SGI® Tempo System Administrator's Guide

007–4993–012

The SGI Tempo systems management software stack 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.

New Features in This Manual

This rewrite of the *SGI Tempo System Administrator's Guide* supports the SGI Tempo systems management software (v2.0).

Major Documentation Changes

Performed the following:

- Added information for the SGI Altix ICE 8400 series systems in "InfiniBand Fabric" on page 22.
- Added a new section called "Redundant Management Network" on page 78.
- Updated the notes section in "Installing Updates on Running Admin, Leader, and Service Nodes " on page 106.
- Added information for the SGI Altix ICE 8400 series systems in "InfiniBand Fabric Overview" on page 177.
- Updated migration information in "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 216.
- Added a new section called "Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode" on page 219.
- Added a new section called "How To Avoid Out of Memory Occurrences on SLES11 and PBS Pro" on page 221.
- Updated "Turning Off the `temperature.pmie` Value" on page 234.
- Added a new section called "Adjusting `temperature.pmie` Values" on page 234.
- Added Appendix A, "Out of Memory Adjustment" on page 247.

Record of Revision

Version	Description
001	July 2007 Original publication.
002	October 2007 Updated to support the SGI Tempo systems management software (v1.1)
003	January 2008 Updated to support the SGI Tempo systems management software (v1.2)
004	May 2008 Updated to support the SGI Tempo systems management software (v1.3)
005	July 2008 Updated to support the SGI Tempo systems management software (v1.4)
006	October 2008 Updated to support the SGI Tempo systems management software (v1.5)
007	January 2009 Updated to support the SGI Tempo systems management software (v1.6)
008	April 2009 Updated to support the SGI Tempo systems management software (v1.7)
009	July 2009 Updated to support the SGI Tempo systems management software (v1.8)

010	October 2009 Updated to support the SGI Tempo systems management software (v1.9)
011	February 2010 Updated to support the SGI Tempo systems management software (v1.10)
012	April 2010 Updated to support the SGI Tempo systems management software (v2.0)

Contents

Figures

Examples

Procedures

About This Guide

This guide is a reference document for people who manage the operation of SGI Altix ICE systems with SGI ProPack 7 for Linux. It describes how to use SGI Tempo systems management software (v2.0) to perform general system discovery, installation, configuration, and operations on SGI Altix ICE systems.

This manual contains the following chapters:

- Chapter 1, "SGI Altix ICE System Overview" on page 1
- Chapter 2, "System Discovery, Installation, and Configuration" on page 31
- Chapter 3, "System Operation" on page 117
- Chapter 4, "System Fabric Management" on page 177
- Chapter 5, "System Maintenance, Monitoring, and Debugging" on page 205

Related Publications

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

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

  This is the hardware user's guide for the SGI Alitx ICE 8200 series systems. It describes the features of the SGI Altix 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 Alitx ICE 8400 series systems. It describes the features of the SGI Altix ICE 8400 series system, as well as, troubleshooting, upgrading, and repairing.

For a list of manuals supporting SGI ProPack for Linux releases covering the following topics, see the *SGI ProPack 7 for Linux Start Here*:

- SGI documentation supporting SGI Altix ICE systems
- Novell documentation for SUSE Linux Enterprise Server 11 (SLES11)
- 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 ProPack 7 for Linux Start Here*, the SGI ProPack 7 release notes, which contain the latest information about software and documentation in this release, the list of RPMs distributed with SGI ProPack 7 can be found in the `/docs` directory on the SGI ProPack 7 CD.

  The SGI ProPack 7 for Linux release notes get installed to the following location on a system running SGI ProPack 7: `/usr/share/doc/sgi-propack-7/README.txt`.
- You can view man pages by typing man *title* on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	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 Altix ICE System Overview

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

Hardware Overview

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

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 Altix 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 Altix ICE 8200 series system. Four IRUs can reside in a custom designed 42U high rack.

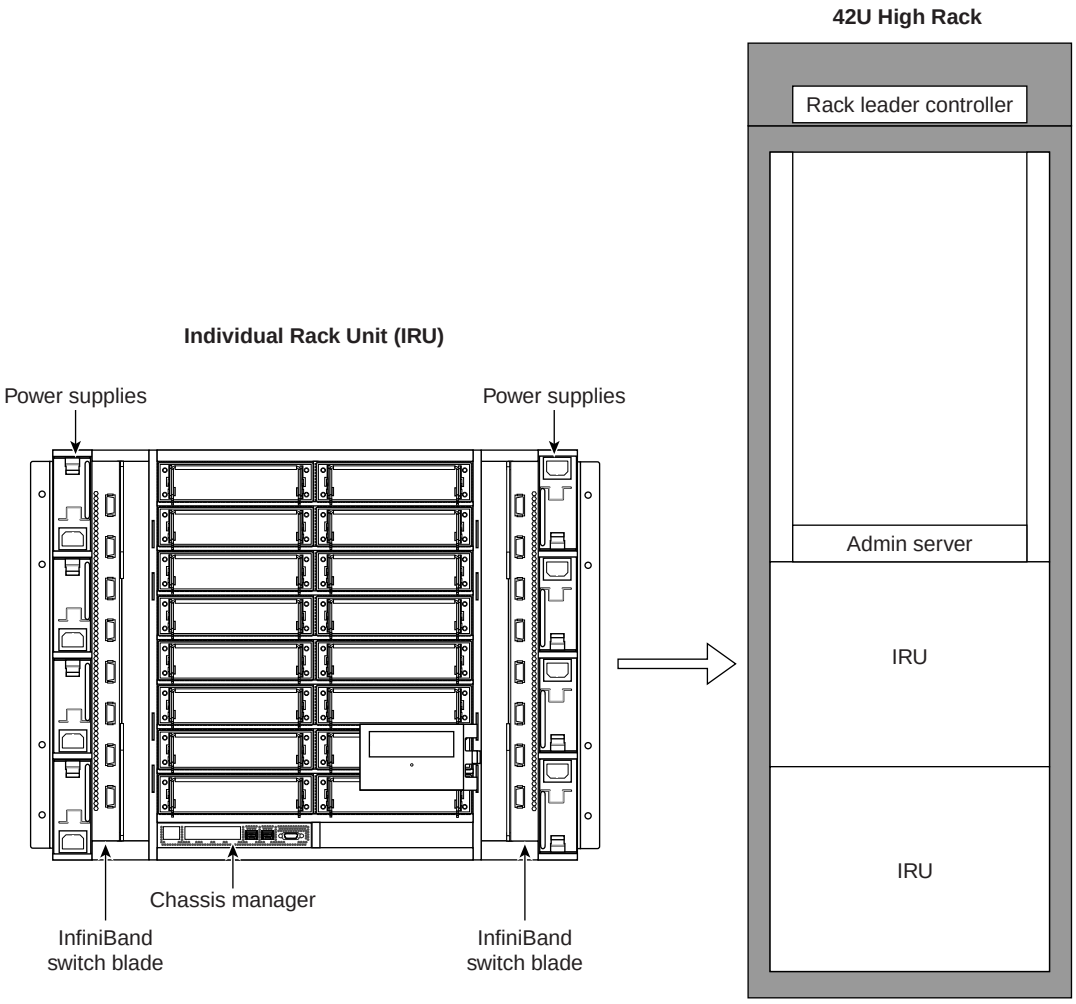

Figure 1-1 Basic System Building Blocks for Altix ICE 8200

The Altix ICE 8400 series of computer systems are also based on an InfiniBand I/O fabric and may be equiped 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.

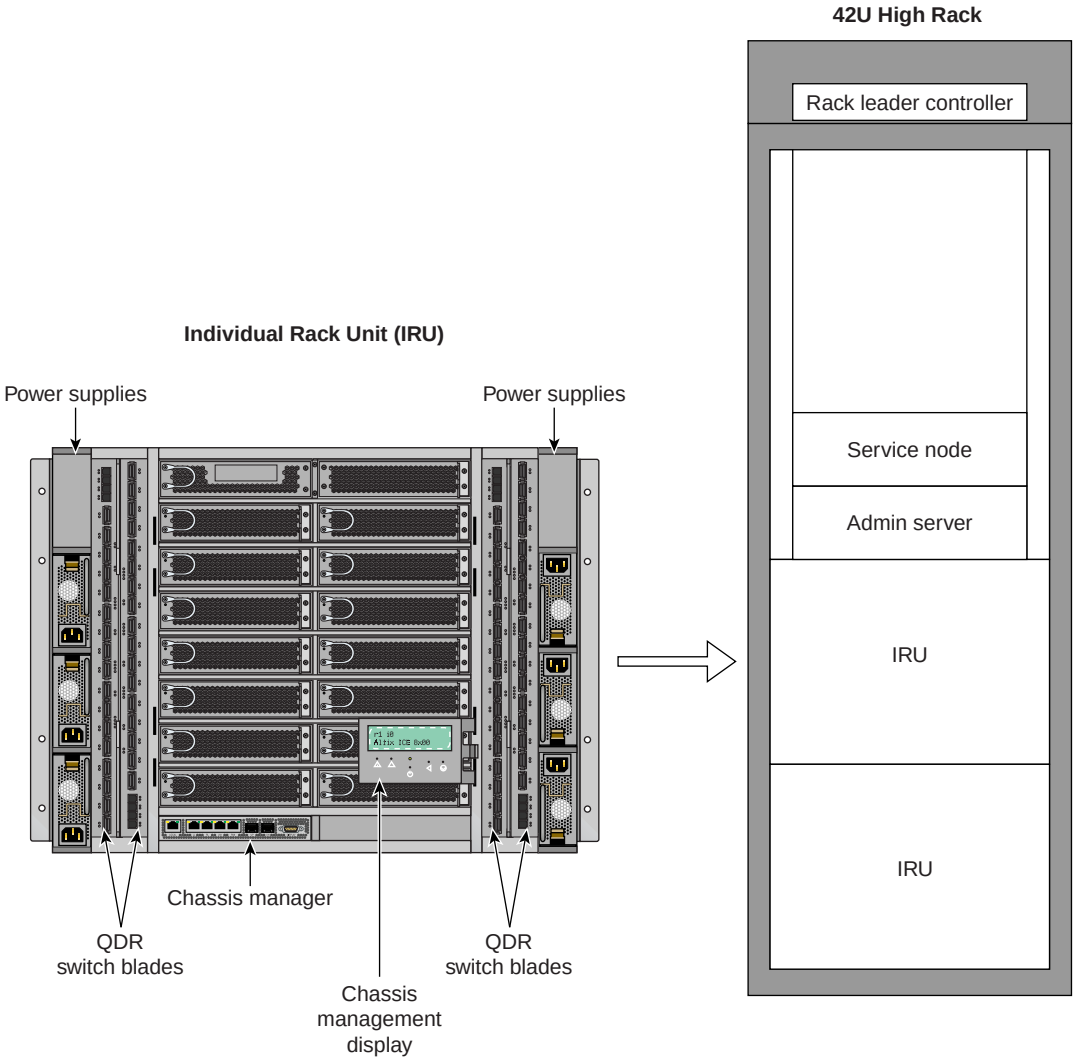

Figure 1-2 Basic System Building Blocks for Altix ICE 8400

For a detailed description of the Altix 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

The SGI Altix 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 Altix 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 athttp://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 177.

