Firmware Update High Level Example

This section describes the steps you need to take to update a set of nodes in your cluster with a new BIOS level, as follows:

- Download the raw firmware file for this system type. You might do this, for example, from SGI Supportfolio web site located at <https://support.sgi.com/login>.
- Add the raw firmware file to the firmware bundle library using a graphical or command line tool.
- The tool will convert the raw firmware file into a firmware bundle and store it in the firmware bundle library. In some cases, you will be required to provide additional information in order to convert the raw firmware file into a firmware bundle. This could be information necessary to facilitate flashing that the framework can not derive from the file on its own.
- Once the firmware bundle is available in the firmware library, you can use the graphical or command line tool to select a firmware bundle and a list of target nodes to which to push the firmware update.
- The underlying tool then creates the appropriate update environment (if required) and facilitates flashing of the nodes.

Firmware Manager Command Line Interface (`fwmgr`)

The `fwmgr` command is the command line interface (CLI) to the firmware update infrastructure.

For a usage statement, enter `fwmgr --help`. The `fwmgr` command has several sub-commands, each of which can be called with the `--help` option for usage information.

You can use the `fwmgr` command to perform the following:

- List the available firmware bundles
- Add raw firmware files or firmware bundle files to the firmware bundle library. If it is a raw firmware type, it will be converted to a firmware bundle and placed in the library.

- Remove firmware bundles from the firmware bundle library
 - Rename an existing firmware bundle in the firmware bundle library
 - Install a given firmware bundle on to a list of nodes
 - Checkout a firmware bundle which allows you to store the firmware bundle itself
-

Note: It is currently not necessary to run the `fwmgrd` command (firmware manager daemon) to use the CLI.

Firmware Manager Daemon (`fwmgrd`)

This `fwmgrd` daemon is installed and enabled by default in SGIMC 1.3 on SGI ICE systems, only. This daemon provides the services needed for the SGI Management Center graphical user interface to communicate with the firmware management infrastructure. This daemon needs to be running in order to access firmware management from the graphical user interface.

Even if you intend to only use the CLI, it is recommended that the `fwmgrd` daemon be left running and available.

By default, the `fwmgrd` log file is located at:

```
/var/log/fwmgrd.log
```

View this log for important messages during flashing operations from the SGI Management Center graphical interface.

Notes specific to Management Center 1.3

The first release of the Firmware Management framework only supports SGI ICE firmware, released as RPMs. This includes: `sgi-ice-blade-bios`, `sgi-ice-blade-ib`, `sgi-ice-blade-zoar`, `sgi-ice-cmc`, and `sgi-ice-ib-switch`. This includes the SGI ICE compute nodes but does not yet include other managed node types.

SGI intends to expand this firmware management framework to support additional node types in SGI ICE and SGI Rackable cluster hardware in later releases.

Note: SGI ICE integrated InfiniBand switches are supported but only on SGI ICE 8400 series systems or later. Some integrated InfiniBand switch parts in the SGI ICE 8200 series systems will **not** flash properly with this framework.

System Fabric Management

The InfiniBand network on SGI ICE systems uses Open Fabrics Enterprise Distribution (OFED) software. This section describes the InfiniBand fabric and how to manage it. For background information on OFED, see <http://www.openfabrics.org>.

InfiniBand Fabric Management

This section describes the InfiniBand fabric and covers the following topics:

- "InfiniBand Fabric Overview" on page 265
- "InfiniBand Management Tool Graphical User Interface" on page 266
- "Fabric Component `sgifmcli` Command" on page 270
- "InfiniBand Fabric Management Configuration and Operation Overview" on page 274
- "InfiniBand Fabric Failover Mechanism" on page 279
- "Configuring the InfiniBand Fat-tree Network Topology" on page 282
- "Configuring the Lightweight Fabric" on page 283
- "Useful Utilities and Diagnostics" on page 285

InfiniBand Fabric Overview

InfiniBand fabric management on SGI ICE systems is done using the OFED OpenSM software package and the `sgifmcli` tool (see "Fabric Component `sgifmcli` Command" on page 270). The InfiniBand fabric connects the service nodes, rack leader controllers (leader nodes), and the compute nodes. It does not connect to the system admin controller (admin node) or the chassis management control (CMC) blades. SGI ICE systems usually have two separate InfiniBand fabrics, which are generally referred to as "ib0" and "ib1" within this manual, see "InfiniBand Fabric" on page 26.

Note: The LX series only has one ib fabric, "ib0". Any references to "ib1" in this manual do not apply to LX systems.

On SGI ICE 8200 systems, each InfiniBand fabric (also sometimes called an InfiniBand subnet) has its own subnet manager (SM), which runs on a rack leader controller (leader node). For a system with two or more racks, the SM for each fabric is usually configured to run on different leader nodes. In a single rack system, both SMs will run on the single leader node. Each SM may also be paired with a standby SM which can take over in the event of the failure of the primary SM. For more information, see "InfiniBand Fabric Failover Mechanism" on page 279.

On SGI ICE 8400 series systems, rack leader controllers (leader nodes) will not always have InfiniBand fabric host channel adapters (HCA) depending on the system configuration. In some cases, one to two RLCs will have HCAs to run the OFED subnet manager. In other cases, this will be done on separate fabric management nodes, in this case no RLCs will have InfiniBand HCAs.

Rack leader controllers associate a SM instance with a particular port on the leader node. Usually, `ib0` is mapped to port 1 of the InfiniBand host channel adapter (HCA) on the SM node, and `ib1` is mapped to port 2 of the HCA on the SM node (see Figure 1-10 on page 27). SM for `ib0` and `ib1` is configured using the corresponding `/etc/ofa/opensm-ib[01].conf` file.

Note: After a system reboot, the `opensm` daemons start running automatically.

SGI supports the following topologies: hypercube, enhanced hypercube, and fat tree.

InfiniBand Management Tool Graphical User Interface

You can use the InfiniBand management tool graphical user interface (GUI) to configure, administer, or verify the InfiniBand fabric on your SGI ICE system. You can use it to configure, start, stop, restart, cleanup, or get status for the InfiniBand fabric.

From the system admin controller (admin node), enter the following command:

```
admin:~ # smc-configure-fabric
```

The **InfiniBand Management Tool** GUI appears, as shown in Figure 4-1 on page 267.

You can also access this command from the `configure-cluster` GUI main menu **Configure Infiniband Fabric** option (see "configure-cluster Command Cluster Configuration Tool" on page 56). For more information, see Figure 4-1.

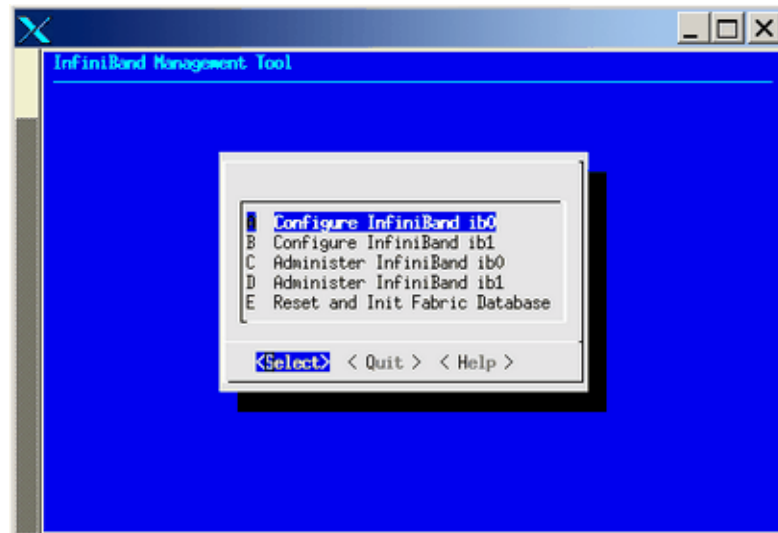


Figure 4-1 InfiniBand Management Tool Screen

Use the **Select** button to select the action you want to perform. A submenu will appear. Use the **Quit** button to return to the previous screen. Use the InfiniBand Management GUI to manage your InfiniBand fabric. You can use the **Help** button to get online help for each of the GUI actions.

If the `smc-configure-fabric` command fails in a configuration or administrative operation, it suggests that you use the `sgifmcli(8)` command (described in "Fabric Component `sgifmcli` Command" on page 270) to debug the problem. Alternatively, you can use the **Reset and Init Fabric Database** option from the **InfiniBand Management Tool** main menu (see Figure 4-1 on page 267) to start over and completely reconfigure the InfiniBand fabrics.

From the **Configure InfiniBand** screen, make sure you select the **Configure Topolgy** option to set the topology as shown in Figure 4-2 on page 268. For more information, see "Network Topology" on page 276.

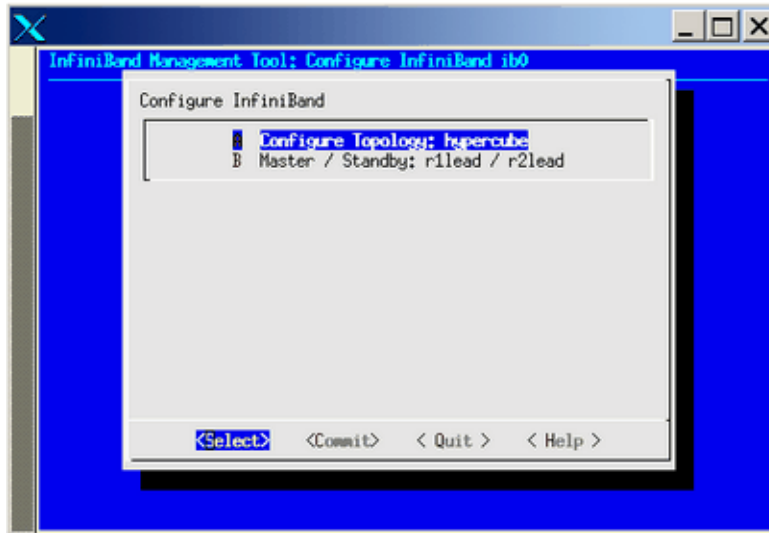


Figure 4-2 Configure Topology Screen

Use the the online help available with this tool to guide you through the InfiniBand configuration. After configuring and bringing up the InfiniBand network, select the **Administer InfiniBand ib0** option or the **Administer InfiniBand ib1** option, the **Administer InfiniBand** screen appears as shown in Figure 4-3. You can use this screen to start, stop, restart, or refresh a fabric.

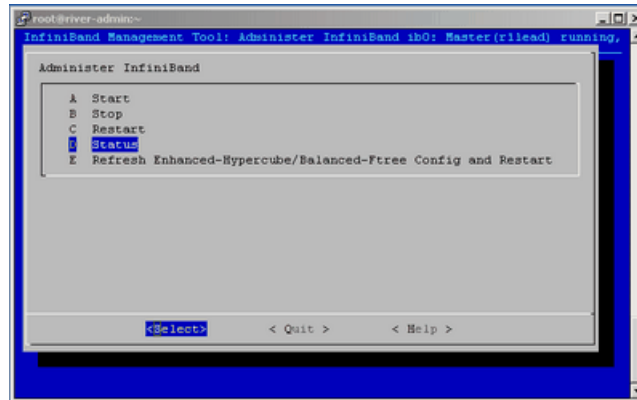


Figure 4-3 Administer InfiniBand Tool Screen

You can verify the status via the **Status** option, as shown in Figure 4-4 on page 269.



Figure 4-4 Administer InfiniBand Status Option

The **Refresh Enhanced Hypercube Config and Restart** option applies only to the Enhanced Hypercube topology. You are required to refresh the fabric configuration when you either add, remove, or move one or more compute blades or service nodes. The refresh action updates the `guid` routing order file which is used to balance InfiniBand traffic for the Enhanced Hypercube topology. In addition, this action also automatically restarts the master subnet manager (SM) and the optional standby SM for the specified fabric (see "InfiniBand Fabric Failover Mechanism" on page 279).

Ideally, the refresh action for a fabric should be taken when there are no jobs running in the system. Restarting the subnet manager can have an adverse impact on the running jobs in the system.

Fabric Component `sgifmcli` Command

For the most common fabric management operations, the `smc-configure-fabric` command (described in "InfiniBand Management Tool Graphical User Interface" on page 266) is entirely sufficient, and recommended. The `sgifmcli(8)` command can be used for more advanced fabric management tasks.

The most common operations that `sgifmcli` would be used for are, as follows:

- Initializing and configuring external InfiniBand switches
- Verifying the integrity of the InfiniBand fabric(s)

For more information, see the `sgifmcli(8)` man page.

Currently, the following switches are supported:

Switch Type	Description
<code>voltaire-isr-9024</code>	Voltaire ISR 9024
<code>voltaire-isr-2004</code>	Voltaire ISR 2004
<code>voltaire-isr-2012</code>	Voltaire ISR 2012
<code>voltaire-isr-9096</code>	Voltaire ISR 9096
<code>voltaire-isr-9288</code>	Voltaire ISR 9288
<code>voltaire4036</code>	Voltaire Grid Director 4036
<code>mellanox5030</code>	Mellanox IS5030

To configure an external InfiniBand switch, cluster-wide InfiniBand connectivity is not required. The only necessity is that the supplied switch host name is resolvable and a

working networking connection to the external InfiniBand switch exists. See the `sgifmcli(8)` man page for more information about adding external InfiniBand switches to your cluster's fabric.

Verify the integrity of an InfiniBand fabric requires that the InfiniBand network is first configured properly. This is most easily done using `smc-configure-fabric` (see "InfiniBand Management Tool Graphical User Interface" on page 266). See the `sgifmcli(8)` man page for details about the fabric verification operation.

`sgifmcli` SGI Fabric Component Command

The `sgifmcli(8)` command is, as follows:

```
sgifmcli [type action [options]] | [options]
```

Note: You can use shortened versions of the following `sgifmcli` options as long as the option is unambiguous. For example, `sgifmcli --vers` for `sgifmcli --version`.

It accepts the following general options:

General Option	Description
<code>-h, --help</code>	Displays a help message and the exits
<code>-V, --version</code>	Shows the version number of the program
<code>-v, --verbose</code> [DEBUG INFO ERROR]	Select verbosity level (default: ERROR). Most the messages from <code>sgifmcli</code> are written to a log file named <code>/var/log/sgifmcli.log</code> . The default level reports error messages only. INFO provides the user with details about the operation of <code>sgifmcli</code> in addition to error messages. The DEBUG level produces output that is tailored toward the developer to help with bug fixing. In addition, the DEBUG level also produces INFO and ERROR messages.

It accepts the following detailed options:

Detailed Option	Description
<code>type</code>	The <code>type</code> option is one of the following: <ul style="list-style-type: none"> <code>--mastersm</code> - Master subnet manager

- `--standby` - Standby subnet manager
 - `--ibswitch` - InfiniBand switch
 - `--ibfabric` - InfiniBand fabric
- `action` The `action` option is one of the following:
- `--init` - Initializes the switch or fabric
 - `--start` - Starts a subnet manager
 - `--stop` - Stops a subnet manager
 - `--status` - Prints the status of a subnet manager
 - `--verify` - Verifies the fabric
 - `--refresh` - Update a InfiniBand fabric (for Enhanced Hypercube)
 - `--set` - Sets specific SM configuration parameter (see `arglist`)
 - `--add` - Adds a subcomponent to its container, for example, add a switch to a fabric
 - `--delete` - Deletes a subcomponent from its container, for example, delete a switch from a fabric
Removes the switch or fabric
 - `--remove` - Removes an entity
 - `--showconfig` - Prints fabric configuration
 - `--switchlist` - Lists switches in a fabric
 - `--create-node-name-map` - Creates a node name map for internal SGI Alitx ICE switches
- `options` The `options` option is one or more of the following with no duplicates, for example, the `--fabric` option must be either `ib0` or `ib1`, not both:
- `--id` - Unique identifier, for example, host name
 - `--hostname` - Name of the node on which to run OpenSM

- `--switchtype` - Type of switch (leaf or spine)
- `--model` - Switch model (voltaire-isr-9024, voltaire-isr-2004, voltaire-isr-2012, voltaire-isr-9096, or voltaire-isr-9288)
- `--fabric` - Fabric, either `ib0` or `ib1`
- `--topology` - InfiniBand topology, either `hypercube`, `enhanced-hypercube`, or `ftree`
- `--arglist` - List of Subnet Manager configuration parameters: `param_1=val_1, param_2=val_2, ...`

EXIT CODES

To facilitate the use of the `sgifmcli(8)` command in shell scripts, an exit code is returned to give an indication of what occurred during a given connection.

The exit codes returned by `sgifmcli` are, as follows:

0	Successful termination.
255	Abnormal termination.

For a detailed man page, perform the following command from the admin node:

```
admin:~ # man sgifmcli
```

The `sgifmcli(8)` fabric administration utilities man page appears.

`sgifmdb` Fabric Management Database Command

The fabric component maintains a database (DB) of the objects it manages (managed objects). The database version is automatically set during cluster install. You do not need to set it. Most likely, this database will change over time. To manage multiple database versions and also to aid in field support, SGI has added another command line tool that currently reports the managed objects database version.

The `sgifmdb` command is, as follows:

```
sgifmdb [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]
```

It accepts the following general options:

General Option	Description
-g, --get	Reads the database version object from the database
-d, --dump	Dumps the database. This option allows the you to see what fabric objects are currently stored in the fabric database.
-v, --version	Prints version
-r, --reset	Resets the database and starts clean
-h, --help	-h, -help

Example 4-1 Getting `sgifmdb(8)` Command Help

For a `sgifmdb` command usage statement, perform the following from the admin node:

```
admin:~ # sgifmdb -h
SGI Fabric Component DB tool
Usage: db_version [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]

    -g, --get          Read DB version object from DB
    -d, --dump         Dump the DB
    -v, --version      Print version
    -r, --reset        Reset the database and start clean
    -h, --help         Show this text
```

InfiniBand Fabric Management Configuration and Operation Overview

Each subnet manager (SM) performs a light sweep of the fabric it is managing, every 10 seconds by default. The time interval is set by setting the `sweep_interval` variable in the `/opt/sgi/var/sgifmcli/opensm-ib0.conf.templ` file and then doing a **Commit** operation in the `smc-configure-fabric` GUI. Alternately, the `sgifmcli` command has a `--arglist` option to set various subnet manager configuration parameters including the sweep interval.

Note: If your cluster is larger than 256 nodes, SGI highly recommends increasing this variable to 90 seconds or even larger value.

If an SM detects a change in the fabric during a light sweep, such as, the addition or deletion of a node, it performs a *heavy* sweep. The heavy sweep actually changes the

fabric configuration to reflect the current state of the system. For more information, see the `opensm(8)` man page on the leader node.

The `opensm-ibx.conf` configuration files are located in the `/opt/sgi/var/sgifmcli` directory on the admin node.

Each `opensm` instance (one for each fabric) associates itself with a particular globally unique identifier (GUID) for a port on the node where `opensm` runs (see Figure 4-5 on page 275). This association is configured with the "guid" entry in the corresponding `opensm-ib[01].conf` file.

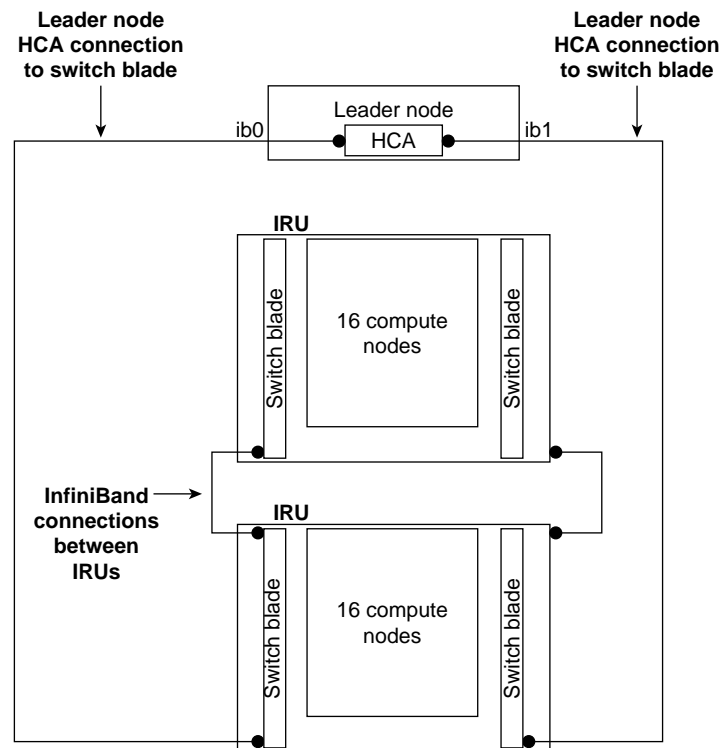


Figure 4-5 Two InfiniBand Fabrics in a System with Two IRUs

Network Topology

For SGI ICE systems with a hypercube topology, SGI uses the dimension order routing (DOR) algorithm.

The dimension order routing algorithm is based on the min hop algorithm and so uses shortest paths. Instead of spreading traffic out across different paths with the same shortest distance, it chooses among the available shortest paths based on an ordering of dimensions.

For SGI ICE systems with a fat-tree topology, SGI uses `updn` as the default routing algorithm. Unicast routing algorithm (UPDN) is also based on the minimum hops to each node, but it is constrained to ranking rules.

For more information on routing variables, see the `opensm(8)` man page.

Hypercube network topology is well suited for smaller node count MPI jobs or jobs that have communication patterns that are not sensitive to bisection bandwidth. Fat-tree network topology is well suited for large node count MPI jobs that are sensitive to bi-section bandwidth.

As stated above, there are two `opensm` daemons, one for each fabric, `opensmd-ib0` and `opensmd-ib1`, respectively. They are controlled by the `init.d` scripts. Each `init.d` script has a separate configuration file for each fabric, `opensm-ib0` and `opensm-ib1`, respectively.

You can use the `sminfo` command to show the GUID of the SM master.

Configuring the InfiniBand Fabric

This section describes how to configure and administer the InfiniBand fabric using the `sgifmcli(8)` command.

Note: SGI highly recommends that you use the `smc-configure-fabric` GUI to configure and administer the fabric (see "InfiniBand Management Tool Graphical User Interface" on page 266).

Procedure 4-1 Configure the Master Subnet Manager

When configuring the SM master, the following rules apply:

- Each InfiniBand fabric needs to have a subnet manager (SM) master.
- There can be at most one SM master per InfiniBand fabric.
- Fabric configuration and administration can only be done via the SM master.
- Fabric configuration becomes active after (re)starting the SM master.
- Deleting an SM master automatically deletes its standby, if it exists.

The syntax to configure an SM master is, as follows:

```
sgifmcli --mastersm --init --id identifier --hostname hostname --fabric fabric --topology topology
```

This command creates a master with the name provided by the `--id` option. The `identifier` can be any arbitrary string. The `hostname` determines the host on which the SM master manager is launched. The `fabric` option associates the SM master manager with either `ib0` or `ib1`. The `topology` option refers to the InfiniBand topology, which can be either hypercube, enhanced hypercube, or fat tree.

To configure a master for the fabric `ib0` on a hypercube cluster, perform the following steps:

1. From the admin node to configure an SM master, perform the following:

```
# sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube
```

This creates an SM master for `ib0`. The underlying topology is a hypercube and thus the routing algorithm `dor` will be used. This SM master, named `master_ib0`, is configured to run on the host `r1lead`.

2. The syntax to start an SM master is, as follows:

```
sgifmcli --start --id identifier
```

To start the `master_ib0` SM master, perform the following:

```
# sgifmcli --start --id master_ib0
```

At this point a master for the fabric `ib0` is running on the `r1lead` and thus the fabric `ib0` is available for compute jobs. If a standby has been defined, it will be launched automatically, in addition, to the master.

3. The syntax to stop an SM master is, as follows:

```
sgifmcli --stop --id identifier
```

To stop the `master_ib0` SM master, perform the following:

```
# sgifmcli --stop --id master_ib0
```

The SM master `master_ib0` running on host `r1lead` is stopped. If a standby has been defined then it will be stopped automatically, in addition to the master.

4. The syntax to check the status of an SM master is, as follows:

```
sgifmcli --status --id identifier
```

To check the status of the `master_ib0` SM master, perform the following:

```
# sgifmcli --status --id master_ib0
Master SM
Host = rlead
Guid = 0x0002c902002838f5
Fabric = ib0
Topology = hypercube
Routing Engine = dor
OpenSM = running
```

The status of the master SM master `master_ib0` running on host `r1lead` is reported. If a standby has been defined, its status will be reported in addition to the master.

5. The syntax to remove an SM master is, as follows:

```
sgifmcli --remove --id identifier
```

To remove the `master_ib0` SM master, first stop it and then perform the `-remove` option, as follows:

```
# sgifmcli --stop --id master_ib0

# sgifmcli --remove --id master_ib0
```

The SM master is removed from the entity list. If a standby has been defined, it is removed, in addition to the master.

6. To find the ID of the master SM in the database, perform the following:

```
# sgifmcli --dump --id ib0 | grep MASTER
```

7. To print the fabric configuration, run the following:

```
# sgifmcli --showconfig
```

```
-----  
NAME = ib1  
TYPE = ibfabric  
MASTER =  
STANDBY =  
SWITCH_LIST =  
-----  
NAME = ib0  
TYPE = ibfabric  
MASTER =  
STANDBY =  
SWITCH_LIST =
```

InfiniBand Fabric Failover Mechanism

Each subnet manager (SM) has a failover mechanism. If the master SM fails, the standby SM takes over operation of the fabric. This failover operation is performed automatically by the `opensm` software. Typically, `rack1` is the MASTER for the `ib0` fabric and `rack2` has the MASTER for the `ib1` fabric, as shown in Figure 4-6 on page 280.

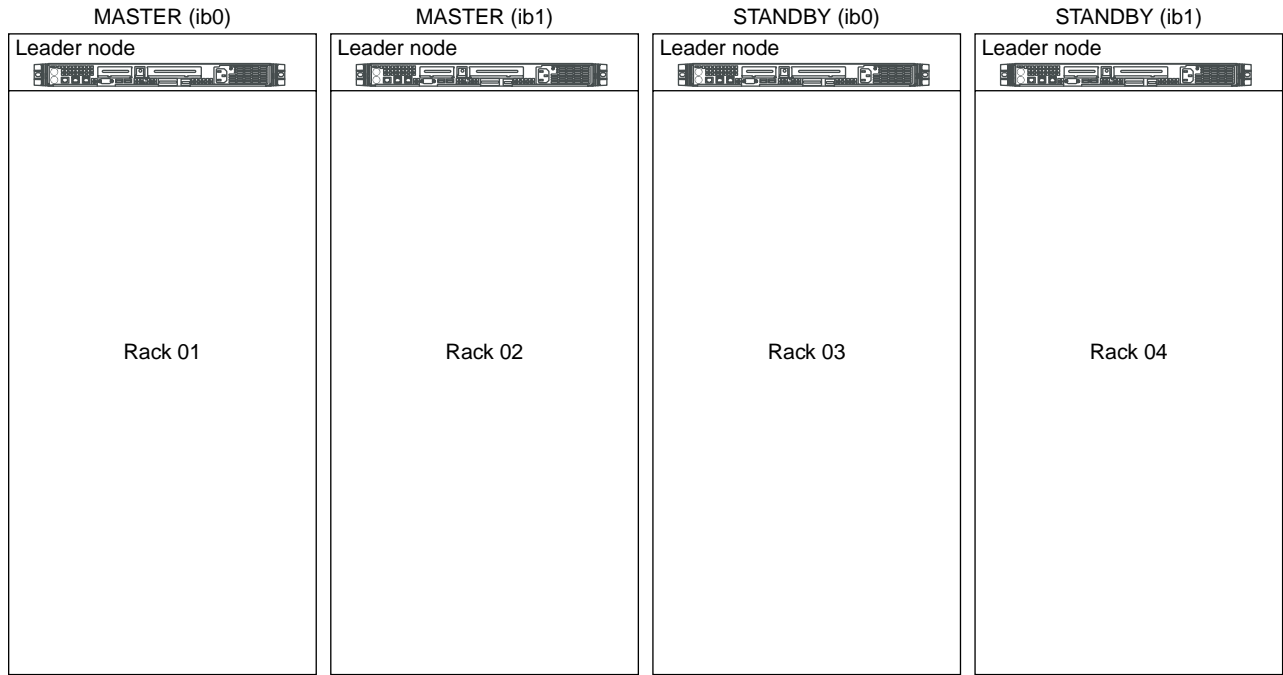


Figure 4-6 opensm Software Failover

The following procedure describes how to setup the failover mechanism.

Procedure 4-2 Enabling the InfiniBand Failover Mechanism

When enabling the InfiniBand failover mechanism, the following rules apply:

- Each InfiniBand fabric can optionally have exactly one standby.
- A standby SM can only be created for a particular fabric when a master already exists.
- When adding a standby after a master has already been defined and started, the master needs to be stopped before the standby is defined via the `--init` option. After defining the standby via `--init`, restart the master.
- A SM master and SM standby for a particular fabric can not coexist on the same node.

SGI highly recommends that you use the `smc-configure-fabric` GUI to configure the failover mechanism. If it is necessary to use `sgifmcli(8)` to enable the InfiniBand failover mechanism, perform the following steps:

1. If an SM master is defined and running, stop it, as follows:

```
# sgifmcli --stop --id master_ib0
```

If the SM master has not been defined, define it, as follows:

```
# sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube
```

2. Define the SM standby, as follows:

```
# sgifmcli --standbysm --init --id standby_ib0 --hostname r2lead --fabric ib0
```

3. Start the SM master, as follows:

```
# sgifmcli --start --id master_ib0
```

This automatically starts the SM master and the SM standby for `ib0`.

4. Now check the status for the subnet manager of `ib0`, as follows:

```
sgifmcli --status --id master_ib0
```

```
Master SM
Host = r1lead
Guid = 0x0008f10403987da9
Fabric = ib0
Topology = hypercube
Routing Engine = dor
OpenSM = running
Standby SM
Host = r2lead
Guid = 0x0008f10403987d25
Fabric = ib0
OpenSM = running
```

- 5.

To remove the `standby_ib0` SM standby, first stop its master and then perform the `remove` option, as follows:

```
# sgifmcli --stop --id master_ib0
# sgifmcli --remove --id standby_ib0
```

The SM standby is removed from the entity list. If a standby has been defined, it is removed, in addition to the master.

Configuring the InfiniBand Fat-tree Network Topology

This section describes how to configure InfiniBand fat-tree network topology. The fat-tree topology involves external InfiniBand switches. For the list of supported external switches, see "Fabric Component `sgifmcli` Command" on page 270.

InfiniBand switches are generally classified as being of two types: edge switches and core or spine switches. Edge switches are used to connect to compute nodes. Core or spine switches are used to connect edge switches together. The integrated InfiniBand switches in SGI ICE systems are considered to be edge switches and external InfiniBand switches used to connect these edge switches together in a fat-tree topology are considered to be spine switches.

The `sgifmcli` command allows two types of fat-tree topologies to be configured: FTREE and BFTREE. BFTREE is a balanced fat-tree. If the fat-tree topology is not balanced choose FTREE, otherwise; choose BFTREE for a balanced fat-tree.

SGI recommends that you use the SMC for SGI ICE `discover` command (see "discover Command" on page 80) to discover external IB switches. After discovery is completed, an external switch can also be initialized and added to the InfiniBand system using the `sgifmcli` command.

The `--init` and `--add` options below are completed by the SMC for SGI ICE `discover` command when the external switch is discovered with the `--switch` option. If the external switch is discovered not to be an external switch but as a general node, then the `--init` and `--add` options below, need to done.

Procedure 4-3 Configuring InfiniBand Fat-tree Network Topology

To configure the InfiniBand fat-tree network topology on an SGI ICE system, perform the following steps:

1. Make sure that your switch is properly connected to the InfiniBand network. Also, make sure that the admin port of the switch is properly connected to the Ethernet network.
2. Power on the switch. See the switch manual for operation information.

- From the admin node, initialize the switch. The syntax to initialize the switch is, as follows:

```
sgifmcli --init --ibswitch --model --id --switchtype [leaf | spine]
```

An example command is, as follows:

```
# sgifmcli --init --ibswitch --model voltaire-isr-2004 --id isr2004 --switchtype spine
```

This configures a Voltaire switch ISR2004 with hostname `isr2004` as a spine switch. `isr2004` refers to the admin port of the switch and needs to be configured previously to allow for switch access. The switch is now initialized and the root GUID from the spine switches have been downloaded.

- From the admin node, add the switch to the fabric. The syntax to add the switch is, as follows:

```
sgifmcli --add --id <fabric> --switch <hostname>
```

An example command is, as follows:

```
# sgifmcli --add --id ib0 --switch isr2004
```

In this example, ISR2004 is connected to the `ib0` fabric.

- For the new switch to be activated, the SM master and the optional SM standby need to be (re)started.

```
# sgifmcli --start --id master_ib0
```

If the SM master was running while the switch was added, you first need to stop and then start the master, as follows:

```
# sgifmcli --stop --id master_ib0
# sgifmcli --start --id master_ib0
```

If a standby has been defined, then in case of an SM master failure the SM standby subnet manager will automatically take over and assume control over the switch.

- The switches related to a particular fabric can be listed, as follows:

```
# sgifmcli --switchlist --id <fabric>
```

Configuring the Lightweight Fabric

This section describes how to configure the lightweight fabric with fat-tree topology using external Mellanox switches.

Procedure 4-4 Configuring the Lightweight Fabric

To configure the Lightweight Fabric, perform the following steps:

1. The switch should be setup to use dynamic host configuration protocol (DHCP), as part of the initial setup. This is done by SGI in the factory. You only need to go through the process if a new switch is being installed. For configuration information, see the Mellanox Technologies *IS5025/5030/5031/5035 Installation Guide*. See the section called "Configuring the switch for the First Time". When asked about using DHCP answer "yes". For IP configuration information, see Table 4 - "Configuration Wizard Session - IP Configuration by DHCP".
2. Use the `discover` command, to discover external switches. See "discover Command" on page 80. The switch model to be used is "mellanox5030". The `discover` command supports external switches in a manner similar to racks and service nodes, except that switches do not have BMCs and there is no software to install.
3. Discover all external switches.
4. Use `smc-configure-fabric` to configure the fabric, as described in "InfiniBand Management Tool Graphical User Interface" on page 266.

In the **Configure Topology** option, use **BFTREE** as the topology. The **FAT TREE** topology option should **not** be used. Proceed with the steps, described in "InfiniBand Management Tool Graphical User Interface" on page 266, to configure and verify the fabric.

Verifying the InfiniBand Network

After your InfiniBand fabric has been configured and started, you can use the `sgifmcli(8)` command to verify the health of the fabric.

Procedure 4-5 Verifying the InfiniBand Network

The fabric can be either `ib0` or `ib1`. This version of the InfiniBand verifier runs the recommended OFED test suite. In addition, the SMC for SGI ICE cluster view is compared with the InfiniBand cluster view and potential differences are reported.

To verify the `ib0` fabric, perform the following command:

```
# sgifmcli --verify --id <fabric>
```

For more information, see the `sgifmcli(8)` man page.

Useful Utilities and Diagnostics

The `infiniband-diags-pp` package contains useful tools and diagnostic software for Open Fabrics Enterprise Distribution (OFED). This section describes some of these tools. These tools reside on the rack leader controller (leader node) in the `/usr/sbin` directory. To see a full list of diagnostics, from the leader node, use the following command:

```
# rpm -ql infiniband-diags-pp | grep "/usr/sbin"
```

This section covers the following topics:

- "ibstat and ibstatus Commands" on page 285
- "perfquery Command" on page 287
- "ibnetdiscover Command" on page 288
- "ibdiagnet Command" on page 289

ibstat and ibstatus Commands

You can use the `ibstat` command to see the current status of the host channel adapters (HCA) in your InfiniBand fabric including the HCAs on rack leader controllers. The following view is **prior** to starting the fabric management:

```
r1lead:/usr/bin # ibstat
CA 'mthca0'
  CA type: MT25208 (MT23108 compat mode)
  Number of ports: 2
  Firmware version: 4.7.600
  Hardware version: a0
  Node GUID: 0x0008f104039881a8
  System image GUID: 0x0008f104039881ab
  Port 1:
    State: Initializing
    Physical state: LinkUp
    Rate: 20
    Base lid: 0
    LMC: 0
    SM lid: 0
    Capability mask: 0x02510a68
    Port GUID: 0x0008f104039881a9
```

```
Port 2:
  State: Initializing
  Physical state: LinkUp
  Rate: 20
  Base lid: 0
  LMC: 0
  SM lid: 0
  Capability mask: 0x02510a68
  Port GUID: 0x0008f104039881aa
```

The following shows output from the `ibstat` command **after** the fabric management software has been started:

```
r1lead:/opt/sgi/sbin # ibstat
CA 'mthca0'
  CA type: MT25208 (MT23108 compat mode)
  Number of ports: 2
  Firmware version: 4.7.600
  Hardware version: a0
  Node GUID: 0x0008f104039881a8
  System image GUID: 0x0008f104039881ab
  Port 1:
    State: Active
    Physical state: LinkUp
    Rate: 20
    Base lid: 1
    LMC: 0
    SM lid: 1
    Capability mask: 0x02510a6a
    Port GUID: 0x0008f104039881a9
  Port 2:
    State: Active
    Physical state: LinkUp
    Rate: 20
    Base lid: 1
    LMC: 0
    SM lid: 1
    Capability mask: 0x02510a6a
    Port GUID: 0x0008f104039881aa
```

You can use the `ibstatus` (less verbose than `ibstat`) command to show the link rate, as follows:

```
rllead:/opt/sgi/sbin # ibstatus
Infiniband device 'mthca0' port 1 status:
    default gid:    fe80:0000:0000:0000:0008:f104:0398:81a9
    base lid:       0x1
    sm lid:         0x1
    state:          4: ACTIVE
    phys state:     5: LinkUp
    rate:           20 Gb/sec (4X DDR)
```

```
Infiniband device 'mthca0' port 2 status:
    default gid:    fe80:0000:0000:0000:0008:f104:0398:81aa
    base lid:       0x1
    sm lid:         0x1
    state:          4: ACTIVE
    phys state:     5: LinkUp
    rate:           20 Gb/sec (4X DDR)
```

Note: If link rate is not 20 Gb/sec 4xDDR, and you have a DDR capable HCA, there is a physical link problem with your system.

perfquery Command

The `perfquery` command is useful for finding errors on a particular HCA (or a number of them) and switch ports. You can also use `perfquery` to reset HCA and switch port counters.