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 12 and "NAS Configuration for Multiple IB Interfaces" on page 98). 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 177.

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 18.

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 Altix 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 Altix 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.

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 Altix ICE system. For more detailed information, see "VLANs" on page 18 and "Network Interface Naming Conventions" on page 23.

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 8 shows chassis manager cabling.

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-3 on page 8 and Figure 1-4 on page 13 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the Altix 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 20).

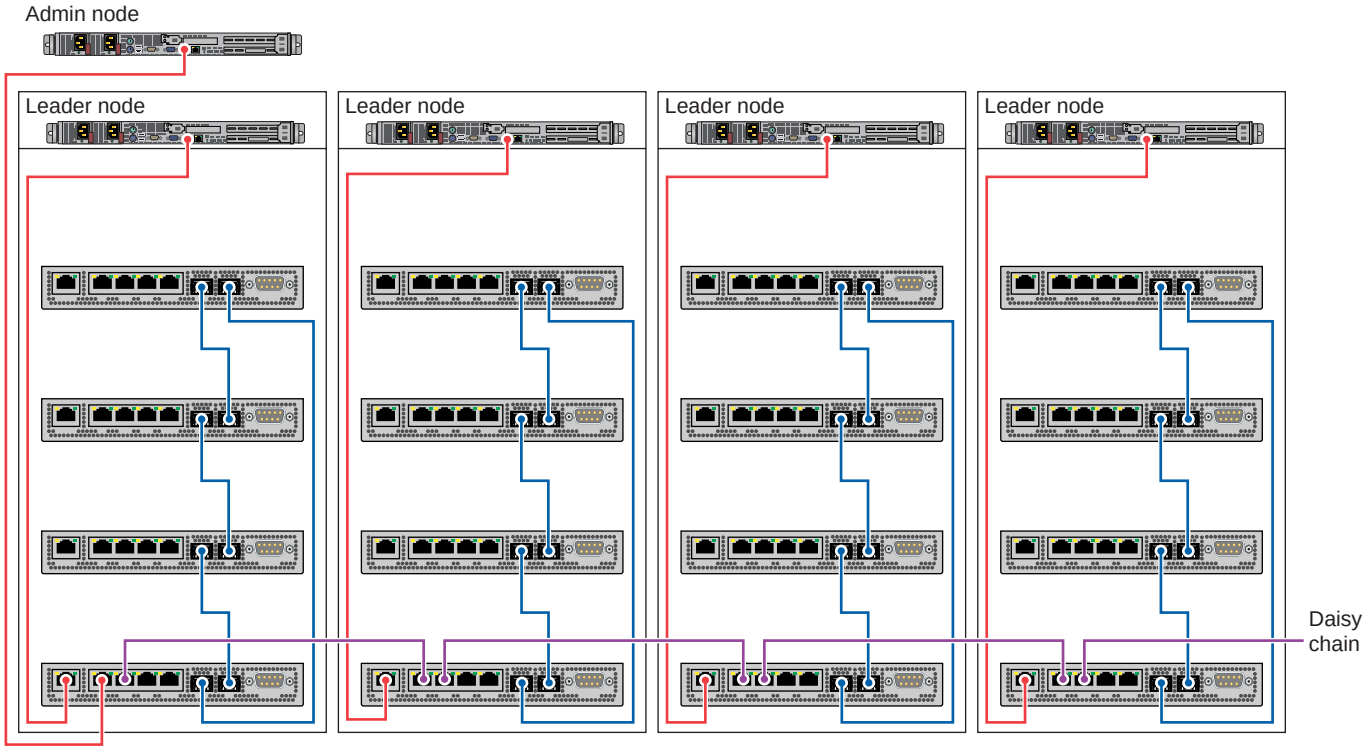

Figure 1-3 Chassis Manager Cabling

Figure 1-4 on page 13 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 Altix ICE system and covers the following topics:

System Admin Controller

The system admin controller (admin node), is used by a system administrator to provision (install) and manage the SGI Altix ICE system using SGI Tempo systems management software. There is only one system admin controller per SGI Altix ICE system, as shown in Figure 1-3 on page 8 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 Tempo 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 8. 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 13 and Figure 1-5 on

page 15. 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 Altix 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 Altix ICE System" on page 124 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 Altix 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 15. The compute nodes are controlled and monitored by the leader node for their rack as shown in Figure 1-3 on page 8. 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 16). For more information on these jacks, see "Gigabit Ethernet (GigE) and 10/100 Ethernet Connections" on page 16.

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 165
- "Enabling or Disabling the Compute Node iSCSI Swap Device" on page 168
- "Changing the Size of Per-node Swap Space" on page 168

Individual Rack Unit

The individual rack unit (IRU) is one of the basic building blocks of the SGI Altix 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 Altix ICE system via the InfiniBand fabric and GigE to the public customer network as shown in Figure 1-5 on page 15. 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 Altix 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 Altix ICE system. There can be multiple storage service nodes for larger Altix ICE system configurations. Figure 1-4 on page 13 shows a service node and a storage service node (NAS cube).

For smaller Altix 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 (IB), and requires that Internet Protocol (IP) over IB be properly configured on the system to allow the Altix ICE nodes to be attach to the XE250 provided storage.

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-3 on page 8 and Figure 1-4 on page 13 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the Altix ICE hierarchical design. They are not meant as cabling diagrams.

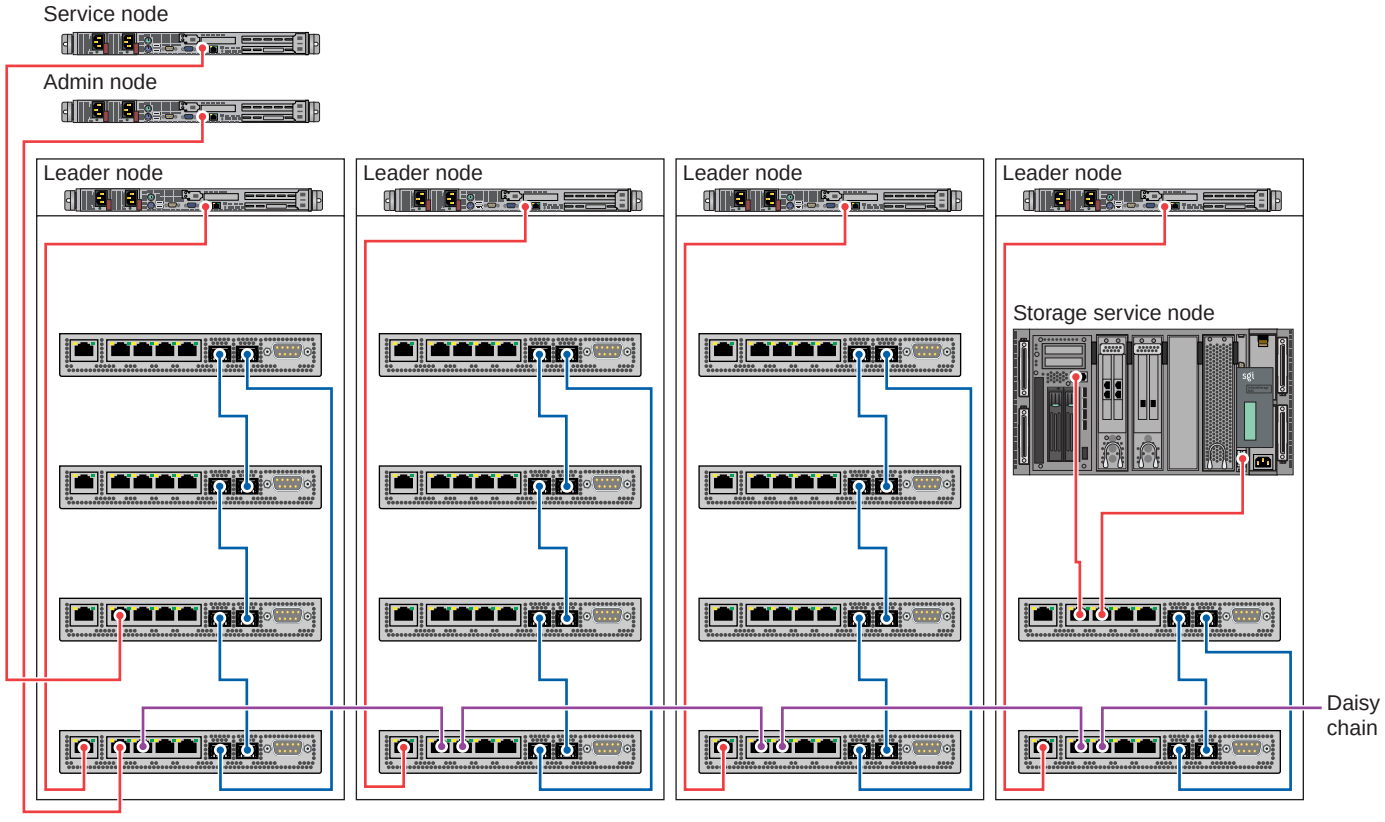

Figure 1-4 Service Nodes

Networks

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

Networks Overview

This section describes the various network connections in the SGI Altix 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 15. A single service node can provide both login and batch services.