To see a usage statement for the `perfquery` command, perform the following:

```
rllead:/opt/sgi/sbin # perfquery --help
Usage: perfquery [-d(efug) -G(uid) -a(all_ports) -r(reset_after_read) -C ca_name -P ca_port -R(reset_only)
-t(timeout) timeout_ms -V(ersion) -h(help)] [<lid|guid> [[port] [reset_mask]]]
Examples:
    perfquery                # read local port's performance counters
    perfquery 32 1           # read performance counters from lid 32, port 1
    perfquery -e 32 1       # read extended performance counters from lid 32, port 1
    perfquery -a 32         # read performance counters from lid 32, all ports
    perfquery -r 32 1       # read performance counters and reset
```

```
perfquery -e -r 32 1 # read extended performance counters and reset
perfquery -R 0x20 1 # reset performance counters of port 1 only
perfquery -e -R 0x20 1 # reset extended performance counters of port 1 only
perfquery -R -a 32 # reset performance counters of all ports
perfquery -R 32 2 0x0fff # reset only error counters of port 2
perfquery -R 32 2 0xf000 # reset only non-error counters of port 2
```

Some sample output from the perfquery command is, as follows:

```
rllead:/opt/sgi/sbin # perfquery
# Port counters: Lid 1 port 1
PortSelect:.....1
CounterSelect:.....0x0000
SymbolErrors:.....0
LinkRecovers:.....0
LinkDowned:.....0
RcvErrors:.....0
RcvRemotePhysErrors:.....0
RcvSwRelayErrors:.....0
XmtDiscards:.....0
XmtConstraintErrors:.....0
RcvConstraintErrors:.....0
LinkIntegrityErrors:.....0
ExcBufOverrunErrors:.....0
VL15Dropped:.....0
XmtData:.....0
RcvData:.....0
XmtPkts:.....0
RcvPkts:.....0
```

ibnetdiscover Command

The `ibnetdiscover` command allows you discover the IB fabric.

To see a usage statement for the `ibnetdiscover` command, perform the following:

```
rllead:/opt/sgi/sbin # ibnetdiscover --help
Usage: ibnetdiscover [-d(ebug)] -e(rr_show) -v(erbose) -s(how) -l(ist)
-g(rouping) -H(ca_list) -S(witch_list)
-V(ersion) -C ca_name -P ca_port -t(imeout) timeout_ms
--switch-map switch-map] [<topology-file>]
--switch-map <switch-map> specify a switch-map file
```

Note: Only abbreviated output is shown in the this example.

Some sample output from the `ibnetdiscover` command is, as follows:

```

rlllead:/opt/sgi/sbin # ibnetdiscover
#
# Topology file: generated on Tue Jul 17 14:05:20 2007
#
# Max of 3 hops discovered
# Initiated from node 0008f104039881a8 port 0008f104039881a9

vendid=0x2c9
devid=0xb924
sysimgguid=0x8006900000000dd

...

Switch   : 0x08006900000000dc ports 24 devid 0xb924 vendid 0x2c9
"MT47396 Infiniscale-III Mellanox Technologies"
Switch   : 0x08006900000000a4 ports 24 devid 0xb924 vendid 0x2c9
"MT47396 Infiniscale-III Mellanox Technologies"

rlllead:/opt/sgi/sbin # ibnetdiscover -H (HCA's)
Ca       : 0x0030487aa7940000 ports 1 devid 0x6274 vendid 0x2c9 "MT25204 InfiniHostLx Mellanox Technologies"
Ca       : 0x0030487aa78c0000 ports 1 devid 0x6274 vendid 0x2c9 "rli0n8-ib0 HCA-1"
Ca       : 0x0008f10403988198 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"
Ca       : 0x0030487aa7840000 ports 1 devid 0x6274 vendid 0x2c9 "rli0n1-ib0 HCA-1"
Ca       : 0x0030487aa79c0000 ports 1 devid 0x6274 vendid 0x2c9 "rli1n0-ib0 HCA-1"
Ca       : 0x0030487aa7900000 ports 1 devid 0x6274 vendid 0x2c9 "rli1n8-ib0 HCA-1"
Ca       : 0x0030487aa7980000 ports 1 devid 0x6274 vendid 0x2c9 "rli1n1-ib0 HCA-1"
Ca       : 0x0008f104039881a8 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"

```

ibdiagnet Command

The `ibdiagnet` command is a useful diagnostic tool.

To see a usage statement for the `ibdiagnet` command, perform the following:

```
r1lead:/opt/sgi/sbin # ibdiagnet --help
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
```

NAME

`ibdiagnet`

SYNOPSIS

```
ibdiagnet [-c ] [-v] [-r] [-o ]
          [-t ] [-s ] [-i ] [-p ]
          [-pm] [-pc] [-P <>]
          [-lw <1x|4x|12x>] [-ls <2.5|5|10>]
```

DESCRIPTION

`ibdiagnet` scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.

It then produces the following files in the output directory defined by the `-o` option (see below):

- `ibdiagnet.lst` - List of all the nodes, ports and links in the fabric
- `ibdiagnet.fdb` - A dump of the unicast forwarding tables of the fabric switches
- `ibdiagnet.mcfdb` - A dump of the multicast forwarding tables of the fabric switches
- `ibdiagnet.masks` - In case of duplicate port/node GUIDs, these file include the map between masked Guid and real GUIDs
- `ibdiagnet.sm` - A dump of all the SM (state and priority) in the fabric
- `ibdiagnet.pm` - In case `-pm` option was provided, this file contain a dump of all the nodes PM counters

In addition to generating the files above, the discovery phase also checks for duplicate node/port GUIDs in the IB fabric. If such an error is detected, it is displayed on the standard output.

After the discovery phase is completed, directed route packets are sent multiple times (according to the `-c` option) to detect possible problematic paths on which packets may be lost. Such paths are explored, and a report of the suspected bad links is displayed on the standard output.

After scanning the fabric, if the `-r` option is provided, a full report of the fabric qualities is displayed.

This report includes:

- SM report
- Number of nodes and systems
- Hop-count information:

maximal hop-count, an example path, and a hop-count histogram
 All CA-to-CA paths traced
 Credit loop report
 mgid-mlid-HCAs matching table

Note: In case the IB fabric includes only one CA, then CA-to-CA paths are not reported.

Furthermore, if a topology file is provided, ibdiagnet uses the names defined in it for the output reports.

OPTIONS

-c : The minimal number of packets to be sent
 across each link (default = 10)

-v : Instructs the tool to run in verbose mode

-r : Provides a report of the fabric qualities

-o : Specifies the directory where the output
 files will be placed (default = /tmp)

-t : Specifies the topology file name

-s : Specifies the local system name. Meaningful
 only if a topology file is specified

-i : Specifies the index of the device of the port
 used to connect to the IB fabric (in case of
 multiple devices on the local system)

-p : Specifies the local device's port number used
 to connect to the IB fabric

-pm : Dumps all pmCounters values into ibdiagnet.pm

-pc : reset all the fabric links pmCounters

-P <>: If any of the provided pm is greater than its
 provided value, print it to screen

-lw <1x|4x|12x> : Specifies the expected link width

-ls <2.5|5|10> : Specifies the expected link speed

-h|--help : Prints this help information

-V|--version : Prints the version of the tool

--vars : Prints the tool's environment variables and
 their values

ERROR CODES

1 - Failed to fully discover the fabric
 2 - Failed to parse command line options
 3 - Failed to interact with IB fabric
 4 - Failed to use local device or local port

4: System Fabric Management

- 5 - Failed to use Topology File
- 6 - Failed to load required Package

Output which shows no errors means the system is operating correctly:

```
rllead:/opt/sgi/sbin # ibdiagnet
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdml.2
-W- Topology file is not specified.
    Reports regarding cluster links will use direct routes.
-W- A few ports of local device are up.
    Since port-num was not specified (-p option), port 1 of device 1 will be
    used as the local port.
-I- Discovering the subnet ... 10 nodes (2 Switches & 8 CA-s) discovered.

-I-----
-I- Bad Guids Info
-I-----
-I- No bad Guids were found

-I-----
-I- Links With Logical State = INIT
-I-----
-I- No bad Links (with logical state = INIT) were found

-I-----
-I- PM Counters Info
-I-----
-I- No illegal PM counters values were found

-I-----
-I- Bad Links Info
-I-----
-I- No bad link were found

-I- Done. Run time was 0 seconds.
```

You can use `ibdiagnet` to load the fabric to test it, as follows:

```
rlllead:/opt/sgi/sbin # ibdiagnet -c 5000
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdml.2
-W- Topology file is not specified.
    Reports regarding cluster links will use direct routes.
-W- A few ports of local device are up.
    Since port-num was not specified (-p option), port 1 of device 1 will be
    used as the local port.
-I- Discovering the subnet ... 10 nodes (2 Switches & 8 CA-s) discovered.

-I-----
-I- Bad Guids Info
-I-----
-I- No bad Guids were found

-I-----
-I- Links With Logical State = INIT
-I-----
-I- No bad Links (with logical state = INIT) were found

-I-----
-I- PM Counters Info
-I-----
-I- No illegal PM counters values were found

-I-----
-I- Bad Links Info
-I-----
-I- No bad link were found

-I- Done. Run time was 8 seconds.
```


System Maintenance, Monitoring, and Debugging

This chapter describes system monitoring and covers the following topics:

- "Maintenance Procedures" on page 295
- "Node Replacement Procedure for a Cold Spare Admin, Leader, and Service Nodes" on page 298
- "How To Avoid Out of Memory Occurrences on SLES11 and PBS Pro" on page 311
- "Inventory Verification Tool" on page 314
- "System Monitoring Overview" on page 315
- "System Monitoring Operation" on page 317
- "Monitoring System Events Using the Embedded Support Partner" on page 319
- "Monitoring System Metrics with Performance Co-Pilot" on page 334
- "Troubleshooting" on page 340
- "kdump Utility" on page 344
- "System Firmware" on page 345

Maintenance Procedures

This section describes some common maintenance procedures, as follows:

- "Temporarily Take a Node Offline for Maintenance" on page 295
- "Permanently Replace a Failed Blade" on page 296
- "Permanently Remove a Blade " on page 297
- "Add a New Blade" on page 298

Temporarily Take a Node Offline for Maintenance

This section describes how to temporarily take a node offline for maintenance.

Procedure 5-1 Temporarily Take a Node Offline for Maintenance

To temporarily Take a node offline for maintenance, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).

2. Power off the node, as follows:

```
# cpower --down r1i0n0
```

3. Mark the node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

4. Perform any maintenance to the blade that needs to be done.

5. Mark the node online, as follows:

```
# cadmin --set-admin-status --node r1i0n0 online
```

6. Power up the node, as follows:

```
# cpower --boot r1i0n0
```

7. Enable the node in the batch scheduler (depends on your batch scheduler).

Permanently Replace a Failed Blade

Note: See your SGI field support person for the physical removal and replacement of SGI ICE compute nodes (blades).

This section describes how to permanently replace a failed blade.

Procedure 5-2 Permanently Replace a Failed Blade

To permanently replace a failed blade (compute node), perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).

2. Power off the node, as follows:

```
# cpower --down r1i0n0
```

3. Mark the node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

4. Physically remove and replace the failed blade.
5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademon` daemon. See "Discovering Compute Nodes" on page 91, for more information.
6. Set the node to boot your desired compute image (see `cimage --list-images` and "cimage Command" on page 206 for your options), as follows:

```
# cimage --set mycomputeimage mykernel r1i0n0
```

7. Power up the node, as follows:

```
# cpower --boot r1i0n0
```

8. Enable the node in the batch scheduler (depends on your batch scheduler).

Permanently Remove a Blade

This section describes how to permanently remove a blade from your SGI ICE system.

Procedure 5-3 Permanently Remove a Blade

To permanently remove a blade from your system, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

```
# cpower --down r1i0n0
```

3. Mark the node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

4. Physically remove the failed blade.
5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademon` daemon. See "Discovering Compute Nodes" on page 91, for more information.

Add a New Blade

This section describes how to add a new blade to an SGI ICE system.

Procedure 5-4 Add a New Blade

To add a new blade to your system, perform the following steps:

1. Physically insert the new blade
2. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademon` daemon. See "Discovering Compute Nodes" on page 91, for more information.
3. Set the node to boot your desired compute image (see `cimage --list-images` and "cimage Command" on page 206 for your options), as follows:

```
# cimage --set mycomputeimage mykernel r1i0n0
```

4. Power up the node, as follows:

```
# cpower --boot r1i0n0
```

5. Enable the node in the batch scheduler (depends on your batch scheduler).

Node Replacement Procedure for a Cold Spare Admin, Leader, and Service Nodes

This section describe how to install and configure a spare admin, leader, or managed service node. The cold spare can be a shelf spare or a factory-installed cold spare that ships with your system. For more information on cold spare requirements and tools needed to do this procedure, see "Cold Spare Admin or Leader Node Availability" on page 299.

It covers the following topics:

- "Cold Spare Admin or Leader Node Availability" on page 299
- "Identify the Failed Unit and Unplug all Cables" on page 300
- "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304
- "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 306

- "Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode" on page 310

Note: When ordering shelf spare systems from SGI, it is important to order spare nodes appropriate to or in conjunction with your SGI ICE system. This is because the SGI ICE serial number is programmed into the admin node itself. If you try to migrate the admin node to a shelf spare system that does not have the correct SGI ICE system serial number programmed into it, parts of Tempo software may not work correctly. In particular, the Embedded Support Partner (ESP) software will fail to start if the system serial number does not match the number that was previously in use.

Depending on the system ordered, your SGI ICE system should be mounted in an SGI rack or racks. The system admin controller (admin node) and rack leader controller (leader node) are generally installed within (or in some cases on top of) the system rack. For an example, see Figure 1-1 on page 2. The replacement of a failed admin node or leader node is accomplished in four basic steps:

- Identify the failed unit and disconnect system and power cables.
- Transfer the disk drives from the failed server into the cold spare unit.
- Connect the applicable cables to the cold spare server.
- Power-up the new server and restart the ICE system.

For detailed procedures on installing a cold spare, see sections "Identify the Failed Unit and Unplug all Cables" on page 300, "Transfer Disks from Existing Server to the Cold Spare" on page 304, "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304 and "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 306.

Note: If you are using multiple root slots (making use of cascading dual-boot as described in "Cascading Dual-Boot" on page 136) the procedures described in this section will have to be repeated for each slot.

Cold Spare Admin or Leader Node Availability

A cold spare node is like an existing admin or leader node, but it sits on a shelf or is a factory preinstalled node to be used in an emergency.

If the admin or leader node should fail, the cold spare can be swapped in to position to take over the duties of the failed node.

If you wish to make use of cold spare nodes, SGI suggests that you have both an admin node and a leader node on the shelf as available spares. Some of the reasons to have two separate nodes instead of one are (not an exhaustive list), as follows:

- The BIOS settings of an admin and leader node are different. For example, an admin node does not PXE boot by default. However, a leader node must PXE boot each boot. This means the boot order is different for each type.
- The BMC of a leader node is set up to use DHCP by default. An admin node may not be set up this way.
- Given the examples cited about, if you try to use a shelf-spare admin node as a leader, the leader will not be properly discovered.

Shelf Spare Hardware Limitations

Currently, the hardware replacement procedure described in this section only supports SGI ICE `ice-csn` nodes, that is, admin controller and rack leader controller nodes and managed service nodes.

Tools Required

You will need a Video Graphics Array (VGA) screen and a keyboard to perform this procedure. This is because you need to interact with the LSI BIOS tool to import the root volumes. You cannot do this from an Intelligent Platform Management Interface (IPMI) serial console session because of the following:

- For leader nodes, the cluster does not know the MAC addresses of the replacement BMC so there is no way for the cluster to connect to it until the migration script is run.
- The LSI BIOS tool requires the use of `Alt` characters which often do not transfer through the serial console properly.

Identify the Failed Unit and Unplug all Cables

If you have already identified the failed admin node or leader node, proceed with disconnecting the cables from the failed unit. The front panel lights on the server can

indicate if the unit has failed and give you information on why, see Figure 5-1 on page 301.

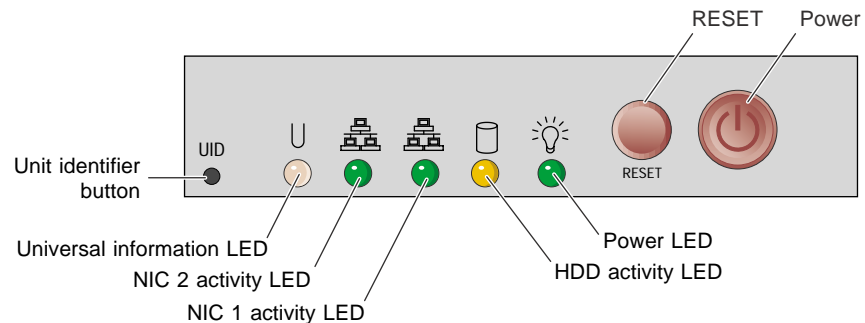


Figure 5-1 Admin/RLC Server Front Panel Controls and Indicator LEDs

The universal information LED (left side of the panel) shows two types of failure that can bring the server down. This multi-color LED blinks red quickly to indicate a fan failure and blinks red slowly for a power failure. A continuous solid red LED indicates a CPU is overheating.

If the unit's power supply has failed or been disconnected, the power LED (far right) will be dark. Check both ends of the power cable for a firm connection prior to switching over to the cold spare.

If you find that an admin node or leader node has failed and you need to replace it with a cold spare system, this section describes what to do in terms of the physical hardware.

Admin nodes are the only node type that store the system-wide serial number. Therefore, if you use a shelf spare leader node as an admin node, ESP will fail to start properly due to the system serial number mismatch and much of the logging and monitoring infrastructure will fail to function. The admin node shelf spares must be ordered from the factory as an admin node shelf spare so that the proper serial number can be stored within.

Procedure 5-5 Replacing a Node with a Cold Spare: Installing the Hardware

To replace an admin node or leader node that has failed, perform the following steps:

1. Power down the failed node (if possible).

2. Disconnect both power cables, see Figure 5-2 on page 303 for server connection locations.
3. Remove the two system disks from the failed node and set them aside for later reinstallation.
4. Unplug the Ethernet cable used for system management (be sure to note the plug number. Label the cables to avoid confusing them. It is important that they stay in the same jacks in the new node). See the example drawing in Figure 1-4 on page 6. This connection is vital to proper system management and communication.
The Ethernet cable must be connected to the same plug on the cold spare unit.
5. If the unit has a system console attached, remove the keyboard, mouse, and video cables.
6. Remove the system from the rack.
7. Install the shelf spare system into the rack.
8. Install the system disks you set aside in step 3 (from the system you are replacing).
9. Connect the Ethernet cables in the same way they were connected to the replaced node.
10. Connect AC power.
11. Connect a keyboard and VGA monitor (and mouse if you like).
12. Do **NOT** power up the system just yet. Proceed to "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304.