System administrators provision (install software) and manage the Altix ICE system via the logical VLAN network running over the GigE connection (see Figure 1-7 on page 19, Figure 1-8 on page 20, and Figure 1-9 on page 21. 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 Altix ICE system (see Figure 1-5 on page 15).

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 pdcp Utilities" on page 157) that enables the execution of any standard command on all Altix ICE system nodes.

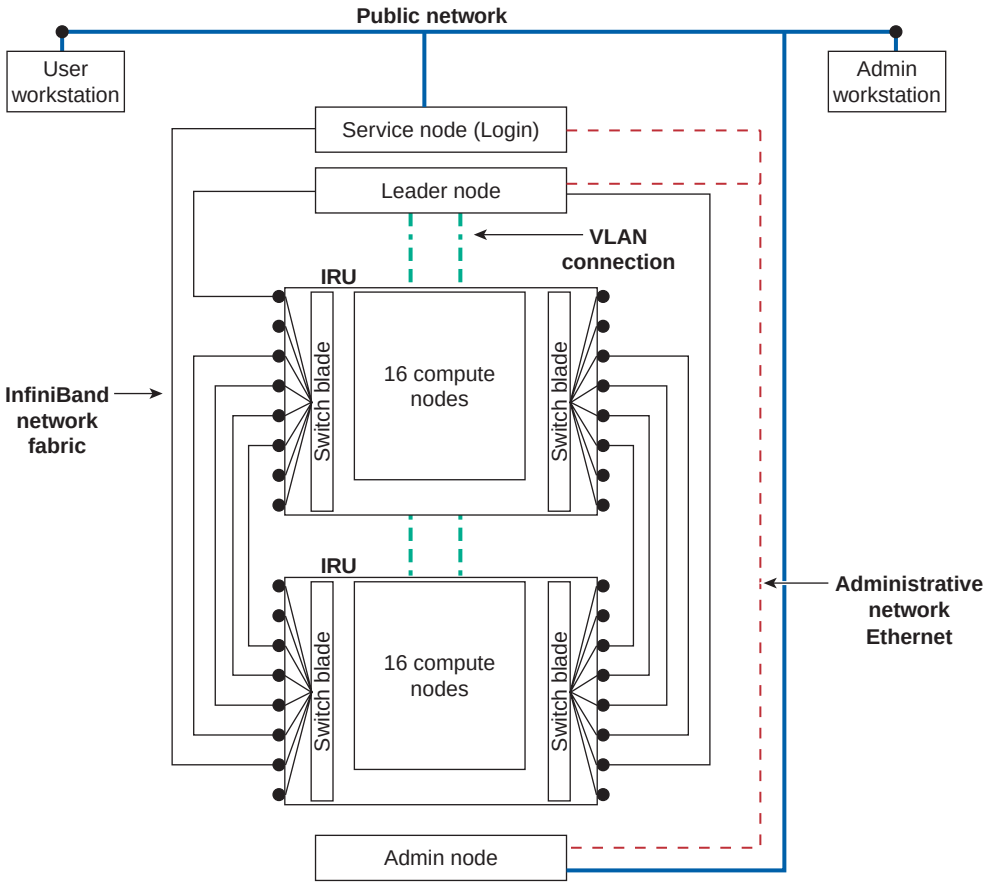

Figure 1-5 Network Connections In a System With Two IRUs

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 223.

Gigabit Ethernet (GigE) and 10/100 Ethernet Connections

The SGI Altix 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 16 shows a more detailed view of the Chassis manager.

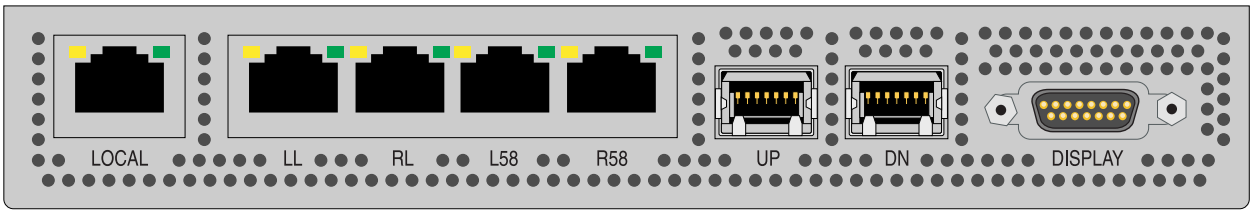

Figure 1-6 Chassis Manager

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 16. 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 16. 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 8.

- **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 8 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 8 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 Altix ICE system as shown in Figure 1-4 on page 13.

For small, one IRU configurations, the **L58** and **R58** ports (see Figure 1-6 on page 16) 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 19.

- `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 19.

`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 Software on the System Admin Controller" on page 34. 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.

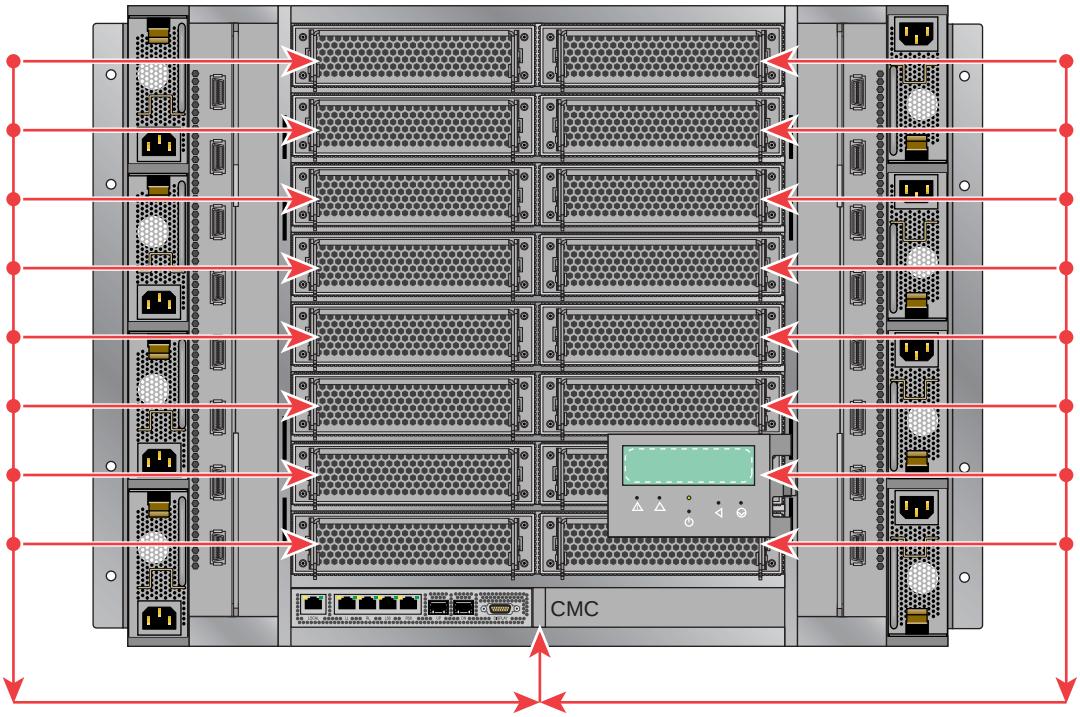

Figure 1-7 VLAN_GBE and VLAN_BMC Network Connections - IRU View

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).