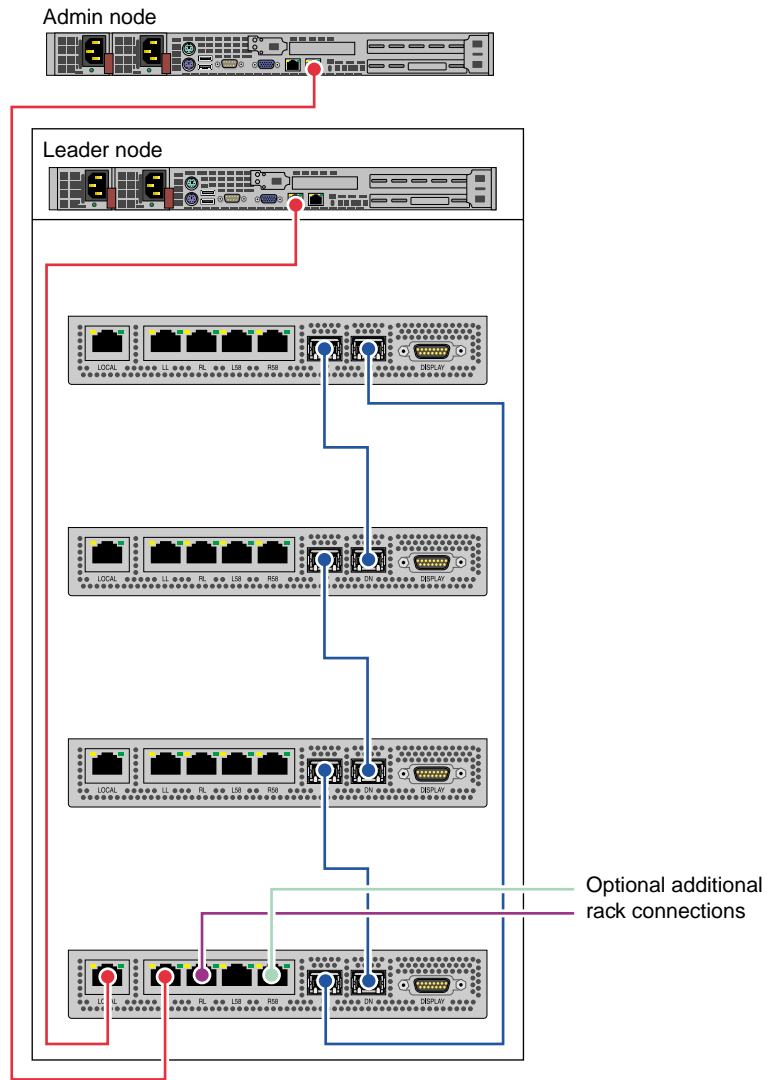


Figure 5-2 Simple CMC LAN (VLAN) Cable Examples

Transfer Disks from Existing Server to the Cold Spare

Note: The factory-installed cold spare does NOT ship with disks so you need to transfer existing disks and PCI cards from the existing server to the cold spare before mounting the spare rack.

Transfer disks from the existing server to the cold spare as shown in Figure 5-3 on page 304.

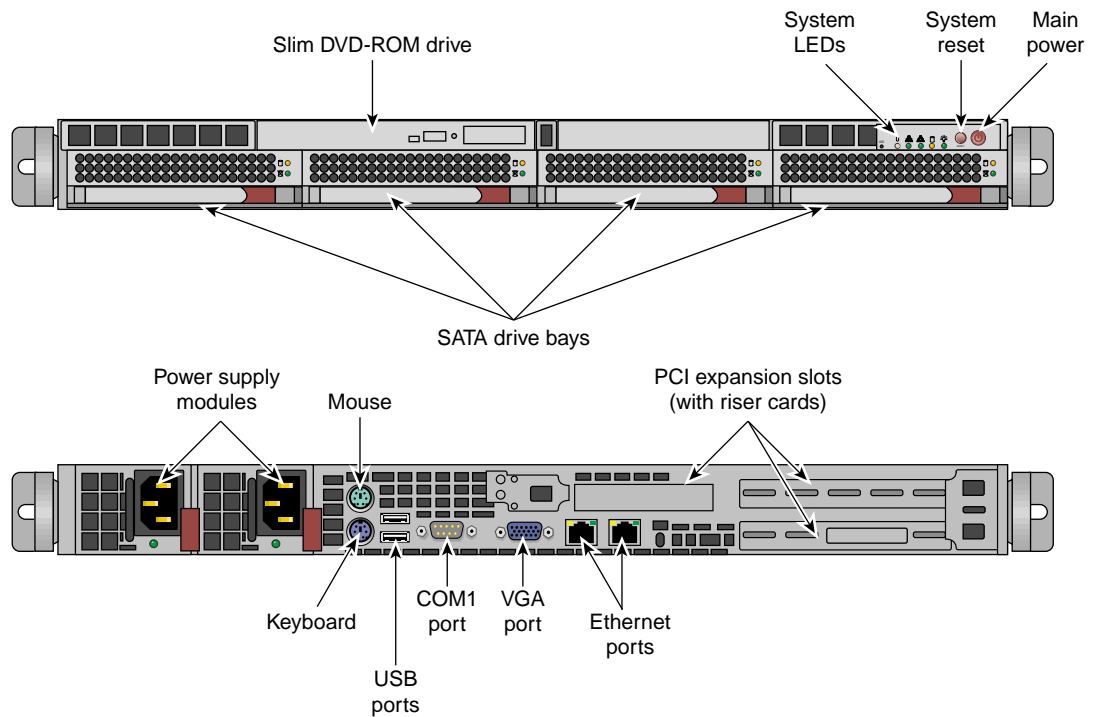


Figure 5-3 Admin/Leader Server Front Features and Rear Connector Locations

Migrating to a Cold Spare: Importing the Disk Volumes

This section describes how to import the disk volumes into the new node installed in "Identify the Failed Unit and Unplug all Cables" on page 300. For LSI 106x based systems, follow the procedure below. For LSI MegaRAID based systems, When you

import the disk pair from the dead system to the new one, they will automatically be imported.



Warning: You must use the same class of system for the shelf spare. That is to say, you cannot move disks formatted with LSI 106x RAID to a megaRAID based system and import the volume.

Although not supported, going from megaRAID to LSI 106x may allow manual importing of the data without data loss.

Procedure 5-6 Migrating to a Shelf Spare: Importing the Disk Volumes

To import the disk volumes into the new node, perform the following steps:

1. At this time, you can power up the system using the power button.
2. Watch the VGA screen output.
3. When you see the LSI BIOS tool come up, enter `Ctrl-C`. This will instruct the LSI BIOS tool to enter the configuration utility.
4. A screen appears listing the LSI controllers in the system. Normally, there is just one. Hit the `Enter` key to proceed.
5. Choose **RAID Properties**.
6. It is important to note that the controller supports only two RAIDs at a time. Therefore, if the system had two volumes at a time in the past, one or more volumes may appear empty now. It is important to use the utility to delete these empty volumes representing disks that are no longer installed before proceeding. Otherwise, if the tool sees more than one volume, activating volumes will not work.
7. Enter `Alt-N` to browse the list of volumes. Delete the empty ones as described in the step, above. Eventually, you will encounter an inactive volume. This inactive volume represents the disks you migrated from the failed node to this node.
8. With the inactive volume selected, choose **Manage Array**.
9. Choose **Activate** and answer `y` to the **activate and exit this menu** choice.
10. At this point, especially if the node has more than one volume, it is important to select the migrated system disk volume as the boot volume. To select the boot volume, choose **SAS Topology**.

11. In **SAS Topology**, you can expand the volumes to see the disks within them if you choose by hitting `Enter` on volumes.
12. Choose the volume that represents your newly imported volume. Highlight it, then enter `Alt-B`.
13. You should see that the volume now has a **Boot** flag associated with it.

Note: If, after you exit the tool, the system does not appear to boot from the disk. You may have selected the wrong volume from which to boot. In that case, reset, re-enter the LSI BIOS Tool, and choose a different volume to be the boot volume.

14. Escape out of the LSI tool and exit.
15. Keep watching the VGA screen! You will have to hit a key at the correct moment in the next section. Go to "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 306.

Migrating to a Cold Spare: Booting for the First Time on the Migrated Node

This section provides information on booting the system for the first time on a replacement node.

Note: Important: If your site is using cascading dual boot, only the currently used slot will be updated or repaired. Therefore, if the admin node is booted to slot 2, the fix up operations documented in these sections only apply to slot 2. The instructions need to be done for each slot you wish to fix up.

In a prior release, automatic recovery was implemented for cascading dual boot clusters. This means, if cascading dual boot is in use, when a managed service node or leader boots after having procedure 5-6 performed, it will go in to an automatic recovery boot, perform some fix up, then reboot again in to its normal operating mode. For the case of the admin node, a script is run by hand to integrate the repaired admin node with the cluster.

Note: Automatic Recovery is disabled by default because it can make certain discovery operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]# cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, when doing `discovery` operations, you may find it convenient to disable it.

For the case of the admin node, you will need to ensure your console output goes to the VGA screen and not serial-over-lan (SOL). For managed service nodes and leaders in cascading dual boot clusters, the default output location during the auto recovery boot is VGA. It is best to leave it VGA since part of the repair procedure will affect the network configuration for the BMC.

How do I know which procedure to follow?

- Admin nodes, all cases: Procedure 5-7. Migrating to a Cold Spare: Booting the Admin or leader/service node in non-cascading dual boot clusters for the first time.
- Managed Service, Leader nodes in a NON-cascading dual boot cluster: Procedure 5-7. Migrating to a Cold Spare: Booting the Admin or leader/service node in non-cascading dual boot clusters for the first time
- Managed service, Leader nodes in a cascading dual boot cluster: Procedure 5-8. Migrating to a Cold Spare: service/leader nodes using Cascading Dual Boot.

Procedure 5-7 Migrating to a Cold Spare in a Non-cascading Dual Boot Cluster Node

This section describes how to boot the admin node or a leader or service node in non-cascading dual boot clusters.

Note: This section applies to admin nodes and sites that are **not** making use of cascading dual boot. Cascading dual boot is set up by default on newer SMC for SGI ICE software releases. If you are using cascading dual boot, follow these instructions **only** for the admin node.

To boot for the first time on a migrated node, perform the following steps:

1. Ensure that the VGA console is powered on.
2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304).

Note: After rebooting, drive 1 will resync with drive 0, automatically. Drive 1 will have the RED LED on during this time. This process takes from eight to 48 hours depending on the drive size. During that period, the RAID redundancy is not available but the system will function normally.

When you see the GRUB boot menu come up, the first boot option will be highlighted by default. This should NOT be the choice starting with Failsafe. As an example, in SMC for SGI ICE 1.5 the highlighted choice should be : **SUSE Linux Enterprise Server 11 SP1**. Enter **e** to edit the boot parameters for this boot only.

3. Enter **e** to edit the kernel parameters.
4. Arrow down once so that the line starting **kernel** is highlighted.
5. Look at the settings. If no serial console is defined, you do not need to change anything. If a serial console is defined, append "`console=tty0`" to the end of the parameter list. This will ensure that console output goes to the VGA screen for this boot.

Note: By default, the admin node output goes to the VGA screen. Therefore, this adjustment does not need to be made. Leader and service nodes have serial consoles by default.

6. Press the `Enter` key.
7. Enter **b** to boot the system.

The system will now boot with console output going to the VGA screen.

Networking will fail to start and some error messages will appear.

It is normal to see that the Ethernet devices were renumbered. This will be fixed below.

Eventually the login prompt will appear.

8. Log in as root.

9. The following script fixes the network settings and update the SMC for ICE database for the new network interfaces, as follows:

```
# migrate-to-shelf-spare-node
```

Note: If you have additional Ethernet cards installed, you may need to check the settings of interfaces not controlled or managed by SMC for X ICE software.

10. Reboot the node and let it boot normally.

Procedure 5-8 Migrating to a Cold Spare: Service or Leader Using Cascading Dual Boot

This section describes what to do for managed service nodes and leader nodes in a cluster making use of cascading dual boot. It does **not** apply to admin nodes. For admin nodes, see Procedure 5-7, page 307.

To boot for the first time on a migrated node, perform the following steps:

1. Ensure that the VGA console is powered on.
 2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304).
-

Note: After rebooting, drive 1 will resync with drive 0, automatically. Drive 1 will have the RED LED on during this time. This process takes from eight to 48 hours depending on the drive size. During that period, the RAID redundancy is not available but the system will function normally.

3. At this time, you can plug the node in to AC power and press the power button on the front of the node.
4. Watch the VGA screen. The system should network boot in to recovery mode. It will do some repairs and reboot itself.
5. At this point, it will boot as a normal node. If, for some reason, it is unable to boot from the disk, the wrong volume may be selected as the boot disk in the LSI BIOS tool (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 304). It is true that the node network boots, but the network boot does a chainload to the first disk and it is still impacted by the BIOS and LSI firmware settings.

Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode

This section gives some advanced details on the Auto Recovery feature including how it is set up and how to control the feature.

Overview

The auto recovery feature allows managed service nodes and leader nodes to automatically make the necessary adjustments for both the node setup itself and the SMC for SGI ICE cluster database. This feature is mainly useful for clusters making use of cascading dual boot. The automated recovery mode applies to managed service nodes and leader nodes in cascading dual boot clusters. The goal is to provide an easy way for these nodes to perform any fix ups to themselves and the SMC for SGI ICE cluster at large when faulty systems are replaced.

Enable or Disable Auto Recovery Mode

Note: Automatic Recovery is disabled by default because it can make certain discovery operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]# cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, when performing discovery operations, you may find it convenient to disable it.

Use the `cadmin --show-auto-recovery` command to show the current state. Use the `cadmin --disable-auto-recovery` command to disable it.

IP Addresses Reserved for Auto Recovery Mode

Four IP addresses are reserved on the head network for auto recovery operations. For clusters being installed for the first time, these tend to be low numbers as they are reserved before any service or leader nodes are discovered. For systems being upgraded from previous SMC for SGI ICE releases, the allocated IP addresses are allocated the first boot after the upgrade and would tend to have higher numbers.

DHCP Set Up for Auto Recovery Mode

When the auto recovery feature is enabled, the `dhcpd.conf` file is configured with DHCP addresses available to unknown systems. That is, when this mode is enabled, any system attached to the head network that is performing DHCP requests will get a generic pool address and then boot in to the auto recovery mode. When the auto recovery mode is disabled, DHCP is configured to not offer these special IP addresses.

Auto Recovery and the `discover` Command

The auto recovery mode conflicts with the way that the `discover` command operates by default. Therefore, the `discover` command automatically and temporarily disables auto recovery (if it was enabled) for the duration of the run of the `discover` command. For more information on the `discover` command, see "discover Command" on page 80.

If you plan to discover a node, start `discover` before applying AC power. This is because auto recovery provides IP addresses to unknown nodes and because the `discover` command temporarily disables this, it is best to start the `discover` command before plugging in AC power to the node being discovered. Otherwise, it may get an unintended IP address.

How To Avoid Out of Memory Occurrences on SLES11 and PBS Pro

SGI ICE is a diskless blade server typically configured with `nfs` root and a small (50 MB) swap space that is served via `iscsi`. A maximum of 64 blades boot from a rack leader controller (leader node). The leader node typically has SATA disks in a mirrored pair for blade filesystems and blade swap space. Some users turn off swap entirely because a full rack of blades swapping has proven to be stressful to the rack leader nodes. When a Linux system has more memory requests than it can provide the kernel takes steps to defend the system using the out of memory (OOM) killer. The following section describes strategies for avoiding the loss of ICE blades due to OOM occurrences when the operating system is SLES11 and the batch scheduler is PBS Pro.

Some general guidelines are, as follows:

- Requesting the proper amount of memory is the first and most important strategy.
- If your application is correctly asking for memory then with PBS Pro configure MOM to enforce memory limits. "The Job Executor or MOM is the daemon/service which actually places the job into execution. This process,

`pbs_mom`, is informally called MOM as it is the mother of all executing jobs.” See the *PBS Professional 9.2 User’s Guide* for a complete description of MOM.

This only works well with when the SGI `memacct` function is installed to properly compute the amount of memory used. This requires that Linux kernel jobs and Comprehensive System Accounting (CSA) are installed. For more information, see the *Linux Resource Administration Guide*. CSA does not have to be configured to log. Modify `/var/spool/PBS/mom_priv/config` file by adding `$enforce mem` to the file. As an example, an application that just allocates memory one megabyte at a time will be killed once it goes over the limit. Applications that allocate in bigger chunks can still get above the limit before PBS can kill the job.

- The PBS Pro `enforce mem` variable has no configuration options. To avoid OOM occurrences you need your own daemon, such as the `policykill` daemon.

The `policykill` daemon looks for swapping in cpusets and works well in both large single-system image (SSI) with multiple cpusets and cluster (single cpuset). On large SSI, use of PBSPro’s cpuset mom is required. On SGI ICE systems use of SGI Altix bundle (example `PBSPro_10.1.0-SGIAltix_pp6_x86_64.tar.gz`) from Altair Engineering, Inc. is suggested. `policykill` has an `init` script, configuration file and daemon process itself. It requires customization for limits and notification methods.

- The Linux kernel Out Of Memory killer (`mm/oom_kill.c`) is responsible for keeping the system alive when memory has been exhausted. A snippet from the code is, as follows:

```
* The formula used is relatively simple and documented inline in the
* function. The main rationale is that we want to select a good task
* to kill when we run out of memory.
*
* Good in this context means that:
* 1) we lose the minimum amount of work done
* 2) we recover a large amount of memory
* 3) we don't kill anything innocent of eating tons of memory
* 4) we want to kill the minimum amount of processes (one)
* 5) we try to kill the process the user expects us to kill, this
*   algorithm has been meticulously tuned to meet the principle
*   of least surprise ... (be careful when you change it)
```

You can use `arrayd` to manage what processes gets killed. For more information on `arrayd`, see the `arrayd(8)` man page and Chapter 3. “Array Services” of the

Linux Resource Administration Guide. arrayd has a configuration option to protect the daemon:

```
-oom oom_daemon,oom_child
```

Specify oom_adj (OutOfMemory Adjustments) respectively for the main arrayd daemon and arrayd children. The default is "-17,0", hence resulting in the arrayd daemon selected as a candidate by the oom kernel killer thread and children selected as candidates. The value range from -17 to 15.

Each pid has an oom_adj (/proc//oom_adj) that you can independently protect. In general, you want root owned processes to be protected and user processes to be able to be killed.

A combination of PBS prologue and cron can set the values at job start and through the job's life span. On SMC for SGI ICE systems, cron is configured off in 80-compute-distro-services which is in

```
/var/lib/systemimager/images/<your compute image>/etc/opt/sgi/conf.d/80-compute-distro-services
```

by commenting out the following line:

```
initDisableServiceIfExists cron
```

To just enable cron on a blade is **not** a good practice. Files in

```
/var/lib/systemimager/images/<your compute image>/etc/cron*
```

must be reviewed for correctness in mixed writeable and read-only environment. For example, sysstat, logrotate, suse.de-cron-local, are the only services available in /etc/cron* directories. For a list of sample scripts, see Appendix A, "Out of Memory Adjustment" on page 349.

- Virtual memory sysctl tuning tries to balance use of system resources for user jobs and for system threads. The default setup is skewed towards user jobs but in the face of OOM system threads need more resources. For more information on sysctl, see the sysctl(8) man page. For an SMC for SGI ICE system, the sysctl parameters might be predefined similar to the following:

```
# Give the kernel a bit more breathing room by requiring more free space
vm.min_free_kbytes = 131072
# Push dirty pages out faster
vm.dirty_expire_centisecs = 1000           # Default is 3000
vm.dirty_writeback_centisecs = 500        # Default (unchanged)
vm.dirty_ratio = 20                       # Default is 40
vm.dirty_background_ratio = 5             # Default is 10
```

If blades are run without swap, set the following variable:

```
vm.swappiness = 0
```

Inventory Verification Tool

You can use the SMC for SGI ICE inventory verification tool to query, take snapshots, analyze and compare the node and network inventory of a cluster. Various hardware, network and operating system configuration properties are available and are presented in user-specified formats.

Note: If you are reinstalling the system admin controller (admin node), you may want to make a backup of the cluster configuration snapshot that comes with your system so that you can recover it later. You can find it in the `/opt/sgi/var/ivt` directory on the admin node; it is the earliest snapshot taken. You can use this information with the interconnect verification tool (IVT) to verify that the current system shows the same hardware configuration as when it was shipped. For more information, see "Installing SMC for SGI ICE Admin Node Software" on page 42.

To make an inventory snapshot of an SGI ICE system, use the following command from the system admin controller (admin node).

```
admin:~ # ivt -M  
Making a cluster inventory snapshot. Takes a couple of minutes...
```

Each snapshot is assigned a unique number and marked with the date and time it was taken. Use the `ivt --L` command to list active snapshot information, as follows:

```
admin:~ # ivt -L  
1 2007-07-13.11:42:47
```

You can query (`-Q` option), compare (`-C` option) and analyze (`-S` option) existing snapshots. A variety of system hardware and configuration properties can be displayed. You can compare two snapshots to see what has changed or analyze a system snapshot for failed nodes and or see network fabric links.