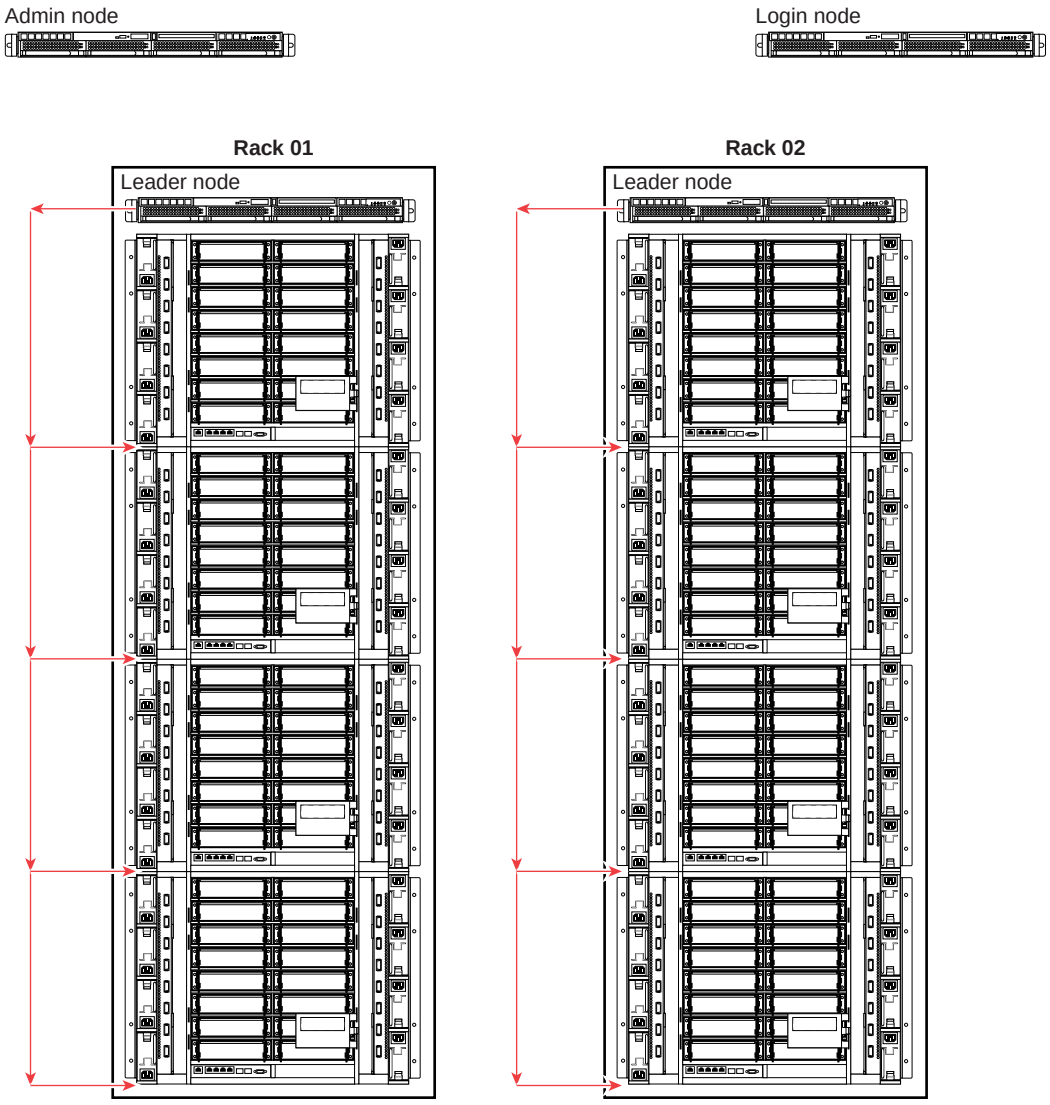

Figure 1-8 VLAN_GBE and VLAN_BMC Network Connections – Rack View

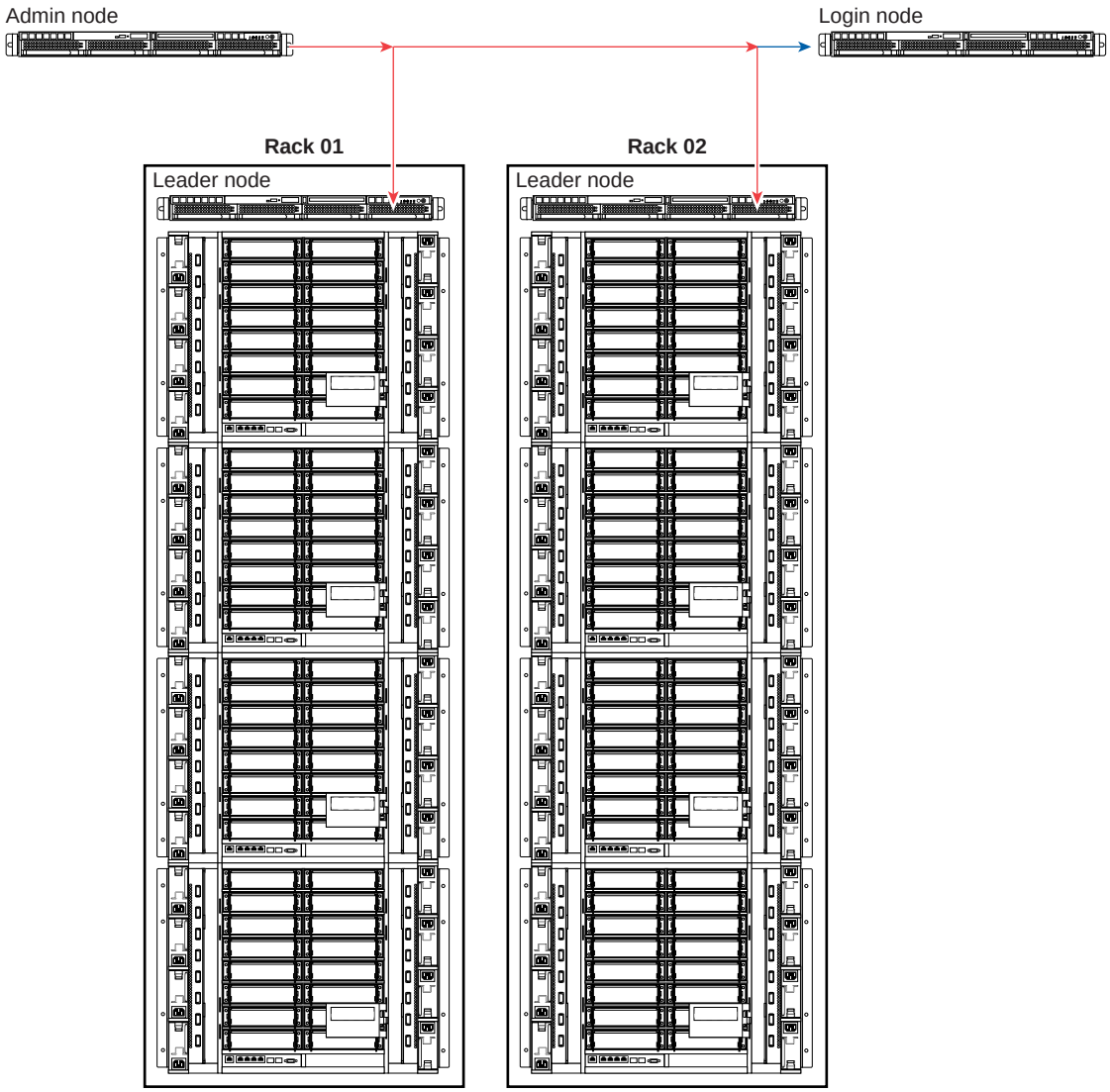

Figure 1-9 VLAN_HEAD Network Connections

In an SGI Altix 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 19, 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 Altix 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 Altix 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 23).

Each IRU has internal InfiniBand switches which interconnect a fabric within the IRU (see Switch blade in Figure 1-10 on page 23). 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 23).

On SGI Altix 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.

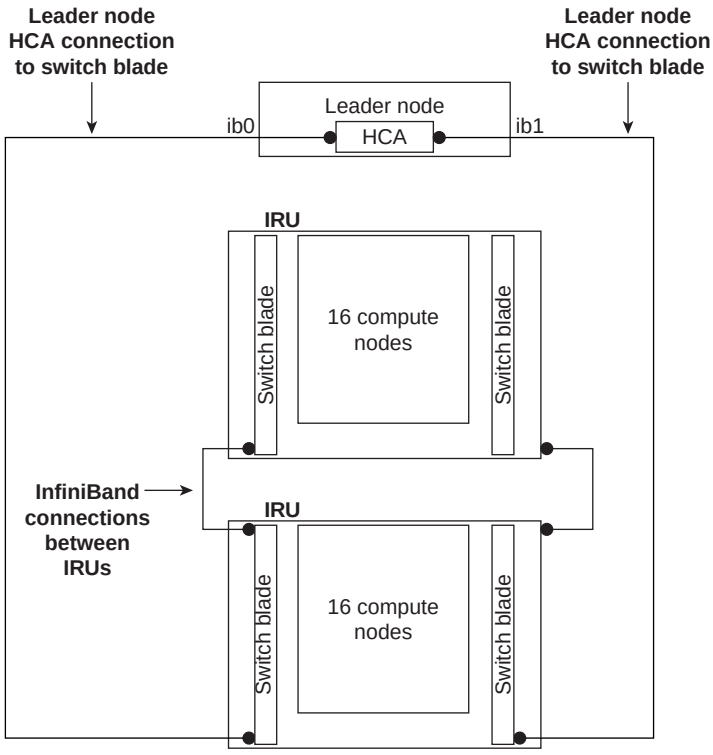

Figure 1-10 Two InfiniBand Fabrics in a System with Two IRUs

Network Interface Naming Conventions

As described in "Networks" on page 14, you can think of an SGI Altix 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 18, 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 24
- "`VLAN_Head` Network Connections" on page 24

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 Altix 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 Altix ICE system. These IDs are partly not routable. See "Non-resolvable Names" on page 27.
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 25 shows the VLAN_Head network connection names. See Figure 1-9 on page 21.

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 71.

VLAN_GBE Network Connections

Table 1-2 on page 25 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 19). 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 26 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 19).

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 Altix 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 27 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 Altix 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 Altix ICE system. Table 1-5 on page 27 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 Altix 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 28. 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`: SGI Tempo Administrative Interface" on page 158).

For the SGI Tempo v1.4 release and later, 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 SGI Tempo node name, there will be `CNAME` alias for the
SGI Tempo 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 29 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.

For the SGI Tempo 1.4 release, 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 28).

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 Tempo systems management software to discovery, install, and configure your Altix ICE system and covers the following topics:

Note: If you are upgrading from a prior release or installing SGI Tempo software patches, see "Installing SGI Tempo Patches and Updating SGI Altix ICE Systems " on page 101 and "Upgrading from Prior SGI ProPack Releases to SGI ProPack 7 " on page 109.

Configuring Factory-installed SGI Altix 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 Altix ICE System

To configure the pre-installed software that comes on the 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
   - Root password
   - Time zone
2. If you need to adjust SGI Altix 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 44).
   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 66.

Overview of Installing Software and Configuring Your SGI Altix ICE System

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

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

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

1. Follow "Installing SLES11 on the Admin Node" on page 34 to install software on your system admin controller (admin node).
2. Follow "`configure-cluster` Command Cluster Configuration Tool" on page 44 to configure the overall cluster.
3. Follow "Installing Software on the Rack Leader Controllers and Service Nodes" on page 71 to install software on the leader nodes and service nodes.
4. Follow "Discovering Compute Nodes" on page 74 to discover compute nodes.
5. Follow "Service Node Discovery, Installation and Configuration" on page 75 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 76 to configure and check the status of the InfiniBand fabric.
7. Follow "Configuring the Service Node" on page 81, "Setting Up an NFS Home Server on a Service Node for Your Altix ICE System" on page 86, and "Setting Up a NIS Server for Your Altix ICE System" on page 92 to complete your system setup.

Installing Software on the System Admin Controller

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 Altix ICE system. The SGI Admin Node Autoinstallation DVD contains a software image for the system admin controller (admin node) and contains SGI Tempo and SGI ProPack for Linux packages, used in conjunction with the packages from the SLES11 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 223.

This section covers the following topics:

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 2-3 Installing 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 35.