You can use the `ivt -c cpu` command to show an inventory of the system compute blades and the number of CPUs each blade contains, as follows:

```
admin:~ # ivt -c cpu  
rli0n0 has 8 CPUs
```

```
rli0n1 has 8 CPUs
rli0n8 has 8 CPUs
rli1n0 has 8 CPUs
rli1n1 has 8 CPUs
rli1n8 has 8 CPUs
```

You can use the `ivt` tool to determine which compute nodes (blades) are up or down, as follows:

```
admin:~ # ivt -Q -w blades -f '$blade $sshstate'
rli0n0 up
rli0n1 down
rli0n8 up
rli1n0 up
rli1n1 down
rli1n8 up
```

You can use the `ivt` tool to determine the GigE Ethernet address for each compute node (blade) , as follows:

```
admin:~ # ivt -Q -w blades -f '$blade $gige_ip_addr'
rli0n0 192.168.159.10
rli0n1 192.168.159.11
rli0n8 192.168.159.18
rli1n0 192.168.159.26
rli1n1 192.168.159.27
rli1n8 192.168.159.34
```

For detailed information on how to use the `ivt` tool, see the `ivt(8)` man page or `ivt -h, --help` usage statement.

System Monitoring Overview

Ganglia is a scalable, distributed monitoring system for monitoring system for high-performance computing systems, such as the SGI ICE system. It displays web browser-based, real-time (on demand) histograms of system metrics, as shown in Figure 5-4 on page 316.



Figure 5-4 Ganglia System Monitor

Detailed information about the Ganglia monitoring system is available at: <http://ganglia.info/>.

SMC for SGI ICE has devised a Ganglia model for the SGI ICE system that makes maximum use of Ganglia’s highly scalable architecture: each compute node (blade) presents a single monitoring source sending its statistics to the rack leader controller. Therefore, the rack leader controller receives, at most, data from 64 blades. After collecting the data, the rack leader controller forwards aggregated rack statistics to the system admin controller (admin node). The rack leader controller also sends its own statistics to the system admin controller. The system admin controller presents the meta-aggregator for the entire SG ICE system. It collects data from all rack leaders and presents the cluster-wide metrics. This model enables SGI to scale-out Ganglia to very large cluster deployments.

The **Node View** as shown in Figure 5-5 on page 317 can aid in system troubleshooting. For every blade in the system, the **Location** field of the **Node View** shows the exact physical location of the blade. This is an extremely useful when trying to locate a blade that is down.

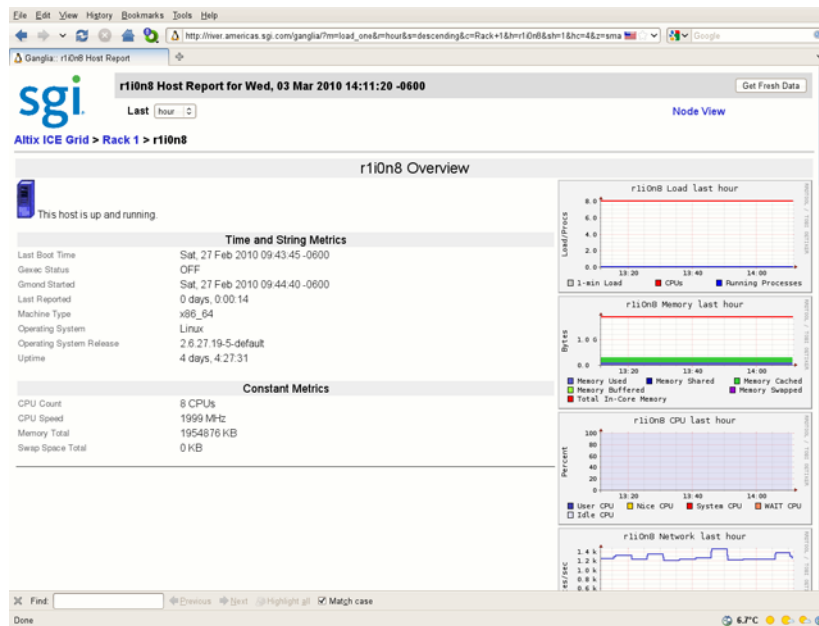


Figure 5-5 Ganglia System Monitoring Node View

System Monitoring Operation

This section describes the operation of the Ganglia system monitor and covers the following topics:

- "Accessing the Ganglia System Monitor" on page 317
- "Monitoring System Metrics" on page 318
- "SEL/Hardware Event Monitoring" on page 318
- "Node Availability Monitoring" on page 319

Accessing the Ganglia System Monitor

To access the Ganglia system monitor, point your browser to the following location:
http://admin_pub_name/ganglia

Monitoring System Metrics

By default, Ganglia monitors standard operating system metrics like CPU load, memory usage. The **Grid Report** view shows an overview of your system, such as the number of CPUs, the number of hosts (compute nodes) that are up or down, service node information, memory usage information, and so on.

The **Last** pull down menu allows you to view performance data on an hourly, daily, weekly, or yearly basis. The **Sorted** pull down menu allows provides an ascending, descending, or by host view of performance data. The **Grid** pull-down menu allows you to see performance data for a particular rack or service node. The **Get Fresh Data** button allows you to see current data performance.

SEL/Hardware Event Monitoring

The system admin controller, rack leader controllers, the service nodes, the chassis management controllers (CMCs) and all the compute nodes (blades) are equipped with a specialized controller, called the Board Management Controller (BMC). This unit provides a broad set of functions as described in the IPMI 2.0 standard. SMC for SGI ICE software uses the BMCs predominantly for remote power management, remote system configuration, and for gathering critical hardware events.

Currently, critical hardware events are gathered for the following nodes: rack leader controllers (leader nodes), CMCs and compute nodes (blades). These events are logged in the following locations:

- /var/log/messages via syslog
- var/log/sel/sel.log

All critical hardware events are summarized under the BMC_CMC event type. One particular event holds the following useful information:

```
MSG ::= <syslog-prefix> SMC:<node> EVENT:<event> APP:<app> Date:<date> VERSION:<version> TEXT <text>
```

The following fields are all of the type string:

<node>	node name, for example, r1i0n5
<event>	BMC_CMC
<app>	SEL-LOGGER
<date>	date / time of the event
<version>	1.0

`<text>` Exact copy of the hardware event description from the BMC

After reading the events from the BMCs, the BMC event logs are cleared on the controller to avoid duplicate events.

Node Availability Monitoring

The availability of each node in an SGI ICE system is monitored by a lightweight daemon called `smchbc`. Each managed service node, rack leader controller (leader node), and compute node runs this daemon and reports its status to the server which monitors it. The server daemon, which runs on the admin node and leader nodes, reports if the client is down after approximately 120 seconds. In this event, a HEARTBEAT Embedded Support Partner (ESP) event is generated. You can observe this event via `syslog` or using ESP. Furthermore, administrator-derived actions can be triggered, for instance sending an e-mail notification to the system administrator. For more information on ESP, see `esp(5)` man page and "Monitoring System Events Using the Embedded Support Partner" on page 319.

The HEARTBEAT event contains the following useful information:

```
MSG ::= <syslog-prefix> SMC:<node> EVENT:HEARTBEAT APP:SMCHBD Date:<date> VERSION:1.0 TEXT <text>
```

The HEARTBEAT event is created when nodes fail or recover, described by the `TEXT` field.

The following fields are all of the type string:

<code><node></code>	node name, for example, <code>r1i0n5</code>
<code><date></code>	date / time of the event
<code><text></code>	Description of event: <pre>'Heartbeat not detected'</pre> <pre>'Heartbeat lost'</pre>

Monitoring System Events Using the Embedded Support Partner

This section describes Embedded Support Partner (ESP) and covers the following topics:

- "Overview" on page 320
- "Setting up Automatic Call Logging" on page 322

- "Additional Useful ESP Utilities" on page 325
- "ESP Graphical User Interface" on page 331

Overview

This section describes the Embedded Support Partner (ESP) and covers the following topics:

- "Embedded Support Partner (ESP)" on page 320
- "System Group Manager (SGM)" on page 320
- "ESP Support for SGI ICE Systems" on page 321

Embedded Support Partner (ESP)

The Embedded Support Partner (ESP) provides an infrastructure for monitoring the health of an SGI system. ESP monitors an SGI system, looking for abnormal system events that suggest a potential problem has occurred. When system events do occur, ESP will pro-actively notify the system administrator so that corrective measures can be taken.

ESP was designed as an integral part of the SGI customer support process and is tied directly to the SGI service call system. One of the key features of ESP is its ability to forward system event data to a "back-end" server at SGI. There, the event data is compared to a set of business rules to determine if a service call should automatically be opened (the back-end server drills into the support call system, does entitlement verification, and so on). Note that only systems that are under warranty or service contract are eligible for automatic service call generation.

System Group Manager (SGM)

In addition to operating in single-system mode, ESP can be configured as a system group manager (SGM). When running in SGM mode, ESP manages configuration and system event data for a group of systems. Each SGM client system is identified by a unique system ID (SysID), which is usually the MAC address from one of the node's network interfaces. The SysID is part of all data records pertaining to that client. Each client system also has a unique system serial number (SSN). The SSN is necessary when seeking technical assistance for a problem on an SGI system. SGI Customer Support will use the SSN to determine if a system is under warranty or support contract. The back-end server at SGI will also use the SSN when determining

support entitlement. Note that a license is required to enable the SGM functionality. The license is available free of charge to customers with systems under warranty or who have an active software support agreement.

ESP Support for SGI ICE Systems

ESP provides full support for SGI ICE systems. An instance of ESP running on the system admin controller (admin node) manages the entire ICE cluster as an SGM managed group. Individual system nodes are added to the group as SGM client systems. Instances of ESP also run on each rack leader controller (leader node) and service node. On these nodes, however, ESP is configured to run in single-system mode. It is also configured so that no event data gets stored on the local node. Instead, all event data is forwarded to the admin node for processing and logging. In the case of the leader node, information pertaining to the chassis management controller (CMC) and all compute nodes contained within the rack the leader controls also gets forwarded to the admin node. This arrangement ensures that system event data for all nodes in an SGI ICE system will be stored in a centralized location. Administrators can obtain information about all nodes and all related system data using the ESP graphical user interface (ESP GUI) or various ESP command line tools. ESP can also be configured so that system event data for all nodes in an SGI ICE system is forwarded to the ESP back-end server at SGI for processing. Note that a license is not required for ESP to run in SGM mode on an ICE admin node. That is because the SGM managed group on the admin node only consists of nodes that are part of a single SGI ICE system.

Each ICE node in the SGM managed group has a unique system ID (SysID). This allows the SGM to track event data pertaining to each of the nodes. Each node does not, however, have a unique system serial number (SSN). Instead, all ICE nodes in an SGM managed group share the same SSN (the SSN for the system of which the node is part). That is because all ICE nodes are part of a system that was sold as a unit and is supported as a unit. This allows the ESP back-end server at SGI to determine support entitlement when events for various ICE nodes get forwarded to SGI for processing.

With a traditional ESP SGM configuration, client systems are added to an SGM managed group manually by the system administrator. ESP obtains system information directly from the client system, plugging it into the system record in the ESP database. On an SGI ICE system, however, ESP automatically sets up the SGM managed group using information obtained from the cluster management database. It does not actually poll the various client systems to get the relevant system data. That would only be possible for leader and service nodes, in any case, as ESP is not installed on compute nodes in an SGI ICE rack.

SYSLOG messages from each compute node in a rack are forwarded to the leader node for processing. The instance of ESP running on the leader node uses regular expression pattern matching to parse the SYSLOG messages looking for potential problems. ESP system events are generated for any messages that map to known ESP event types. The hostname (ICE node name) and time-stamp are extracted from the SYSLOG message and plugged into the ESP event data record that is generated. The event data record is then forwarded to the admin node where it is processed and logged in the ESP database.

Various actions can be configured to take place when specific ESP events get logged. The types of actions that can be configured include: Email notification, Pop-up window notification, and Custom action (path to any utility plus arguments). ESP provides a throttling mechanism for specifying how many seconds must pass before repeat actions for the same event type are launched. This is particularly important when actions are tied to events happening in very rapid succession.

ESP comes with a number of standard actions already in place. One that is of particular interest to SGI ICE system administrators sends an email to root any time an SGI ICE specific event gets logged. For some event types, an email will be sent for every occurrence of the event. For other event types, there is a throttle in place that prevents a flood of emails for the same event happening on a repetitive basis.

Setting up Automatic Call Logging

As was mentioned in a previous section, ESP is automatically configured on all SGI ICE systems. There are, however, a number of things that the system administrator can do to ensure ESP adds the most value to the SGI system support process. Perhaps the most important thing is to make sure the system is registered with the ESP back-end at SGI. The following section outlines the steps that should be followed to set an SGI ICE system up for automatic call logging.

Set up the Customer Profile

The customer profile contains contact information for the person (customer) responsible for a given system. This information is required in order for ESP to automatically log calls with SGI. The information is used when performing an authentication check, and to determine if a system is under warranty or service contract. Without a customer profile, the ESP automatic call logging feature will be disabled.

You can use the `espadmin` utility or the ESP GUI to set up the customer profile. Whichever option you choose, you need to make sure all required fields contain valid data. Required fields include the following:

- First name
- Last name
- Phone number
- Email address
- Country

There are also a number of optional fields that provide additional information about the location of the system. Optional fields include the following:

- Site ID
- Address
- City
- State
- Postal/Zip Code

Using the `espadmin` command line utility is the easiest way to add or update customer profile information. The syntax of the `espadmin` command is, as follows:

```
espadmin -create customer_profile
    -fname <first name> -lname <last name>
    -phone <phone number> -email <email address>
    [-street1 <street address (line 1)>]
    [-street2 <street address (line 2)>]
    [-street3 <street address (line 3)>]
    [-city <city name>]
    [-state <state or province>]
    [-post <postal/zip code>]
    -country <country>
    [-site_id <site id>]
    [-sgmclient <client alias>|-sysid <system id>]
```

Note that any data item provided as an `espadmin` command parameter that contains two or more words separated by blank space must be enclosed in quotes. An

example of how customer profile information would be entered when the command is run on the admin node is, as follows:

```
# espadmin -create customer_profile -fname FIRSTNAME -lname LASTNAME  
-phone 123-456-7890 -email myemail@mydomain.com -country "United States"
```

Also note that on an SGI ICE system, customer profile information only needs to be entered for the admin node. Since all nodes in an SGI ICE system are considered to be part of a single system, the same customer profile information will be included in events for each of the nodes in the system.

Sending Event Data to the ESP Database at SGI

One of the key features of ESP, is its ability to send event data to SGI for processing. ESP can send event data to SGI in one of two ways:

- Email

Event data sent to SGI via email can be sent in plain text form or it can be sent encrypted.

The following command enables call logging using email:

```
# espadmin -enable call_logging -log_type email
```

- Web Services

The Web Services interface provides a secure channel for communication between the client system and the ESP back-end server at SGI. All event data will be sent to the ESP back-end server via this secure channel. Note that the Web Services interface is the preferred method for sending ESP event data to SGI.

The following command enables call logging using Web Services:

```
# espadmin -enable call_logging -log_type web_services
```

You can also use the `espadmin` utility to check the status of call logging on the local system, as follows:

```
# espadmin -show call_logging
```

```
Call Logging : ENABLED  
Logging Mode : 2  
Logging Type : WEB_SERVICES  
Log Server : PRODUCTION
```

System	3048342B08
Call Logging	ON

There are a number of other useful utilities and commands that provide information about an SGI ICE system, as follows:

Additional Useful ESP Utilities

There are a number of other useful utilities and commands that provide information about an SGI ICE system, as follows:

- Get the SysID for a local system (node)

Use the `espsysid` command line utility to get the ESP SysID for a system, as follows:

```
# espsysid -s
3048342B08
```

The `espsysid` utility can also be used to determine if a given SysID value is valid for the local system. This is useful on a system where individual nodes can have multiple network interfaces and multiple SysID values are possible (any one of which will work as an ESP SysID). Note that on an SGI ICE system, it is only possible to use the `espsysid` utility on the admin, leader, and service nodes (ESP is not installed on compute nodes).

```
# espsysid -v 401E8537
VALID SYSID: 0x00000000401E8537 (LICENSEID)
```

- Get the system serial number for an ICE system

Use the `espsysid` command line utility on the ICE admin node to get the system serial number for the ICE system, as follows:

```
# espsysid --sys_serial
AA000036
```

- Determining the ICE system configuration

ESP contains a command line tool that provides information about all of the nodes in an SGI ICE system. The information comes directly from the cluster

management database and is arranged in hierarchical order based on actual cluster configuration.

```
# iceinfo
iceCluster : begin
  ClusterId : 1
  ParentId  : 1
  Name      : "oscar"
  iceNode  : begin
    NodeName   : "oscar_server"
    Hostname   : "shamu-admin.americas.sgi.com"
    HostAlias  : "shamu-admin"
    IpAddr     : "150.166.39.60"
    SysId      : 0x3048342b08
    NodeType   : 12
    NodeId     : 1
    ClusterId  : 1
    OpStatus   : 12
    AdminStatus : 16
  iceNode : end
  iceNode  : begin
    NodeName   : "service0"
    Hostname   : "service0"
    HostAlias  : "service0"
    IpAddr     : "172.23.0.7"
    SysId      : 0x30483355b0
    NodeType   : 6
    NodeId     : 3
    ClusterId  : 1
    OpStatus   : 12
    AdminStatus : 16
  iceNode : end
iceCluster : end
iceCluster : begin
  ClusterId : 2
  ParentId  : 1
  Name      : "rack1"
  iceNode  : begin
    NodeName   : "r1lead"
    Hostname   : "r1lead"
    HostAlias  : "r1lead"
```

```
        IpAddr      : "172.23.0.6"
        SysId       : 0x423d30966
        NodeType    : 8
        NodeId      : 2
        ClusterId   : 2
        OpStatus    : 12
        AdminStatus : 16
    iceNode : end
    iceNode : begin
        NodeName     : "r1i0n8"
        Hostname     : "r1i0n8"
        HostAlias    : "r1i0n8"
        IpAddr      : "192.168.160.108"
        SysId       : 0x30487aa8e6
        NodeType    : 4
        NodeId      : 12
        ClusterId   : 2
        OpStatus    : 9
        AdminStatus : 16
    iceNode : end
    iceNode : begin
        NodeName     : "r1i0c"
        Hostname     : "r1i0c"
        HostAlias    : "r1i0c"
        IpAddr      : "192.168.160.2"
        SysId       : 0x80069151c98
        NodeType    : 3
        NodeId      : 20
        ClusterId   : 2
        OpStatus    : -1
        AdminStatus : 16
    iceNode : end
    iceCluster : end
    iceStatus : begin
        STATUS_SHOULD_NOT_BE_INSTALLED : 1
        STATUS_SHOULD_BE_INSTALLED     : 2
        STATUS_UNDEFINED_1              : 3
        STATUS_UNDEFINED_2              : 4
        STATUS_UNDEFINED_3              : 5
        STATUS_UNDEFINED_4              : 6
        STATUS_UNDEFINED_5              : 7
```

```

STATUS_FINISHED           : 8
STATUS_OFF                 : 9
STATUS_NOT_BOOTED         : 10
STATUS_BOOTING            : 11
STATUS_BOOTED             : 12
STATUS_GOING_DOWN         : 13
STATUS_ERROR              : 14
STATUS_REMOVED            : 15
STATUS_ONLINE             : 16
STATUS_OFFLINE            : 17
STATUS_NOT_EXIST          : 18
STATUS_IGNORE             : 19
iceStatus : end
iceGroup : begin
    ICE_CORE               : 1
    ICE_DEFAULT            : 2
    ICE_OSCAR_CMC          : 3
    ICE_OSCAR_CLIENTS     : 4
    ICE_OSCAR_SWITCH_BLADES : 5
    ICE_OSCAR_SERVICE     : 6
    ICE_IMAGE              : 7
    ICE_OSCAR_LEADER      : 8
    ICE_OSCAR_MGMT_SWITCH : 9
    ICE_INCLUDED          : 10
    ICE_OSCAR_IB_SWITCH   : 11
    ICE_OSCAR_SERVER      : 12
    ICE_ALL                : 13
    ICE_DEFAULT_NODE      : 14
iceGroup : end

```

- Check to see which ICE nodes are included in the SGM managed group

There are a number of ways to check the current state of ESP, to see which ICE nodes have been added to the SGM managed group.

```
# espsconfig -show systems
```

List of known systems:

```

+-----+-----+-----+-----+-----+
| System Id   | IP address | System Hostname | Alias | Active|
+-----+-----+-----+-----+-----+
| 3048342B08 | 150.166.39.60 | shamu-admin.americas.sgi. | shamu-admin | Yes |
|             |             | com |             |     |

```

30483355B0	172.23.0.7	service0	service0	Yes
423D30966	172.23.0.6	r1lead	r1lead	Yes
30487AA8E6	192.168.160.108	r1i0n8	r1i0n8	Yes
80069151C98	192.168.160.2	r1i0c	r1i0c	Yes

- Make sure ESP is configured correctly on an SGI ICE system

As was mentioned in a previous section, ESP on an SGI ICE system is configured automatically. All data necessary to set the system up is contained in the cluster management database on the admin node. If there is any question about the current state of ESP configuration on an SGI ICE system, do the following to ensure ESP is configured correctly:

```
# sgadmin --setup ice_system --no_rmt_subscr
```

This command walks you through the node information in the cluster management database, checking to make sure each node is included in the SGM managed group. Any node this is missing will be added to the group along with all associated data (for example, event profiles).

- Setup ESP from scratch on an SGI ICE system

There may be times when you would like to wipe the ESP slate clean and start as if it were a new install. This is, in fact, fairly easy to do. Here are the specific steps you would follow:

- Check the status of the Event Manager and ESP to make sure they are both running, as follows:

```
# /etc/init.d/eventmonitor status
```

```
# /etc/init.d/esp status
```

The status for each service should be "running".

- Unload all ESP event profiles (this ensures all ESP events are unsubscribed), as follows:

```
# espadmin -unload eventprofile allprofiles
```

- Make sure that `availmon` (if configured on), the Event Manager, and ESP are all stopped, as follows:

```
# /etc/init.d/availmon stop
```

```
# /etc/init.d/esp stop
```

```
# /etc/init.d/eventmonitor stop
```

- Make a copy of Event Manager and ESP environments (just in case you would like to go back to your original configuration), as follows:

```
# cp -rf /var/esp /usr/tmp/esp
```

```
# cp -rf /var/emgr /usr/tmp/emgr
```

- Start the Event Manager, as follows:

```
# /etc/init.d/eventmonitor start
```

- Start ESP in Recovery Mode, as follows:

```
# /etc/init.d/esp recover
```

When this operation is complete, ESP will be set up as if it had just been installed from scratch and the ESP database will be reconstructed completely from scratch. On an SGI ICE system, all nodes in the SGI ICE system will have been added as SGM group members and all associated data for each of the nodes will be added to the ESP database. Existing event data will be erased. Any customer configuration, including customer contact information, registration with the ESP back-end system at SGI, and so on, will need to be repeated.

- Reverting to original ESP

If for any reason, you wish to revert to how ESP was configured before the start of the above operation, perform the following:

Make sure that ESP and the Event Manager `asr` stopped, as follows:

```
# /etc/init.d/esp stop
```

```
# /etc/init.d/eventmonitor stop
```

Preserve current ESP and Event Manager state (if desired), as follows:

```
# cp -rf /var/esp /usr/tmp/esp.hold
```

```
# cp -rf /var/emgr /usr/tmp/emgr.hold
```

Remove the current `esp` and `emgr` directories, as follows:

```
# rm -rf /var/esp
```

```
# rm -rf /var/emgr
```

Copy the original (saved) directory trees back, as follows:

```
# cp -rf /usr/tmp/esp /var
```

```
# cp -rf /usr/tmp/emgr /var
```

Start ESP, as follows:

```
# /etc/init.d/esp start
```

The `esp` init script will start up the Event Manager.

When this operation is complete, ESP will be restored to the exact state it was in when the `/var/esp` and `/var/emgr` directories were saved.

ESP Graphical User Interface

System administrators can use the ESP web-based graphical user interface (ESP GUI) to view details of ESP data, such as, system events, hardware and software configuration, and so on. The ESP GUI can also be used for the administration of ESP, however, this is not recommended on an SGI ICE system. That is because ESP is configured automatically on an ICE system and much of the configuration is custom and is SGI ICE platform specific. Please see the sections above for details on how to set up or reconfigure ESP on an SGI ICE system using various ESP command line utilities.

Before you can use the ESP GUI to access ESP data, you need to enable access described in the procedure below.

Procedure 5-9 Setting up the ESP GUI to Access ESP Data

To enable access to the ESP data using the ESP GUI, perform the following steps:

1. Log on to the admin node as **root** user.
2. Create the administrator account.

User permissions provide a layer of security to prevent access by unauthorized individuals using the ESP GUI. The ESP administrator account provides full access to a remote user to all ESP data. By default, the administrator account does not exist. Before using the ESP GUI, create the administrator account, as follows:

```
admin:~ # espcnfig -createadmin
```

Note: The administrator account is only used when connecting to a system using the ESP GUI. It is not necessary when using the ESP command line utilities because you need to be **root** user to use them.

3. Set up network access permissions.

ESP provides a mechanism for enabling and disabling access of specific systems or groups of systems to the ESP Web server. ESP does this through the use of network permissions. ESP network permissions provide a layer of security to prevent unauthorized access of ESP data.

By default, ESP comes with access disabled for all hosts except the local system. To see which IP addresses are enabled and disabled, perform the following:

```
admin:~ # espadmin -list ipaddr
```

##	IP address	Status
0.	127.0.0.0	enabled
1.	127.0.0.1	enabled
2.	*.*.*.*	disabled

To enable access to a specific hosts, perform the following:

```
admin:~ espadmin -enable ipaddr 134.15.68.189
```

To enable access to all hosts on a given subnet, perform the following:

```
admin:~ espadmin -enable ipaddr 134.15.68.*
```

If you would like to enable access for all hosts (**not recommended** unless you are in a completely secure environment), then issue the following command:

```
admin:~ espadmin -enable ipaddr "*.*.*.*"
```

You can also disable access for individual hosts or groups of hosts using the `espadmin -disable` command. Here are examples of how such commands would be issued:

```
admin:~ espadmin -disable ipaddr 134.15.68.189
```

```
admin:~ espadmin -disable ipaddr 134.15.68.*
```

To remove an IP address entirely from the network access list, use the `espadmin -delete` command. Here are examples of how such commands would be issued:

```
admin:~ espadmin -delete ipaddr 134.15.68.189
```

```
admin:~ espadmin -delete ipaddr 134.15.68.*
```



Caution: It is possible to make changes to network permissions using the ESP GUI. Care should be taken when doing this, however, so that you do not accidentally disable access for the system your browser is running on. Also note that it can sometimes take several minutes for the effects of a command to take effect.

4. To access the ESP GUI from your laptop or PC system, point your browser to the following URL: `http://mymachine__-admin:5554` and log into ESP.

When the ESP login screen appears, login as **administrator**, use the password **partner**. After you login, the **System Information** screen will appear.

Monitoring System Metrics with Performance Co-Pilot

A wealth of system metrics are also available through the Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The Performance Co-Pilot collection daemon (PMCD) runs on the admin node, managed service nodes, and rack leader nodes. A performance metrics domain agent (PMDA) is running on the rack leader nodes, which collects metrics from the compute nodes.

The new cluster metrics domain contains metrics that were previously available in other PMDAs. The method in which they are collected is different in a SMC for ICE system, in order to minimize load on the compute nodes. The following metrics are available for each compute node in a system by querying the PMCD on their rack leader node:

```
admin:~ # pminfo -h rtlead cluster
cluster.control.suspend_monitoring
cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
cluster.network.ib.in.bytes
cluster.network.ib.in.errors.drop
cluster.network.ib.in.errors.filter
cluster.network.ib.in.errors.local
cluster.network.ib.in.errors.remote
cluster.network.ib.out.bytes
cluster.network.ib.out.errors.drop
```

```

cluster.network.ib.out.errors.filter
cluster.network.ib.total.errors.link
cluster.network.ib.total.errors.recover
cluster.network.ib.total.errors.integrity
cluster.network.ib.total.errors.vl15
cluster.network.ib.total.errors.overrun
cluster.network.ib.total.errors.symbol

```

Configuring Compute Blade Metrics

The list of metrics that are monitored by the compute node and are pushed to the PMCD on the leader node is configurable. In some cases, it may be even be desirable to disable metric collection entirely, as follows:

```
# cexec --head --all pmstore cluster.control.suspend_monitoring 1 pmstore -h rllead cluster.control.suspend_monitoring
```

The default list of metrics that are collected by each compute node contains 41 metrics. There are dozens more available in the `cluster.*` namespace. The default list is stored on each leader node in the `/var/lib/pcp/pmdas/cluster/config` file. Changing this file will allow you to modify the default metric list with rack granularity. To change the list on a single node store a newline-delimited list of metrics to the node's instance of the `cluster.control.metrics` metric.

To see the current metric list for a compute node, perform the following:

```
# pmval -h rllead -s 1 -i 'rli1n0' cluster.control.metrics
```

```

metric:    cluster.control.metrics
host:      rllead
semantics: discrete instantaneous value
units:     none
samples:   1

```

```

                                rli1n0
"cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.nice
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem

```

```
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.infiniband.port.rate
cluster.infiniband.port.in.bytes
cluster.infiniband.port.in.packets
cluster.infiniband.port.in.errors.drop
cluster.infiniband.port.in.errors.filter
cluster.infiniband.port.in.errors.local
cluster.infiniband.port.in.errors.remote
cluster.infiniband.port.out.bytes
cluster.infiniband.port.out.packets
cluster.infiniband.port.out.errors.drop
cluster.infiniband.port.out.errors.filter
cluster.infiniband.port.total.bytes
cluster.infiniband.port.total.packets
cluster.infiniband.port.total.errors.drop
cluster.infiniband.port.total.errors.filter
cluster.infiniband.port.total.errors.link
cluster.infiniband.port.total.errors.recover
cluster.infiniband.port.total.errors.integrity
cluster.infiniband.port.total.errors.vl15
cluster.infiniband.port.total.errors.overrun
cluster.infiniband.port.total.errors.symbol
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
"
```

An example that changes the metric list to only include the CPU metrics for `r1i1n0` is, as follows:

```
# pmstore -h r1lead -i 'r1i1n0' cluster.control.metrics 'cluster.kernel.percpu.cpu.user cluster.kernel.percpu.cpu.nic
cluster.kernel.percpu.cpu.sys cluster.kernel.percpu.cpu.idle cluster.kernel.percpu.cpu.intr cluster.kernel.percpu.cpu
```

Monitoring SDR Metrics

The sensor data repository (SDR) metrics are available through Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The SDR provides temperature, voltage, and fan speed information for all service nodes, leader nodes, compute nodes, and CMCs. This information is collected from service and compute nodes through their BMC interface, so it is out-of-band and does not impact the performance of the node.

The following metrics are available through the PMCD:

```
admin:~ # pminfo -h r1lead sensor
sensor.value.fan
sensor.value.voltage
sensor.value.temperature
```

Each sensor will have a separate instance within the domain, with the instance of the form:

```
<nodeName>:<nodeType>:<metricName>
```

```
nodeName ::= SMC for SGI ICE node names (rXlead, rXiYc, rXiYnZ)
nodeType ::= "service", "cmc", "blade", "leader"
```

For example, to view voltages for the rack leader node, perform the following

```
admin:~ # pminfo -h r1lead -f sensor.value.voltage | grep -E '(^$|^sensor|r1lead)'
```

```
sensor.value.voltage
  inst [0 or "r1lead:leader:CPU1_Vcore"] value 1.3
  inst [1 or "r1lead:leader:CPU2_Vcore"] value 1.3
  inst [2 or "r1lead:leader:3.3V"] value 3.26
  inst [3 or "r1lead:leader:5V"] value 4.9
  inst [4 or "r1lead:leader:12V"] value 11.71
  inst [5 or "r1lead:leader:-12V"] value -12.3
  inst [6 or "r1lead:leader:1.5V"] value 1.47
  inst [7 or "r1lead:leader:5VSB"] value 4.9
  inst [8 or "r1lead:leader:VBAT"] value 3.31
```

For additional examples on how to retrieve values using `pmval(1)` and for using this data in trend analysis using `pmie(1)`, see the appropriate man page and the *Performance Co-Pilot Linux User's and Administrator's Guide*.

Turning Off the `temperature.pmie` Feature

Currently, in `temperature.pmie` there are values that will "Monitor: shut down components if temp too high". This feature is enabled by default as a safety mechanism. The procedure below describes how to turn it off.

Procedure 5-10 Turning Off the `temperature.pmie` Feature

To turn off the `temperature.pmie` feature, perform the following steps:

1. Edit the `/var/lib/pcp/config/pmie/control` file to comment out or remove the line that calls `/opt/sgi/lib/temperature.pmie`. For example,

```
#LOCALHOSTNAME n PCP_LOG_DIR/pmie/LOCALHOSTNAME/temperaturepmie.log -c /opt/sgi/lib/temperature.pmie
```

2. Run the `/etc/init.d/pmie restart` command. If you just want to adjust `temperature.pmie` values, see "Adjusting `temperature.pmie` Values" on page 338.

This has to be done on the admin and leader nodes. In that case, it is recommended that you turn it off on the leader images too.

Adjusting `temperature.pmie` Values

This section describes how to adjust `temperature.pmie` values.

Procedure 5-11 Adjusting `temperature.pmie` Values

You can adjust the warning or shutdown temperature values manually on the admin node and on each one of the leader nodes (if you choose to). In that case, it is recommended that you adjust it on the leader images too. The settings will be preserved between reboots. To change the values, perform the following steps:

1. Edit the `/opt/sgi/lib/temperature.pmie` file:

```
admin_warning_temperature = 68; // degree Celsius
admin_shutdown_temperature = 73; // degree Celsius
leader_warning_temperature = 68; // degree Celsius
leader_shutdown_temperature = 73; // degree Celsius
service_warning_temperature = 68; // degree Celsius
service_shutdown_temperature = 73; // degree Celsius
cmc_warning_temperature = 48; // degree Celsius
cmc_shutdown_temperature = 53; // degree Celsius
cn_warning_temperature = 68; // degree Celsius
```

```
cn_shutdown_temperature = 73; // degree Celsius
sensor_temperature = "sensor.value.temperature"; // degree Celsius
```

2. Perform the following command to verify that you updated the script correctly, as follows:

```
# pmie -C /opt/sgi/lib/temperature.pmie
```

If there are no errors, the `pmie -C` command returns with no message.

3. Run the `/etc/init.d/pmie restart` command or the `service pmie restart` command to restart the `pmie` service.

To turn off the `temperature.pmie` value, see "Turning Off the `temperature.pmie` Feature" on page 338.

Cluster Performance Monitor

You can use the Cluster Performance Monitor to monitor your SGI ICE system. Log into the admin node using the `ssh -X` command. Execute the `pmice` command and the **pmice - Cluster Performance Monitor** appears, as follows:

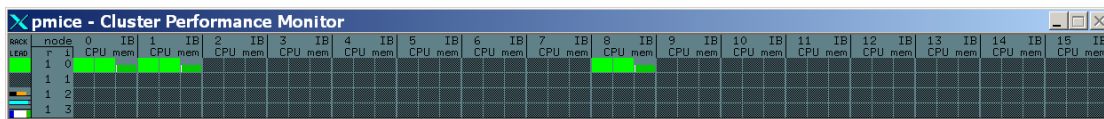


Figure 5-6 pmice- Cluster Performance Monitor

For a usage statement, use the `pmice --h` command, as follows:

```
admin:~ # pmice --h
/usr/bin/pmice: illegal option -- -
Info:
Usage: pmice [options] [pmsgadgets options]

options:
  -K list  Show these CPUs. Comma-separated list
  -N list  Show these nodes. Comma-separated list
  -R list  Show these racks. Comma-separated list
  -V      Verbose/diagnostic output
```

pmgadgets(1) options:

```
-C                check configuration file and exit
-h host          metrics source is PMCD on host
-n pmnsfile      use an alternative PMNS
-t interval      sample interval [default 2.0 seconds]
-z              set reporting timezone to local time of metrics source
-Z timezone      set reporting timezone

-zoom factor      make the gadgets bigger by a factor of 1, 2, 3 or 4
-infofont fontname use fontname for text in info dialogs
-defaultfont fontname use fontname for label gadgets

-display display-string
-geometry geometry-string
-name name-string
-title title-string
-xrm resource
```

Troubleshooting

This section describes some troubleshooting tools and covers these topics:

- "dbdump Command" on page 340
- "smc-info-gather Command" on page 342
- "cminfo Command" on page 343

dbdump Command

You can run the dbdump script to see an inventory of the SGI ICE database.

The dbdump command is, as follows:

```
/opt/sgi/sbin/dbdump --admin
/opt/sgi/sbin/dbdump --leader
/opt/sgi/sbin/dbdump --rack [--rack ]
/opt/sgi/sbin/dbdump
```

- Use the `--admin` argument to dump the system admin controller (admin node)
- Use the `--leader` argument to dump all rack leader controllers (leader nodes)
- Use the `--rack` argument to dump a specific rack
- Use the `dbdump` command without any argument to dump the entire SGI ICE system.

EXAMPLES

Example 5-1 dbdump Command Examples

To dump the entire database, perform the following:

```
admin:~ # dbdump
0 is { cluster=oscar ifname=service0-bmc dev=bmc0 ip=172.24.0.3 net=head-bmc node=service0
  nodetype=oscar_service mac=00:30:48:8e:
1 is { cluster=oscar ifname=service0 dev=eth0 ip=172.23.0.3 net=head node=service0
  nodetype=oscar_service mac=00:30:48:33:53:2e }
2 is { cluster=oscar ifname=service0-ib0 dev=ib0 ip=10.148.0.2 net=ib-0 node=service0
  nodetype=oscar_service }
3 is { cluster=oscar ifname=service0-ib1 dev=ib1 ip=10.149.0.2 net=ib-1 node=service0
  nodetype=oscar_service }
4 is { cluster=oscar dev=eth0 ip=128.162.244.86 net=public node=oscar_server
  nodetype=oscar_server mac=00:30:48:34:2B:E0 }
...
```

Note: Some of the sample output in this section has been modified to fit the format of this manual.