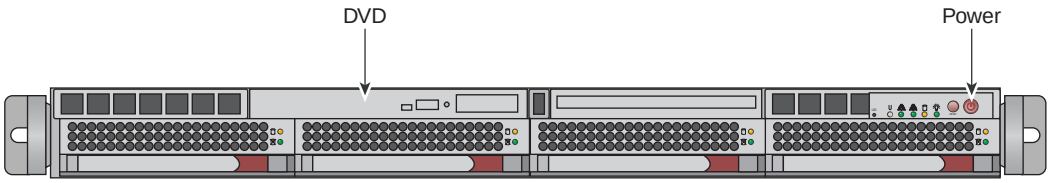

Figure 2-1 System Admin Controller Power On Button and DVD Drive

Prior to the SGI Tempo 1.2 release, 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=ttyS1,38400n8
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 35.

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 109 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 109) 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 37. Click on the **Next** button to continue.

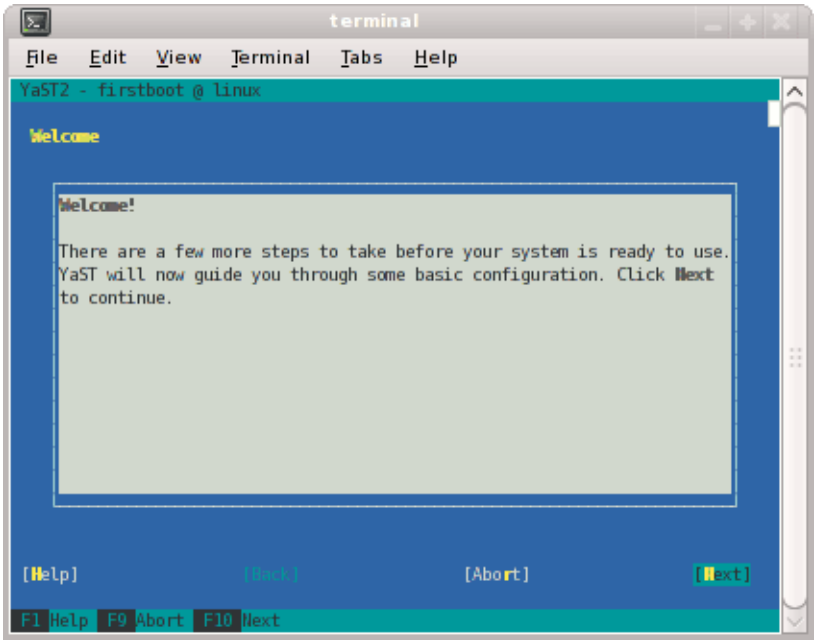

Figure 2-2 YaST2 - firstboot@Linux Welcome Screen

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 38, 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.

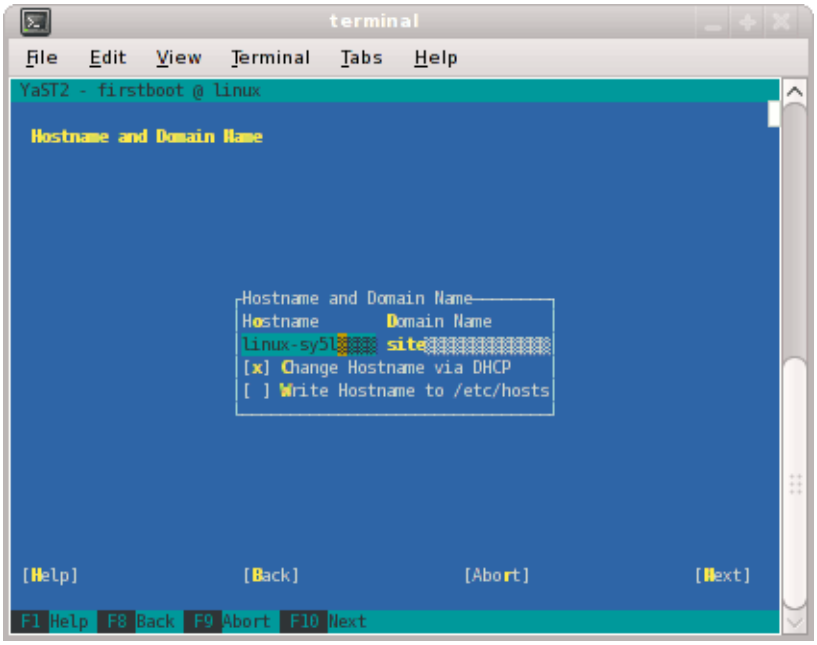

Figure 2-3 Hostname and Domain Name Screen

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 39. Select **Change** and a small window pops up that lets you choose **Network Interfaces...** or **Reset to Defaults**. Choose **Network Interfaces**.

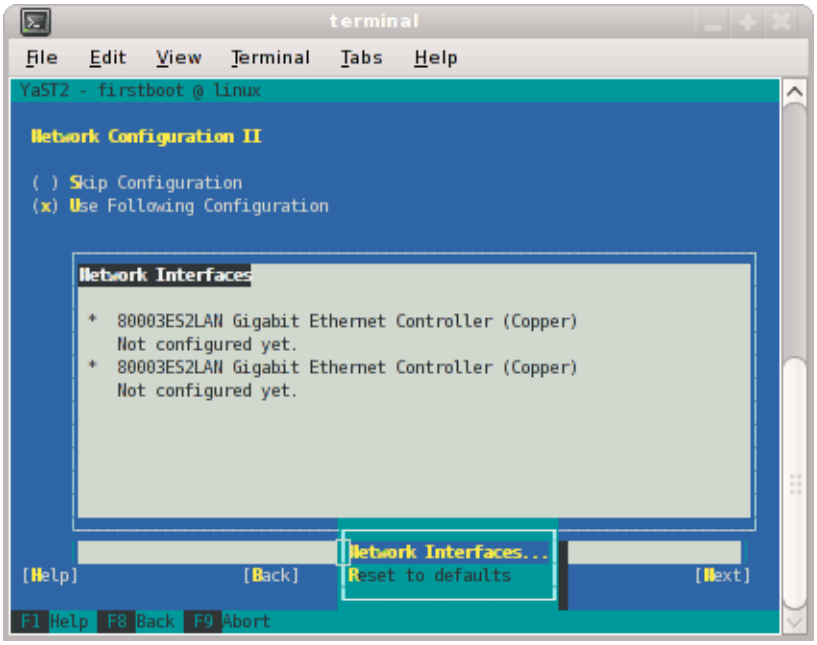

Figure 2-4 Network Configuration II Screen

8. From the **Network Settings** screen, as shown in Figure 2-5 on page 40, configure the first card under **Name** to establish the public network (sometimes called the house network) connection to your SGI Altix ICE system. To do this, highlight the first card and select **Edit**.

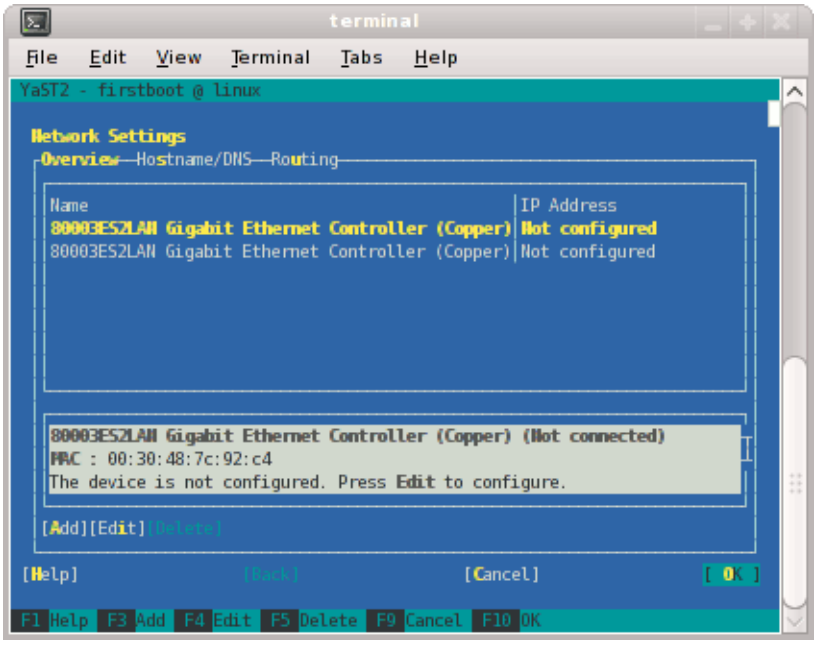

Figure 2-5 Network Settings Screen

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 40.

---

9. The **Network Card Setup** screen appears, as shown in Figure 2-6 on page 41. 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, *mysystem*-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.

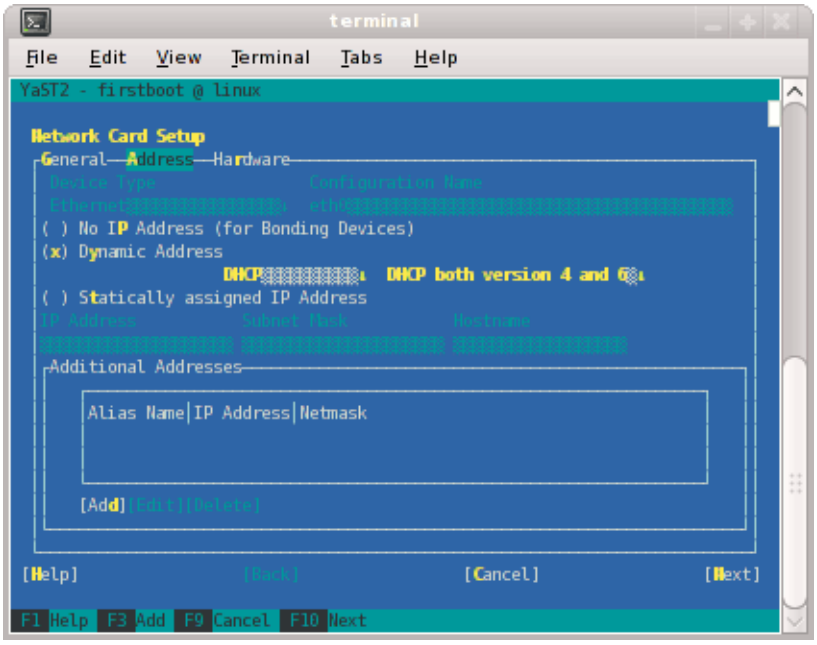

Figure 2-6 Network Card Setup Screen

10. At this point, you are back at the **Network Settings** screen as shown in Figure 2-7 on page 42. 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.