To dump just the rack leader controller, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --leader
0 is { cluster=rack1 ifname=r1lead-bmc dev=bmc0 ip=172.24.0.2 net=head-bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:8a:a4:c2 }
1 is { cluster=rack1 ifname=lead-bmc dev=eth0 ip=192.168.160.1 net=bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
2 is { cluster=rack1 ifname=lead-eth dev=eth0 ip=192.168.159.1 net=gbe node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
3 is { cluster=rack1 ifname=r1lead dev=eth0 ip=172.23.0.2 net=head node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
4 is { cluster=rack1 ifname=r1lead-ib0 dev=ib0 ip=10.148.0.1 net=ib-0 node=r1lead
```

```
nodetype=oscar_leader }
5 is { cluster=rack1 ifname=r1lead-ib1 dev=ib1 ip=10.149.0.1 net=ib-1 node=r1lead
nodetype=oscar_leader }
```

To dump just one rack, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --rack 1
0 is { cluster=rack1 ifname=i0n0-bmc dev=bmc0 ip=192.168.160.10 net=bmc node=r1i0n0
nodetype=oscar_clients mac=00:30:48:7a:a7:96 }
1 is { cluster=rack1 ifname=i0n0-eth dev=eth0 ip=192.168.159.10 net=gbe node=r1i0n0
nodetype=oscar_clients mac=00:30:48:7a:a7:94 }
2 is { cluster=rack1 ifname=r1i0n0-ib0 dev=ib0 ip=10.148.0.3 net=ib-0 node=r1i0n0
nodetype=oscar_clients }
3 is { cluster=rack1 ifname=r1i0n0-ib1 dev=ib1 ip=10.149.0.3 net=ib-1 node=r1i0n0
nodetype=oscar_clients }
4 is { cluster=rack1 ifname=i0n1-bmc dev=bmc0 ip=192.168.160.11 net=bmc node=r1i0n1
nodetype=oscar_clients mac=00:30:48:7a:a7:86 slot=1 }
5 is { cluster=rack1 ifname=i0n1-eth dev=eth0 ip=192.168.159.11 net=gbe node=r1i0n1
nodetype=oscar_clients mac=00:30:48:7a:a7:84 slot=1 }
6 is { cluster=rack1 ifname=r1i0n1-ib0 dev=ib0 ip=10.148.0.4 net=ib-0 node=r1i0n1
nodetype=oscar_clients slot=1 }
7 is { cluster=rack1 ifname=r1i0n1-ib1 dev=ib1 ip=10.149.0.4 net=ib-1 node=r1i0n1
nodetype=oscar_clients slot=1 }
8 is { cluster=rack1 ifname=i0n10-bmc dev=bmc0 ip=192.168.160.20 net=bmc node=r1i0n10
nodetype=oscar_clients slot=10 }
9 is { cluster=rack1 ifname=i0n10-eth dev=eth0 ip=192.168.159.20 net=gbe node=r1i0n10
nodetype=oscar_clients slot=10 }
10 is { cluster=rack1 ifname=r1i0n10-ib0 dev=ib0 ip=10.148.0.13 net=ib-0 node=r1i0n10
nodetype=oscar_clients slot=10 }
...
```

smc-info-gather Command

The `smc-info-gather` command enables to collect vital system data especially when troubleshooting problems. The `smc-info-gather` command collects the information about the following:

- Digital media `dminfo` files, syslogs, Dynamic Host Configuration Protocol (DHCP), network file system (NFS)
- MySQL cluster database dump

- Network service configuration files, for example, C3, Ganglia, DHCP, domain name service (DNS) configuration files
- A list of installed system images
- Log files in `/var/log/messages`
- Chassis management control (CMC) slot table for each rack
- basic input-output system (BIOS), Baseboard Management Controller (BMC), CMC and InfiniBand fabric software versions from all SGI ICE nodes

To see a usage statement for the `smc-info-gather` command, perform the following:

```
admin:/opt/sgi/sbin # smc-info-gather -h
usage: smc-info-gather [-h] [-P path] [-o file]
       smc-info-gather -h           # Print this usage page
       smc-info-gather -o file      # Tar and gzip the directories
into file (imply -n)
       smc-info-gather -p path      # Directory to write the data
(default /var/tmp/smc)
```

cminfo Command

The `cminfo` command is used internally by many of the SMC for SGI ICE scripts that are used to discover, configure, and manage an SGI ICE system.

In a troubleshooting situation, you can use it to gather information about your system. To see a usage statement from a rack leader controller, perform the following:

```
rllead:~ # cminfo --help
Usage: cminfo [--bmc_base_ip|--bmc_ifname|--bmc_iftype|--bmc_ip|--bmc_mac|--bmc_netmask|--bmc_nic|
--dns_domain|--gbe_base_i
p|--gbe_ifname|--gbe_iftype|--gbe_ip|--gbe_mac|--gbe_netmask|--gbe_nic|--head_base_ip|
--head_bmc_base_ip|--head_bmc_ifname|
--head_bmc_iftype|--head_bmc_ip|--head_bmc_mac|--head_bmc_netmask|--head_bmc_nic|--head_ifname|
--head_iftype|--head_ip|--he
ad_mac|--head_netmask|--head_nic|--ib_0_base_ip|--ib_0_ifname|--ib_0_iftype|--ib_0_ip|--ib_0_mac|
--ib_0_netmask|--ib_0_nic|
--ib_1_base_ip|--ib_1_ifname|--ib_1_iftype|--ib_1_ip|--ib_1_mac|--ib_1_netmask|
--ib_1_nic|--name|--rack]
```

```
r1lead:~ # cminfo --bmc_base_ip
```

EXAMPLES

Example 5-2 cminfo Command Examples

To see the rack leader node BMC IP address, perform the following:

```
r1lead:~ # cminfo --bmc_base_ip  
192.168.160.0
```

To see the rack leader DNS domain, perform the following:

```
r1lead:~ # cminfo --dns_domain  
ice.domain_name.mycompany.com
```

To see the BMC nic, perform the following:

```
r1lead:~ # cminfo --bmc_nic  
eth0
```

To see the IP address of the ib1 InfiniBand fabric, perform the following:

```
r1lead:~ # cminfo --ib_1_base_ip  
10.149.0.0
```

kdump Utility

The `kdump` utility is a `kexec`-based crash dumping mechanism for the Linux operating system. You can download `debuginfo` kernel RPMs for use with crash and any kernel dumps at the following location:<http://support.novell.com/linux/psdb/byproduct.html>.

To get a traceback or system dump, perform the following from the system console:

```
console r1i0n0  
^e c l l 8  
^e c l l t      #traceback  
^e c l l c      #dump
```

Note: This example shows the letter “c”, a lowercase L “l”, and the number one “1” in all three lines.

On the admin node, go to `/net/r1lead/var/log/ consoles` for the traceback and `/net/r1lead/var/log/dumps/r1i0n0` for the system dump.

You can dump a compute node, the rack leader, such as, `r1lead`, or a service node, such as, `service0`.

System Firmware

Note: Your SGI ICE system comes preinstalled with the appropriate firmware. See your SGI field support person for any BMC, BIOS, and CMC firmware updates.

The SGI ICE system firmware software consists of the following components:

```
sgi-ice-blade-bmc-1.43.5-1.x86_64.rpm
```

Blade BMC firmware and update tool

```
sgi-ice-blade-bios-2007.08.10-1.x86_64.rpm
```

Blade BIOS image and update tool

```
sgi-ice-cmc-0.0.11-2.x86_64.rpm
```

CMC firmware and update tool

BIOS Version Interrogation

To identify the BIOS you need both the version and the release date. You can get these using the `dmidecode` command. Log onto the node on which you want to interrogate BIOS level and perform the following:

```
# dmidecode -s bios-version; dmidecode -s bios-release-date
```

BMC Revision Interrogation

The BMC firmware revision can be retrieved using the `ipmiwrapper`. For example, from the admin node, the following command gets the BMC firmware revision for `r1i0n0`:

```
# ipmiwrapper r1i0n0 bmc info | grep 'Firmware Revision'
```

CMC Version Interrogation

The CMC firmware version can be retrieved using the `version` command to the CMC. For example, if you are logged onto the `r1lead` rack leader controller, the following command gets the CMC firmware version:

```
# ssh root@r1i0-cmc version
```

InfiniBand Version Interrogation

The `ibstat` command retrieves information for the InfiniBand links including the firmware version. The following command gets the InfiniBand firmware version:

```
# ibstat | grep Firmware
```

Getting Firmware Information for All System Nodes

The `firmware_revs` script on the system admin controller (admin node) collects the firmware information for all nodes in the SGI ICE system, as follows:

```
admin:~ # firmware_revs
BIOS versions:
-----
admin: 6.00
r1lead: 6.00
service0: 6.00
r1i0n0: 6.00
r1i0n1: 6.00
r1i0n8: 6.00
r1i1n0: 6.00
r1i1n1: 6.00
r1i1n8: 6.00
```

```
BIOS release dates:
-----
admin: 05/10/2007
r1lead: 05/10/2007
service0: 05/10/2007
r1i0n0: 05/29/2007
r1i0n1: 05/29/2007
```

```
rli0n8: 05/29/2007
rli1n0: 05/29/2007
rli1n1: 05/29/2007
rli1n8: 05/29/2007
```

BMC versions:

```
-----
admin: 1.31
r1lead: 1.31
service0: 1.31
rli0n0: 1.29
rli0n1: 1.29
rli0n8: 1.29
rli1n0: 1.29
rli1n1: 1.29
rli1n8: 1.29
```

CMC versions:

```
-----
rli0c: 0.0.9pre10
rli1c: 0.0.9pre10
```

Infiniband versions:

```
-----
r1lead: 4.7.600
service0: 4.7.600
rli0n0: 1.2.0
rli0n0: 1.2.0
rli0n1: 1.2.0
rli0n1: 1.2.0
rli0n8: 1.2.0
rli0n8: 1.2.0
rli1n0: 1.2.0
rli1n0: 1.2.0
rli1n1: 1.2.0
rli1n1: 1.2.0
rli1n8: 1.2.0
rli1n8: 1.2.0
```


Out of Memory Adjustment

This section describes sample set of out of memory OOM adjust scripts for cron and PBS prologue and epilogue.

Example A-1 oom_adj.user.pl.txt: OOM Adjustment Script

```
#!/usr/bin/perl
use strict;

use Sys::Hostname;
my $host = hostname();
my $DEBUG=0; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];

sub ResetOomAdj {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $CMD_LINE;
my $RETURN;

$AVOID_UIDS="root|100|nobody|ntp|USER|daemon|postfix|vtunesag";
  open (PS_CMD, "-|") || exec 'ps -e -o user,pid';
  while (<PS_CMD>) {
    chomp;
    ($_userid, $tpid) = split (/s+/, $_);

    if ( $_userid !~ m/^{AVOID_UIDS}/ && $tpid =~ /^[0-9]/ && -e
"/proc/$tpid/oom_adj" ) {
      print "$CALL_SCPT $host: Found processes to set to zero
oom_adj...\n" if $DEBUG;
      $CMD_LINE="echo 0 > /proc/$tpid/oom_adj";
      $RETURN=`$CMD_LINE`;
    }
    elsif ( $tpid =~ /^[0-9]/ && -e "/proc/$tpid/oom_adj" ) {
      print "$CALL_SCPT $host: Found processes to set to protect
oom_adj...\n" if $DEBUG;
      $CMD_LINE="echo -17 > /proc/$tpid/oom_adj";
      $RETURN=`$CMD_LINE`;
    }
  }
}
```

```
    }
  }
  close PS_CMD;
}

&ResetOomAdj();
```

Example A-2 cronentry: Sample cron Entry for oom_adj Script

```
*/2 * * * * /root/oom_adj.user.pl
```

Example A-3 prologue: Sample prologue Script

```
#!/bin/bash
#####
#
# Version: 2.3.1 : Updated 8/12/09
# Date: Oct 16, 2007
# Author: Scott Shaw, sshaw@sgi.com
#
# Script Name: PBS Pro Prologue Script
# The purpose of the Prologue script is to terminate leftover user processes and
# allocated IPCs resources. The prologue script consists of two scripts, the main
# prologue script and a chk_node.pl script. To minimize accessing each node the
# prologue script executes a parallel ssh shell across a set of nodes based on the
# PBS_NODEFILE. For large clusters over 64 nodes serial ssh access is slow so having
# a flexible parallel ssh to help speed up the clean-up process of each node. In
# some cases, a PBS jobs can normally terminate but some MPI implementations do not
# normally terminate the MPI processes due to crappy error code handling or
# segmentation faults within the MPI application thus leaving behind user processes
# still consuming system resources.
#
# When the prologue script is launched by PBS MOM the ssh session is executed and will
# execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
# commands which are executed on each node based on the PBS_NODEFILE.
#
# Execution of the prologue script is based on the root account.
#
```

```

# This script needs to reside on each execution host/node
# Location: /var/spool/PBS/mom_priv
# File name: prologue
# Permissions: 755
# Owner: root
# Group: root
#
# ls output: ls -l /var/spool/PBS/mom_priv/prologue
#      -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/prologue
#
# Modification of the prologalarm maybe necessary if the network access is slow to
# each node. 30 seconds may not be enough time to check 256 nodes in a cluster.
# prologalarm # Defines the maximum number of seconds the prologue
# and prologue may run before timing out. Default:
# 30. Integer. Example:
# $prologalarm 30
#
#####

```

```

JOBID=$1
USERNAME=$2
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

```

```
SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"
```

```

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

```

```

spawn ()
{
    if [[ `jobs | grep -v Done | wc -l` -ge $1 ]]; then
        wait
    fi
}

```

A: Out of Memory Adjustment

```
        fi
        shift
        $@ &
    }

exec_cmd ()
{
    for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
    do
        spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
    done
    wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
    QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
    QSTAT=/opt/pbs/default/bin/qstat

else
    echo "Epilogue Error: The qstat command could not be detected, exiting..."
    exit 1
fi

prefix_flag='${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk '{print $4}' | awk -F. '{print $1}'`
queue='${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk '{print $3}'`

    echo "Start Prologue ${VERSION} `date` "

    if [ $( /bin/uname -m ) = "x86_64" ]; then
        echo "Prefix passed: ${prefix_flag}"
        echo "destination queue: ${queue}"

        case $prefix_flag in
            TB)
                # Enable turbo and do node cleanup
                CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} TB"
            ;;
        esac
    fi
}
```

```

        exec_cmd
        ;;
    BP)
        # Bypass the turbo setting and P/Elog cleanup
        echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
        ;;
    JT)
        # Enable turbo but do not run the node cleanup p/elog scripts
        CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} JT"
        exec_cmd
        ;;
    NT)
        # bypass turbo settings but run the node cleanup
        CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} NT"
        exec_cmd
        ;;
    *)
        # disable turbo and run the node cleanup scripts
        CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue}"
        exec_cmd
    esac
else
    echo "The prologue script is intended to run on x86_64 nodes not 'uname -m'."
    echo "End Prologue ${VERSION} `date` "
    exit -1
fi
echo "End Prologue ${VERSION} `date` "

#Output the cluster details file
if [ -f /var/spool/PBS/mom_priv/cluster_info.out ]; then
    cat /var/spool/PBS/mom_priv/cluster_info.out
else
    echo "WARNING: The cluster info file does not exist. Contact hpc_support and report this warning."
fi

```

Example A-4 epilogue: Sample epilogue Script

```

#!/bin/bash
#####
#
# Version: 2.3.1 : Updated 8/12/09

```

A: Out of Memory Adjustment

```
# Date: Oct 16, 2007
# Author: Scott Shaw, sshaw@sgi.com
#
# Script Name: PBS Pro Epilogue Script
# The purpose of the epilogue script is to terminate leftover user processes and
# allocated IPCs resources. The epilogue script consists of two scripts, the main
# epilogue script and a chk_node.pl script. To minimize accessing each node the
# epilogue script executes a parallel ssh shell across a set of nodes based on the
# PBS_NODEFILE. For large clusters over 64 nodes serial ssh access is slow so having
# a flexible parallel ssh to help speed up the clean-up process of each node. In
# some cases, a PBS jobs can normally terminate but some MPI implementations do not
# normally terminate the MPI processes due to crappy error code handling or
# segmentation faults within the MPI application thus leaving behind user processes
# still consuming system resources.
#
# When the epilogue script is launched by PBS MOM the ssh session is executed and will
# execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
# commands which are executed on each node based on the PBS_NODEFILE.
#
# Execution of the epilogue script is based on the root account.
#
# This script needs to reside on each execution host/node
# Location: /var/spool/PBS/mom_priv
# File name: epilogue
# Permissions: 755
# Owner: root
# Group: root
#
# ls output: ls -l /var/spool/PBS/mom_priv/epilogue
#           -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/epilogue
#
# Modification of the prologalarm maybe necessary if the network access is slow to
# each node. 30 seconds may not be enough time to check 256 nodes in a cluster.
# prologalarm # Defines the maximum number of seconds the prologue
# and epilogue may run before timing out. Default:
# 30. Integer. Example:
# $prologalarm 30
#
#####
```

```
JOBID=$1
USERNAME=$2
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

spawn ()
{
    if [[ `jobs | grep -v Done | wc -l` -ge $1 ]]; then
        wait
    fi
    shift
    $@ &
}

exec_cmd ()
{
    for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
    do
        spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
    done
    wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
    QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
```

A: Out of Memory Adjustment

```
QSTAT=/opt/pbs/default/bin/qstat

else
  echo "Epilogue Error: The qstat command could not be detected, exiting..."
  exit 1
fi

prefix_flag='${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk '{print $4}' | awk -F. '{print $1}'`
queue='${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk '{print $3}'`

echo "Start Epilogue ${VERSION} `date` "
if [ $( /bin/uname -m ) = "x86_64" ]; then
  echo "Prefix passed: ${prefix_flag}"
  echo "destination queue: ${queue}"

  case $prefix_flag in
    TB)
      # Enable turbo and do node cleanup
      CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
      exec_cmd
      ;;
    BP)
      # Bypass the turbo setting and P/Elog cleanup
      echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
      ;;
    JT)
      # Enable turbo but do not run the node cleanup p/elog scripts
      CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset JT"
      exec_cmd
      ;;
    NT)
      # bypass turbo settings but run the node cleanup
      CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog noreset NT"
      exec_cmd
      ;;
    *)
      # disable turbo and run the node cleanup scripts
      CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
```

```

        exec_cmd
    esac
else
    echo "The epilogue script is intended to run on x86_64 nodes not `uname -m`."
    echo "End Epilogue ${VERSION} `date` "

    exit -1
fi
echo "End Epilogue ${VERSION} `date` "

```

Example A-5 chk_node.pl.txt: Script epilogue and prologue Use.

```

#!/usr/bin/perl
# Version: 2.3.1 : Updated 8/12/09
# Orig Date: Oct 10, 2007
# Author: Scott Shaw, sshaw@sgi.com
#
# This perl script is called by PBS Pro prologue and epilogue scripts when
# a user submits a job through PBS Pro. The purpose of this script is to
# sanitize a range of nodes identified by the $PBS_NODEFILE list by
# terminating old user processes, old ipc allocations, temp files,
# and to flush the system buffer cache.
#
# Changes:
# 2/1/08 sshaw@sgi.com
#     - Added a subroutine to clean-up /tmp directory
#     - changed system() to exec since it was corrupting memory
#     - declared all vars to be local to subroutine, before it was loosely defined
#     - added strict checking of perl script
# 3/24/08 sshaw@sgi.com
#     - fixed debug conditional
#     - cleaned up the CleanUpProcesses procedure and added which processes
#       and user being terminated.
#     - Changed the killall to pkill due to userid > 8 chars
# 11/13/08 sshaw@sgi.com
#     - added a subroutine to clean-up /dev/shm since several users
#       use this location for temporary scratch space.
# 03/31/09 sshaw@sgi.com
#     - added subroutines to enable/disable Turbo mode on Intel series 5500 CPUs
# 04/22/09 sshaw@sgi.com
#     - added subroutines to speed step the core processor frequency to a lower freq
# 08/12/09 sshaw@sgi.com