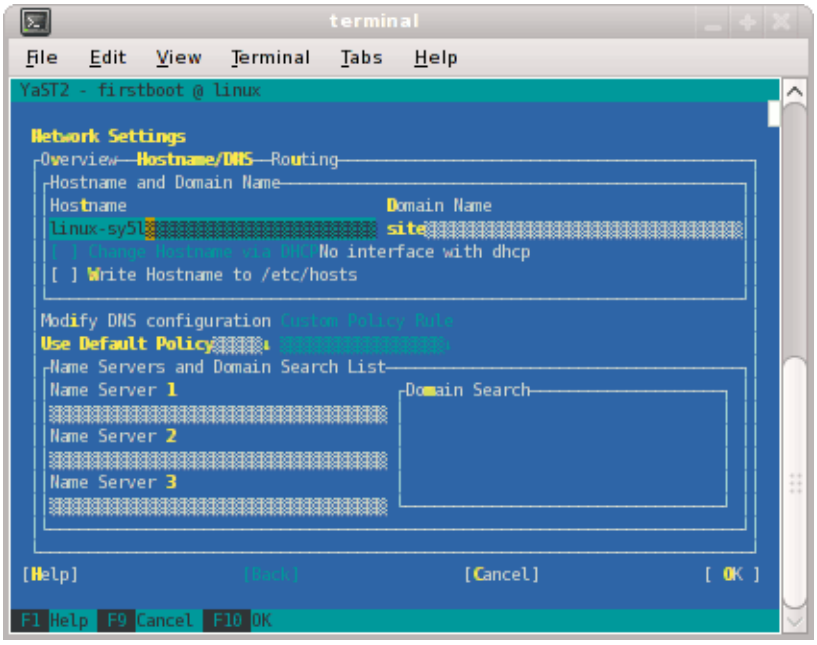

Figure 2-7 Network Settings Screen

11. Select **Routing** shown in Figure 2-8 on page 43 and enter your house/public network default router information there. Now you can select **OK**.

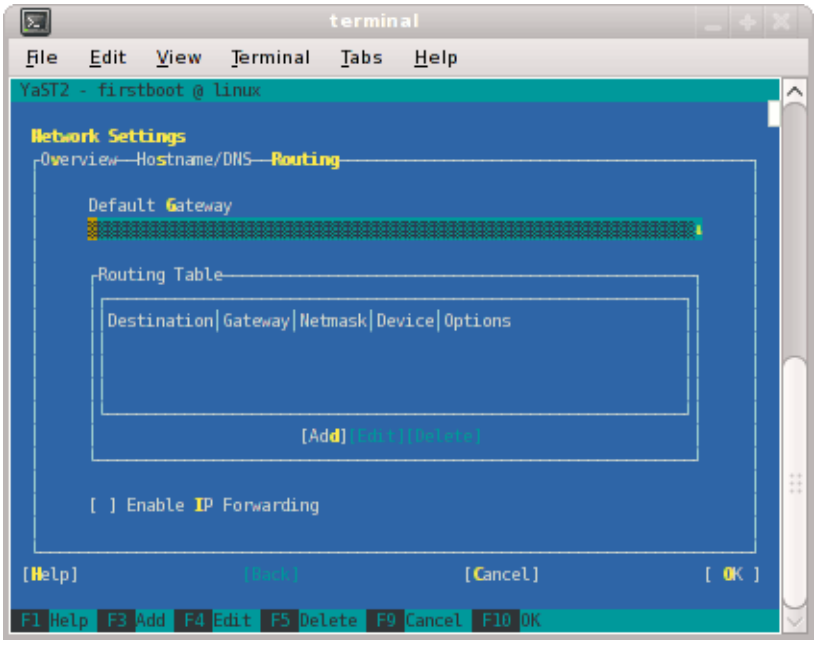

Figure 2-8 Network Settings Routing Screen

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 44.

`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 45 on a previously configured machine. For information on initial configuration, see Procedure 2-4.

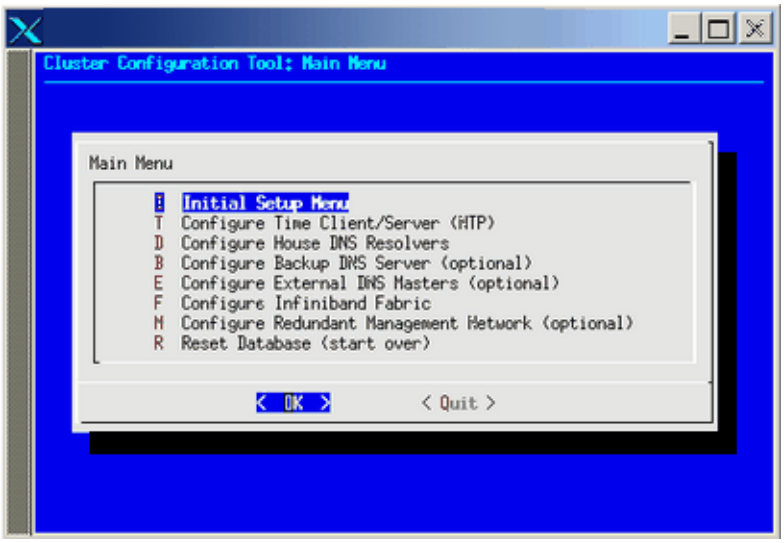

Figure 2-9 Cluster Configuration Tool: Main Menu Screen

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 and optionally SGI ProPack 7. 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 InfiniBand network (see "InfiniBand Configuration" on page 76)

Information on using this tool is described in the procedure in the following section, see "Installing Software on the System Admin Controller" on page 34.

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

Procedure 2-4 Using the Cluster Configuration Tool to Configure Your System Admin Controller

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 46. This tool provides instructions on the steps you need to take to configure your cluster. Click **OK** to continue.

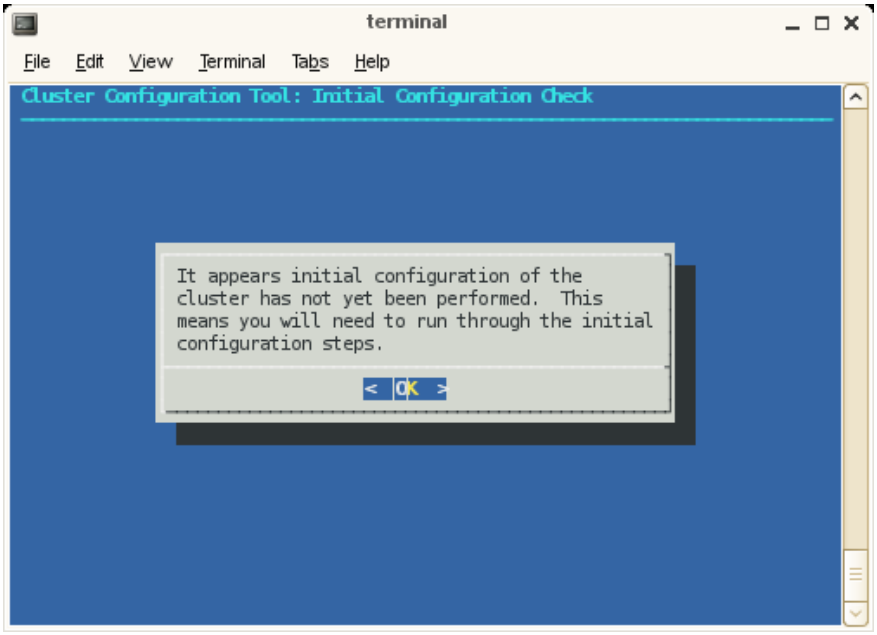

Figure 2-10 Cluster Configuration Tool: Initial Configuration Check Screen

3. The **Cluster Configuration Tool: Initial Cluster Setup** screen appears, as shown in Figure 2-11 on page 47. Read the notice and then click **OK** to continue.

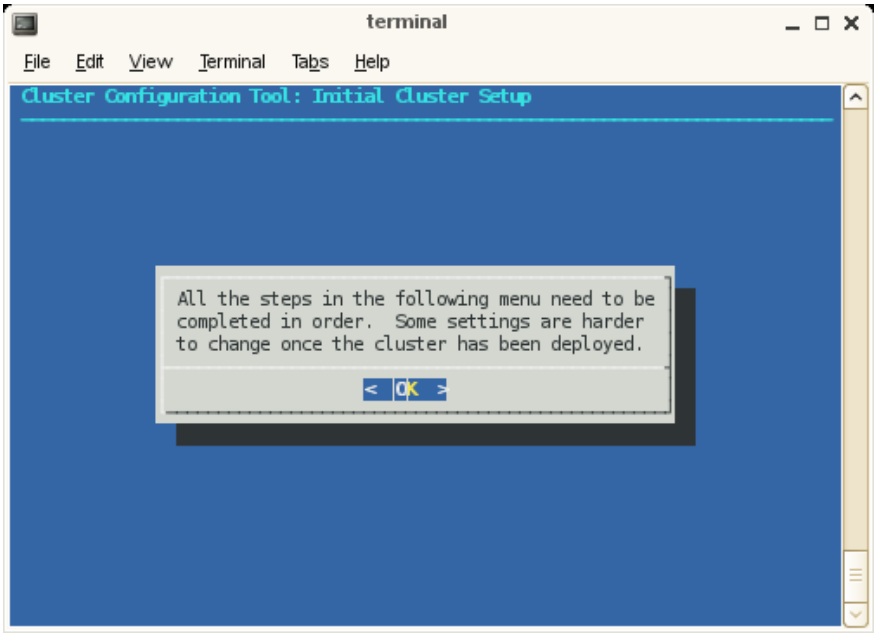

Figure 2-11 Cluster Configuration Tool: Initial Cluster Setup Screen

---

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**.

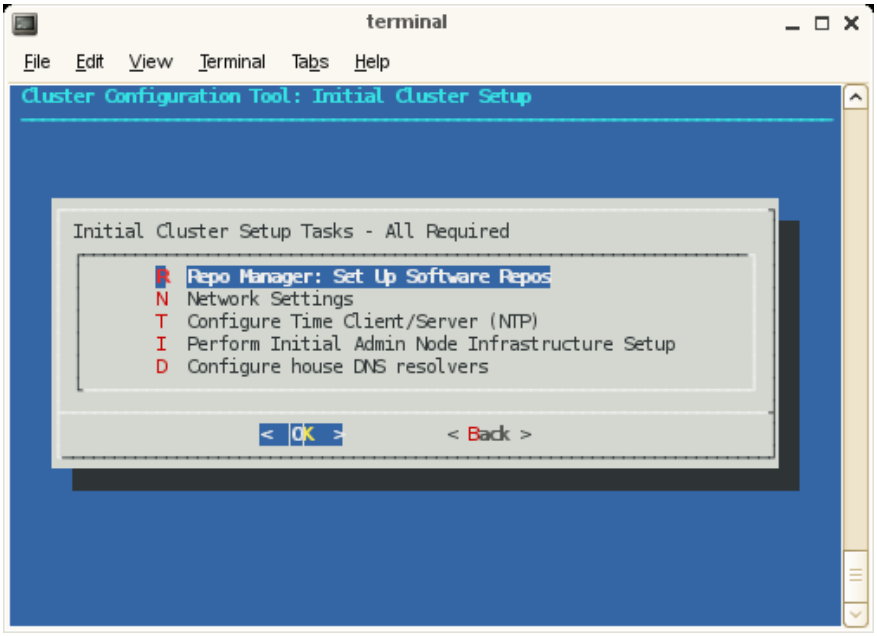

Figure 2-12 Initial Cluster Setup Tasks Screen

5.

Note: The next four screens use the `crepo` command to set up software repositories, such as, SGI Foundation, SGI Tempo, SGI ProPack, or SLES11. For more information, see "`crepo` Command" on page 119.

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

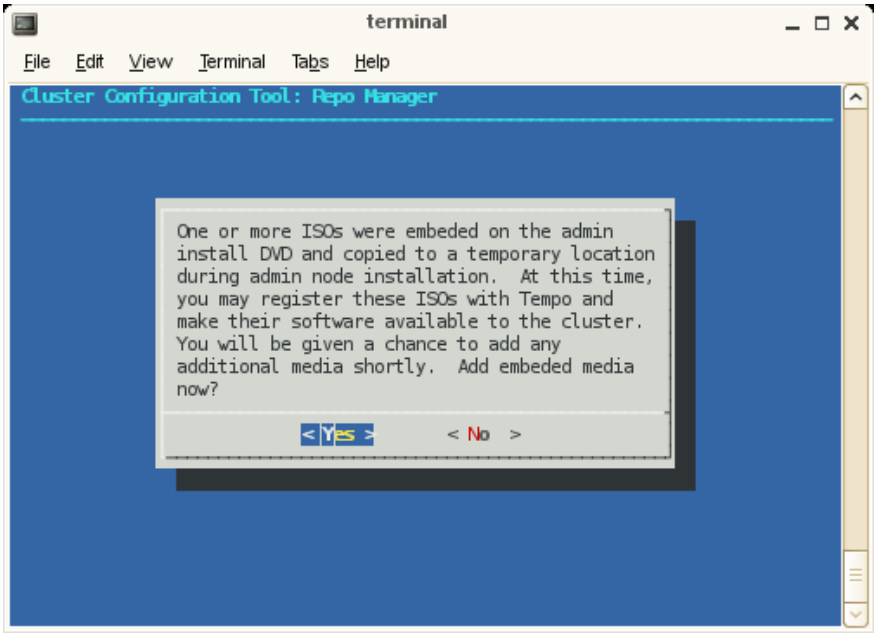

Figure 2-13 Cluster Configuration Tool: Repo Manager Screen One

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

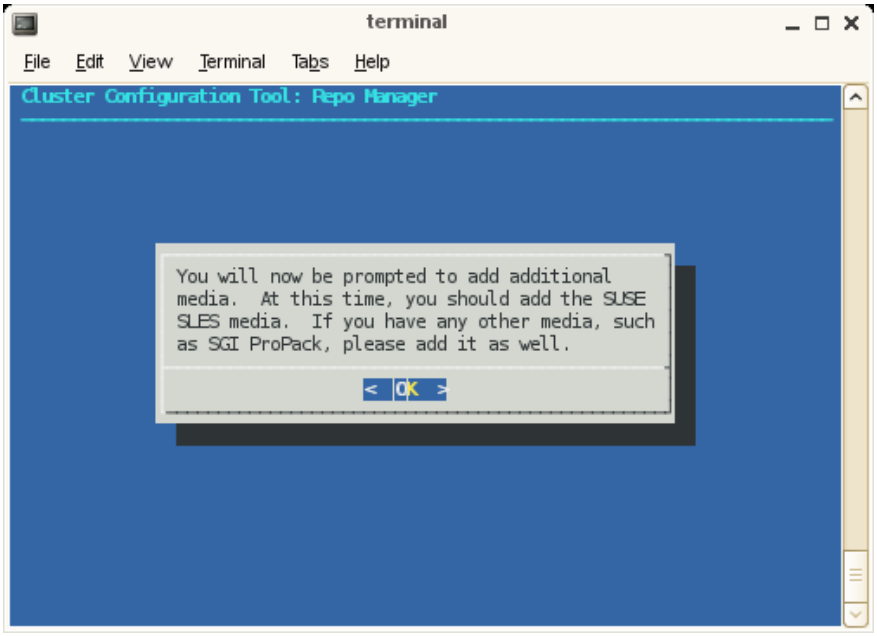

Figure 2-14 Cluster Configuration Tool: Repo Manager Screen Two

7. To register additional media with SGI Tempo, click **Yes**.

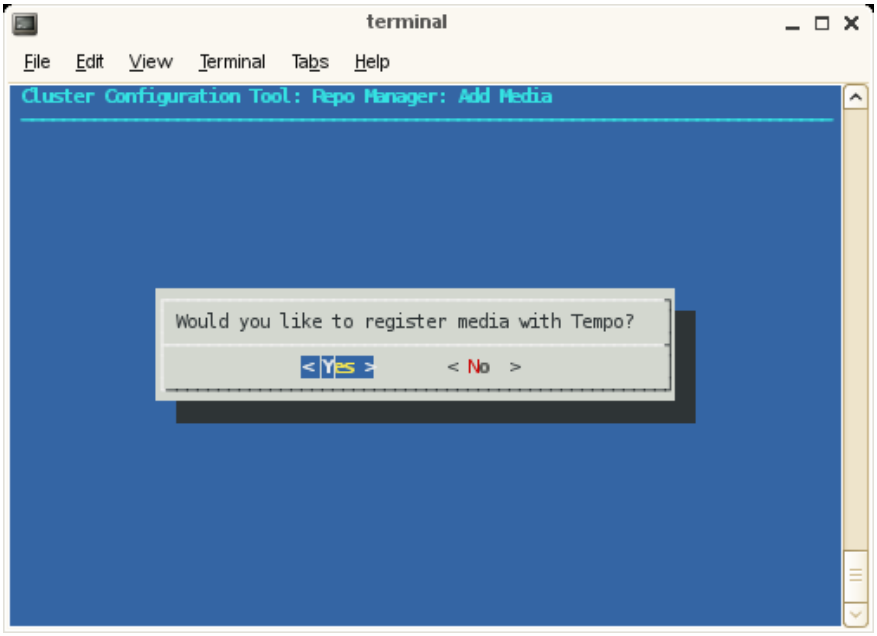

Figure 2-15 Cluster Configuration Tool: Repo Manager Screen Three

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.