```

A: Out of Memory Adjustment

```
#           - fixed minor issues with setting the frequency and fixed cpu freq to max speed

use strict;

use Sys::Hostname;
my $host = hostname();
my $DEBUG=1; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];
my $queue_destination=$ARGV[1];
my $prefix_option=$ARGV[2];
my $set_freq=0;

#####
# The following lines are added for Turbo/SMT mode starting with Intel 5500 series CPUs
my $rdmsr = "/var/spool/PBS/mom_priv/rdmsr";
my $wrmsr = "/var/spool/PBS/mom_priv/wrmsr";
my $msr = "0x199";
my $tbit = 1 << 32;

# Several MPI implementations or MPI applications use IPC shared memory.  When
# a MPI application abnormally terminates it leaves behind allocated resources.
# this subroutine will remove any IPC resources allocated for the user's job.
sub CleanUpIPC_Table {
my $tkey;
my $tshmid;
my $towner;
my $tperms;
my $tbytes;
my $tnattch;
my $tstatus;
my $CMD_LINE;
my $RETURN;

open(IPC_SHARMEM, "-|") || exec 'ipcs -m';
while () {
chomp;
($tkey, $tshmid, $towner, $tperms, $tbytes, $tnattch, $tstatus) = split (/\\s+/, $_);
if ( $tkey =~ /^[0-9]/ ) {
if ( $towner !~ m/root|^ / ) {
```

```

        print "$CALL_SCPT $host: Found IPC_SHR_MEM allocation: $tshmid $towner, terminating...\n" if $DEBUG;
        $CMD_LINE="ipcrm -m $tshmid";
        $RETURN=`$CMD_LINE`;
    }
}
}
close IPC_SHARMEM;
}

```

```

# This subroutine will parse the process list and terminate any user processes or logins
# into the node(s)

```

```

sub CleanUpProcesses {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $tppid;
my $tcpu;
my $tstime;
my $ptty;
my $ttime;
my $tcmd;
my @TERM_USER;
my @TEMP;
my $USER;
my $CMD_LINE;
my $RETURN;

$AVOID_UIDS="root|100|101|nobody|bin|ntp|UID|daemon|postfix|vtunesag";
open (PS_CMD, "-|") || exec 'ps -ef';
while () {
    chomp;
    ($_userid, $tpid, $tppid, $tcpu, $tstime, $ptty, $ttime, $tcmd) = split (/\\s+/, $_);

    if ( $_userid !~ m/^{AVOID_UIDS}/ ) {
        if ( $_userid =~ /^[0-9]/ ) {
            $_userid=`ypcat passwd | egrep $_userid | cut -d ":" -f 1`;
            chomp $_userid;
        }
        print "$CALL_SCPT $host: Found leftover processes $tcmd from $_userid terminating...\n" if $DEBUG;
        $CMD_LINE="pkill -9 -u $_userid"; # Switched to pkill due to length of usernames.
        $RETURN=`$CMD_LINE`;
    }
}

```

```
    }
  }
  close PS_CMD;
  system("/root/oom_adj.user.pl");
}
# This subroutine will remove any temporary files created by MPI application under /tmp.
sub CleanUpTmp {
my $_filename;
my @TEMP;
my @TERM_FILE;
my $CMD_LINE;
my $RETURN;
my $_nofiles;
my $FILE;

open (LS_CMD, "-|") || exec 'ls /tmp';
while () {
  chomp;
  ($_filename) = split (/\\s+/, $_);
  if ( $_filename =~ m/^mpd/ ) {
    @TEMP=$_filename;
    push @TERM_FILE, $TEMP[0];
  }
  elsif ( $_filename =~ m/^ib_pool/ ) {
    @TEMP=$_filename;
    push @TERM_FILE, $TEMP[0];
  }
  elsif ( $_filename =~ m/^ib_shmem/ ) {
    @TEMP=$_filename;
    push @TERM_FILE, $TEMP[0];
  }
}
close LS_CMD;

foreach $FILE (@TERM_FILE) {
  $CMD_LINE="rm -f /tmp/${FILE}";
  $RETURN=`$CMD_LINE`;
}

$_nofiles = scalar @TERM_FILE;
```

```
    if ($_nofiles ne 0) {
        print "$CALL_SCPT $host: Found $_nofiles MPI temp files under /tmp. Removing...\n" if $DEBUG;
    }
}

# Flush the Linux IO buffer cache and the slab cache using the bcfree command.
sub FreeBufferCache {
    my $CMD_LINE;
    my $RETURN;
    my $BCFFREE;
    my $BCFFREE_OPTS;

    $BCFFREE="/usr/bin/bcfree";
    $BCFFREE_OPTS="-a -s";

    if (-e "${BCFFREE}") {
        $CMD_LINE="${BCFFREE} ${BCFFREE_OPTS}";
        $RETURN=`$CMD_LINE`;
    }
}

# This subroutine will remove any temporary files created by MPI application under /dev/shm.
sub CleanUpshm {
    my $_filename;
    my @TEMP;
    my @TERM_FILE;
    my $CMD_LINE;
    my $RETURN;
    my $_nofiles;
    my $FILE;

    open (LS_CMD, "-|") || exec 'ls /dev/shm';
    while () {
        chomp;
        ($_filename) = split (/s+/, $_);
        @TEMP=$_filename;
        push @TERM_FILE, $TEMP[0];
    }
    close LS_CMD;
}
```

A: Out of Memory Adjustment

```
foreach $FILE (@TERM_FILE) {
    if (${FILE} !~ m/sysconfig/) {
        $CMD_LINE="rm -rf /dev/shm/${FILE}";
        $RETURN=`$CMD_LINE`;
        print "${RETURN}" if $DEBUG;
        print "$CALL_SCPT $host: Found ${FILE} dir/file under /dev/shm. Removing it...\n" if $DEBUG;
    }
}

}

sub chk_msr_state {
# Hyperthreading Assumption, if the first core has the bit set to enable/disable
# then it is assumed all other cores within the node have the same setting.

my $msr_lsmode=`lsmod | grep -c msr`;    # 0=not loaded, 1=msr loaded

    if ( $msr_lsmode == 0 ) {
        print "Loading MSR Kernel Modules...\n";
        `modprobe msr`; # we need the msr kernel modules loaded to read the msr values
        sleep(1); # give time for the msr modules to load
    }
}

sub enable_turbo_mode {
my $ncpus = `cat /proc/cpuinfo | grep processor | wc -l`;
my $i;
my $val;
my $nval;

    chk_msr_state();
    print "${host}: Enabling turbo mode...\n";
    chomp($val = `rdmsr -p 0 $msr`);
    $val = hex("100000017");
    $nval = $val ^ $tbit;
    printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
    for ($i = 0; $i < $ncpus; $i++) {
        `wrmsr -p $i $msr $nval`;
    }
    load_system_services();
}
}
```

```
sub disable_turbo_mode {
my $ncpus = `cat /proc/cpuinfo | grep processor | wc -l`;
my $i;
my $val;
my $nval;
    chk_msr_state();
    print "${host}: Disabling turbo mode...\n";
    chomp($val = `rdmsr -p 0 $msr`);
    $val = hex(16);
    # $val = hex($val);
    $nval = $val ^ $tbit;
    printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
    for ($i = 0; $i < $ncpus; $i++) {
        `wrmsr -p $i $msr $nval`;
    }
}

sub load_system_services {
my $powersave_loaded=`ps -ef | grep -v grep | grep -c power`;

    if ($powersave_loaded == 0 ) {
        print "${host}: Loading system services...\n";
        system("/etc/init.d/acpid start;/etc/init.d/powersaved start)&> /dev/null");
        sleep(1);
        system("/usr/bin/powersave -f");
    }
    else {
        print "Powersaved already loaded.\n";
    }
}

sub unload_system_services {
    print "${host}: Unloading system services...\n";
    system("/etc/init.d/acpid stop;/etc/init.d/powersaved stop)&> /dev/null");
}

sub run_cleanup {
    &CleanUpshm();
    &CleanUpTmp();
    &CleanUpIPC_Table();
}
```

A: Out of Memory Adjustment

```
&CleanUpProcesses();
&CleanUpProcesses();
}

sub set_processor_speed {
my $freq=shift;
my $ncpus = `cat /proc/cpuinfo | grep processor | wc -l`;
my $i;
my $file;
    load_system_services();
    $freq = $freq * 1000;
    printf("${host}: Setting Proc Core speed to: %.3f GHz\n",($freq/1000000)) ;
    for ($i = 0; $i < $ncpus; $i++) {
        $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_min_freq";
        open FILE1, ">", $file or die $!;
        print FILE1 "$freq\n";
        close FILE1;

        $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_max_freq";
        open FILE2, ">", $file or die $!;
        print FILE2 "$freq\n";
        close FILE2;
    }
}

#
#print "$prefix_option\n";
#print "$queue_destination\n";
#
# if ( $queue_destination =~ /^f/ ) {
#     my $b=0;
#     ($a,$set_freq) = split (/f/, $queue_destination);
#     set_processor_speed($set_freq);
# }

# Don't run on systems with earlier than Nehalem processors
# Based on the prefix_option set turbo mode accordingly and run node cleanup routines.
    #if( $prefix_option =~ m/TB/ ){
        #enable_turbo_mode();
```

```
        #run_cleanup();
#}
#elsif ( $prefix_option =~ m/JT/ ) {
    #print " * * * * ENABLE TURBO and bypass PBS Prologue and Epilogue scripts * * * *\n";
    #enable_turbo_mode();
#}
#elsif ( $prefix_option =~ m/NT/ ) {
    #print " * * * * Bypassing the Turbo checks and run just node clean-up * * * *\n";
    #run_cleanup();
#}
#elsif ( $queue_destination =~ /^f/ ) {
    #my $b=0;
    #($a,$set_freq) = split (/f/, $queue_destination);
    #set_processor_speed($set_freq);
#}
#elsif ( $queue_destination =~ /^reset/ ) {
    #set_processor_speed(2934);
    #disable_turbo_mode();
    #unload_system_services();
    #run_cleanup();
#}
#}
#else {
    #disable_turbo_mode();
    #unload_system_services();
    #run_cleanup();
#}

run_cleanup();
```

Index

A

- admin node
 - installing software, 42
- attribute
 - boot option, 201
 - compute node boot option, 201
 - modify boot option, 201
- avoiding out of memory occurrences, 311

B

- backing up and restoring the system data base, 257
- baseboard management controller (BMC), 6
- basic system building blocks, 1
- batch service node, 13
- bittorrent provisioning, 133
- blademon daemon, 89
- boot option
 - compute node, 201
- boot order
 - service nodes, 225

C

- C3 commands, 227
- C4 administrative interface
 - cadmin, 232
- cadmin command, 232
 - set service node boot order, 235
- cascading dual-boot, 136
- cattr command, 252
- changing the dhcp option code number, 86
- changing the global default password, 36
- changing the size of /tmp, 242

- changing the size of per-node swap space, 245
- chassis management control (CMC), 11
- chassis management control (CMC) blade
 - embedded Ethernet switches, 20
 - RJ45 connections, 21
- chassis management controller (CMC) , 6
- cimage command, 206
- cinstallman command, 195
- cluster manager software, 35
- cmcdetected daemon, 79
- cminfo command, 343
- cnodes command, 219
- commands
 - cadmin, 232
 - cattr, 252
 - cimage, 206
 - cinstallman, 195
 - cminfo, 343
 - cnodes, 219
 - configure-cluster, 57
 - console, 238
 - cpasswd, 36
 - cpower, 220
 - crepo, 191
 - dbdump, 340
 - discover, 80
 - discover-rack
 - blademon daemon, 89
 - mysqldump, 259
 - smc-info-gather, 342
- compute node, 12
 - software
 - customizing, 198
 - customizing for additional network interfaces, 201
 - modify compute node image kernel boot options, 201

- services turned off, 190
- compute node software, 187
- Configure backup DNS server, 94
- configure-cluster command, 57
- configuring factory-installed SGI ICE system
 - RHEL, 38
 - SLES, 36
- configuring service nodes, 105
- configuring the service node
 - for DNS, 109
 - for NAT, 106
 - for NFS, 109
 - for NIS for the house network, 109
 - using external DNS for compute node name resolution, 108
- conserv console management package, 238
- conserv console software package, 238
- console management, 238
- cpasswd command, 36
- cpower command, 220
- creating user accounts, 125
- crepo command, 191

D

- database for the system back up and restore procedure, 257
- dbdump command, 340
- DHCP option code, 86
- dhcp options
 - changing, 86
- disabling InfiniBand switch monitoring, 96
- disabling the iSCSI swap device, 245
- discover command, 80
- discover rack command, 89
- discovering compute nodes, 91
- DNS
 - service node configuration, 109

E

- Embedded Support Partner (ESP), 319
 - additional ESP utilities, 325
 - graphical user interface (GUI), 331
 - overview, 320
 - setting up automatic call logging, 322
- enabling the iSCSI swap device, 245

F

- firmware management, 259
 - fwmgrd daemon, 262
 - license requirement, 259
 - terminology, 260

G

- gateway service node, 13
- getting firmware information for all system nodes, 346
- grub boot-loader, 257

H

- HA-RLC
 - administrative resource migration, 179
 - confirm compute node set up, 173
 - confirm initial set up, 173
 - confirm installation, 171
 - create repositories, 165
 - crm_gui utility, 176
 - crm_mon command line utility, 175
 - discover racks, 169
 - HAE, 165
 - HAE documentation, 159
 - Hawk browser interface, 177
 - initial cluster configuration, 168

- monitor compute node set up, 172
- monitor installation, 171
- monitor rack status, 175
- network changes, 163
- replace the HA-RLC instance, 185
- sequential discovery of HA-RLC, 174
- update repositories, 165
- HA-RLC naming, 162
- HAE with HA-RLC , 164, 165
- hardware hierarchy, 6
- hardware overview, 1
- hierarchy of nodes, 6
- High Availability Extension (HAE)
 - documentation, 159
- high availability extension documentation, 151
- high availability extension, virtualization,
 - updates, 151
- home directories on NAS, 116

I

- individual rack unit (IRU), 12
- InfiniBand configuration, 96
 - disabling InfiniBand switch monitoring, 96
- InfiniBand fabric, 26
 - configuration and operation overview, 274
- diagnostic commands
 - ibdiagnet, 289
 - ibnetdiscover, 288
 - ibstat, 285
 - ibstatus, 285
 - perfquery, 287
- management, 265
- management tool graphical user interface (GUI), 266
- routing engine variables, 276
- sgifmcli command, 270
- utilities and diagnostics, 285
- Infiniband network, 33
- initial configuration of a RHEL 6 admin node, 54
- installing SLES11 on the admin node, 43

- installing SMC for SGI ICE patches, 125
- installing software on rack leader controllers, 86
- installing software on service nodes, 86
- interconnect verification tool (IVT) , 19
- introduction, 1
- inventory verification tool (IVT), 314

K

- kdump utility
 - system dump, 345
 - traceback, 345
- keeping time synchronized, 240

L

- local storage for swap and scratch disk space, 248
- login service node , 12

M

- main power, 6
- memory
 - out of memory adjustments, 311
- modify boot option, 201
- modify compute image kernel boot options, 201
- monitoring system metrics with Performance Co-Pilot, 334
- MPI
 - default configuration, 4
- multiroot, 137
- mysqldump command, 259

N

- NAS home directories, 116
- NAT

- configuring the service node, 106
- network interface naming conventions, 27, 32
 - hostnames, 32
 - Infiniband network, 33
 - non-resolvable Names, 31
 - system component names, 28
 - VLAN_1588, 31
 - VLAN_BMC, 30
 - VLAN_GBE, 29
 - VLAN_Head, 28
- network time protocol (NTP), 240
- networks
 - Gigabit Ethernet (GigE) and 10/100 Ethernet connections, 20
 - InfiniBand fabric, 26
 - network interface naming conventions, 27
 - overview, 18
 - virtual local area networks (VLANs), 22
- NFS
 - service node configuration, 109
- NIS
 - service node configuration for the house network, 109
- node replacement procedure, 298
- nodes
 - batch service node, 13
 - compute, 12
 - gateway, 13
 - login service, 12
 - rack leader controller
 - leader node, 10
 - storage service, 13
 - system admin controller
 - admin node, 10

O

- out of memory occurrences, 311
- overview, 1

P

- pdsh and pdcop utilities, 231
- Performance Co-Pilot, 334
- PMIE temperature feature, 338
- power management
 - cpower command, 220
 - IPMI-style commands, 222
 - IRU, rack, and system domains, 223
 - operation on nodes, 222
 - shutting down and booting, 224
 - boot order, 225
- power supply
 - BMC, 6
 - CMC, 6
 - compute blades, 6
 - main power, 6

R

- rack leader controller, 6, 10
 - high availability
 - administrative resource migration, 179
 - compute node boot requirement, 159
 - confirm compute node set up, 173
 - confirm HA-RLC set up, 173
 - confirm installaton, 171
 - create repositories, 165
 - crm_gui utility, 176
 - crm_mon command line utility, 175
 - discover racks, 169
 - documentation, 159
 - HAE overview, 162
 - HAE with HA-RLC, 164
 - hardware requirements, 158
 - Hawk browser interface, 177
 - initial cluster configuration, 168
 - installing admin node, 165
 - monitor compute node set up, 172
 - monitor installation, 171

- monitor rack HAE status, 175
- naming, 162
- network changes, 163
- overview, 158
- physical connections, 159
- replace HA-RLC, 185
- sequential discovery of HA-RLC , 174
- software requirements, 158
- update repositories, 165
- rack leader controller high availability solution
 - See also "rack leader controller high availability", 158
- RAID utility, 254
- redundant management network, 98
- restoring the grub boot-loader , 257

S

- scratch space, 248
- service node boot order, 225
- service node discovery, installation, and configuration, 92
- setting up a NIS Server, 117
- setting up an NFS home server on a service node, 111
 - partitioning, creating, and mounting filesystems, 113
- setting up local storage space for swap and scratch disk space, 248
- setting up serial over LAN connection, 20
- shelf spare replacement, 299
 - booting a replacement system, 306
 - importing the disk volumes, 304
 - installing hardware, 301
- smc-info-gather command, 342
- storage service node, 13
- switch management network, 100
- switching compute nodes to a tmpfs root, 247
- system admin controller
 - high availability
 - SMC licenses, 155

- system admin controller, 6, 10
 - high availability, 147
 - accessing GUI and HA cluster account, 154
 - allocate physical node IP addresses, 154
 - configure SGI helper scripts, 152
 - customization, 155
 - documentation, 151
 - external storage configuration, 150
 - hardware requirements, 148
 - install SGI helper scripts, 152
 - migrating to virtual admin node live, 156
 - network configuration, 151
 - physical node BMC configuration, 150
 - physical requirements, 149
 - software requirements, 149
 - virtual admin guest service, 155
 - virtual admin node resource, 155
 - virtual machine installation, 154
 - installing software, 42
- system component names, 28
- system firmware, 345
 - BIOS version interrogation, 345
 - BMC revision interrogation, 345
 - CMC revision interrogation, 346
 - getting firmware information for all system nodes, 346
 - InfiniBand version interrogation, 346
- system management software, 1
- system monitoring
 - operation, 317
 - overview, 315
 - with Performance Co-Pilot, 334
 - monitoring SDR metrics, 337
- system overview, 1

T

- temperature.pmie feature
 - turning off, 338
- temperature.pmie values

- adjusting, 338
- troubleshooting, 340
 - cminfo, 343
 - dbdump, 340
 - smc-info-gather, 342
- troubleshooting service node configuration for NAT, 107

U

- user accounts
 - creating, 125

V

- viewing the compute node read-write quotas, 252
- virtual local area networks (VLANs), 22
 - VLAN_1588, 22
 - VLAN_BMC, 22
 - VLAN_GBE, 22
 - VLAN_HEAD, 22
- VLAN_1588 network connections, 31
- VLAN_BMC network connections, 30
- VLAN_GBE network connections, 29
- VLAN_Head network connections, 28