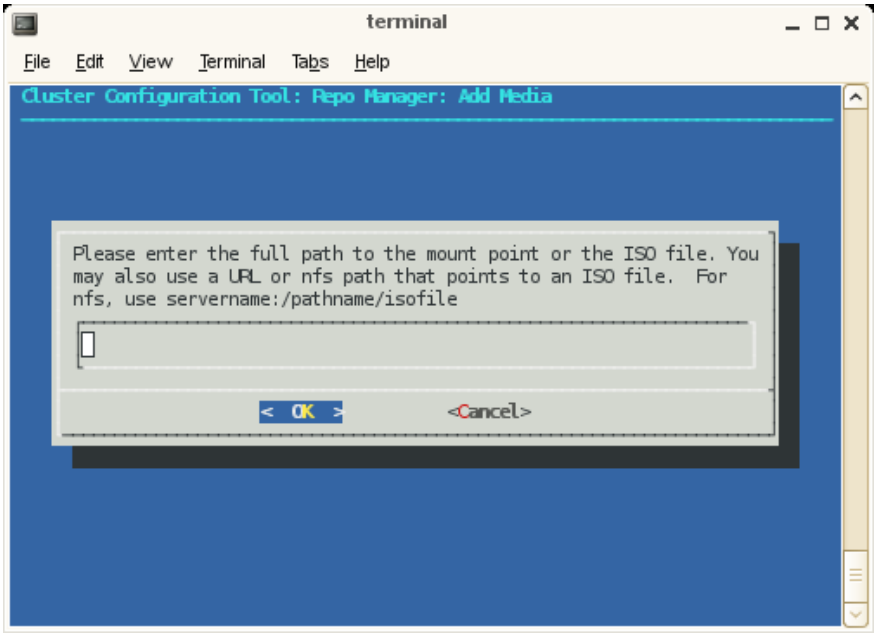

Figure 2-16 Cluster Configuration Tool: Repo Manager Screen Four

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

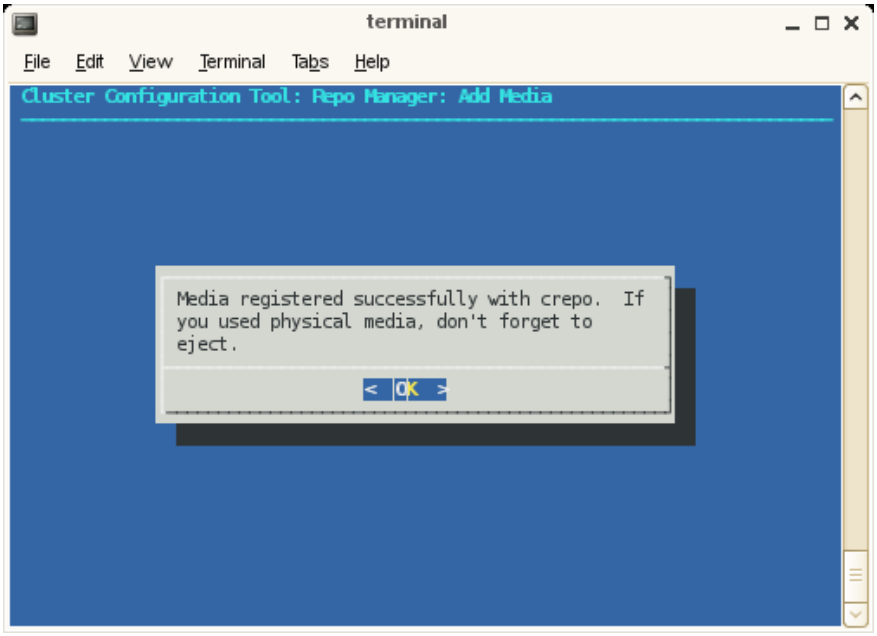

Figure 2-17 Cluster Configuration Tool: Repo Manager: Add Media Screen Four

---

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 **Inital 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 54.

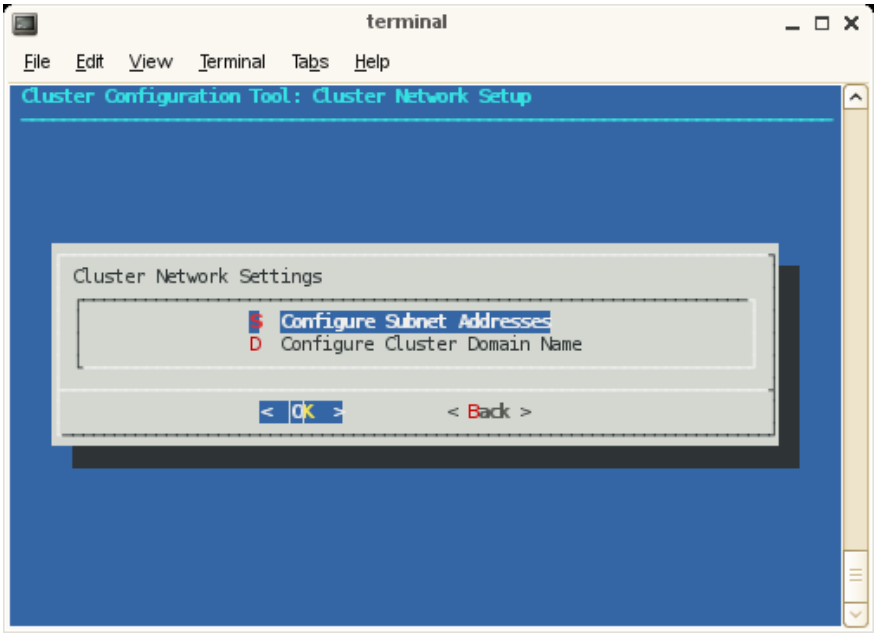

Figure 2-18 Cluster Network Setup Screen

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 55.

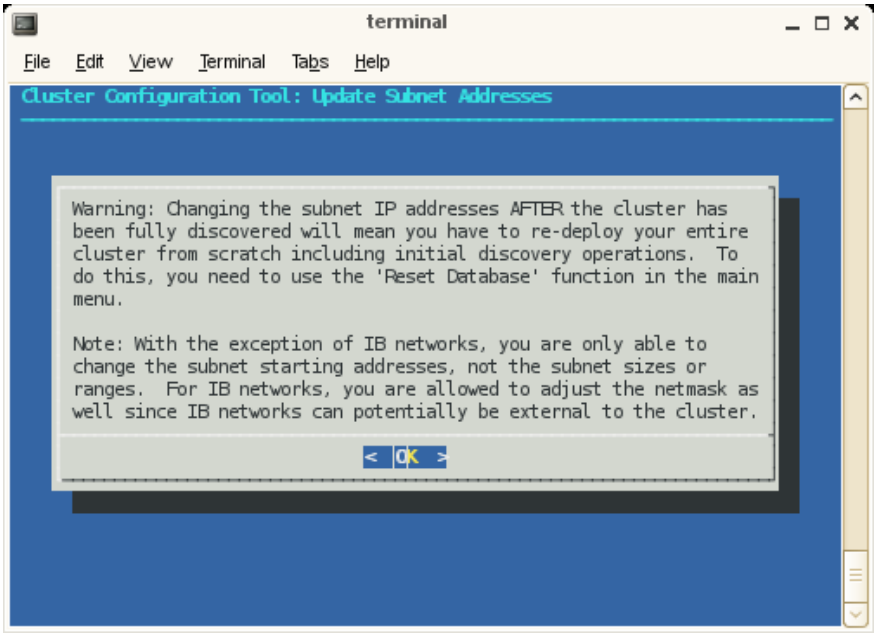

Figure 2-19 Update Subnet Address Warning Screen

Once you deploy your Altix 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 56.

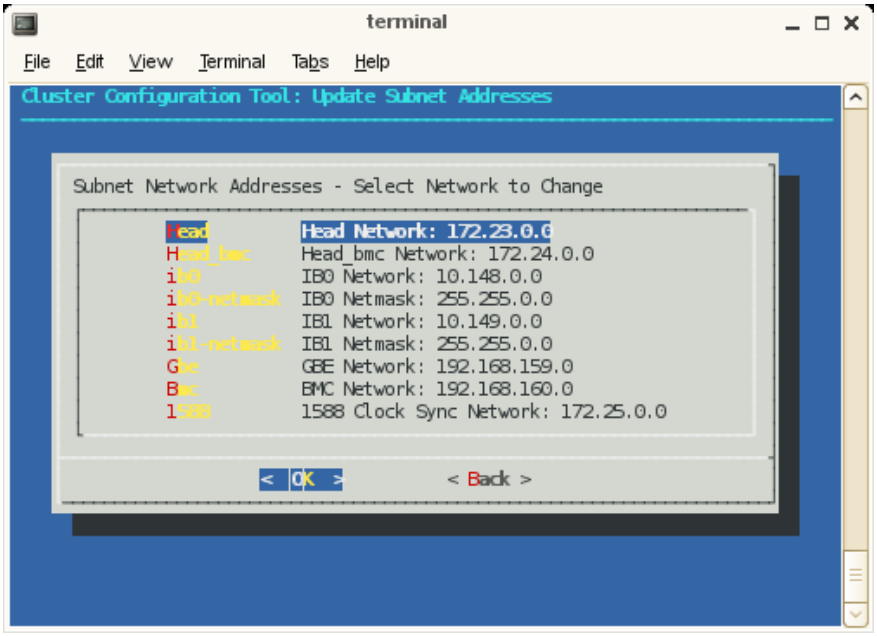

Figure 2-20 Update Subnet Addresses Screen

The default IP address of the system admin controller which is the **Head Network** for the Altix 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 Altix ICE system, as shown in Figure 2-21 on page 57. Click **OK** to continue (this will be a subdomain to your house network, by default).

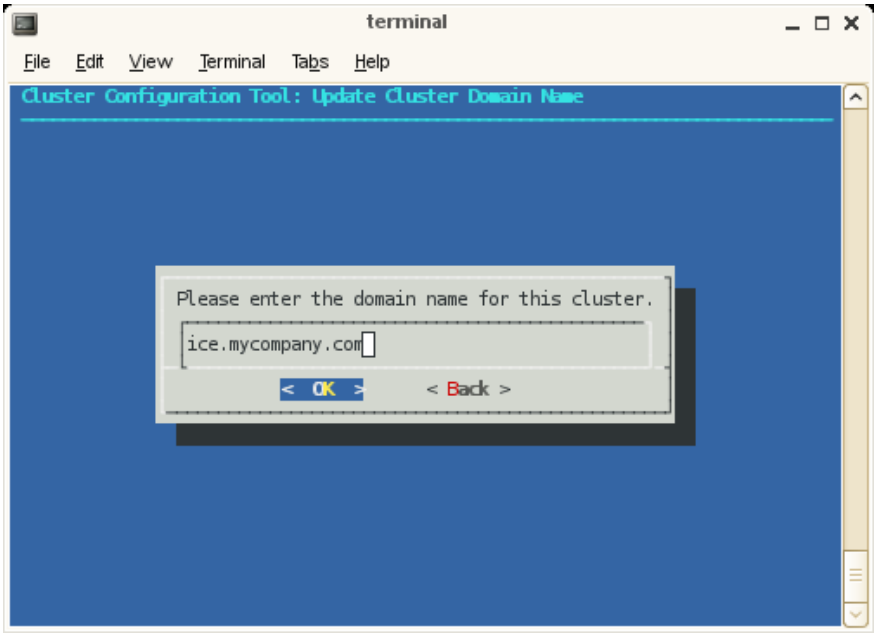

Figure 2-21 Update Cluster Domain Name Screen

13. The next operation in the **Initial Cluster Setup** menu is **NTP Time Server/Client Setup**. This procedure changes your NTP configuration file. Click on **OK** to continue. This sets the system admin controller to serve time to the Altix ICE system and allows you to add time servers on your house networks, which you may optionally use.

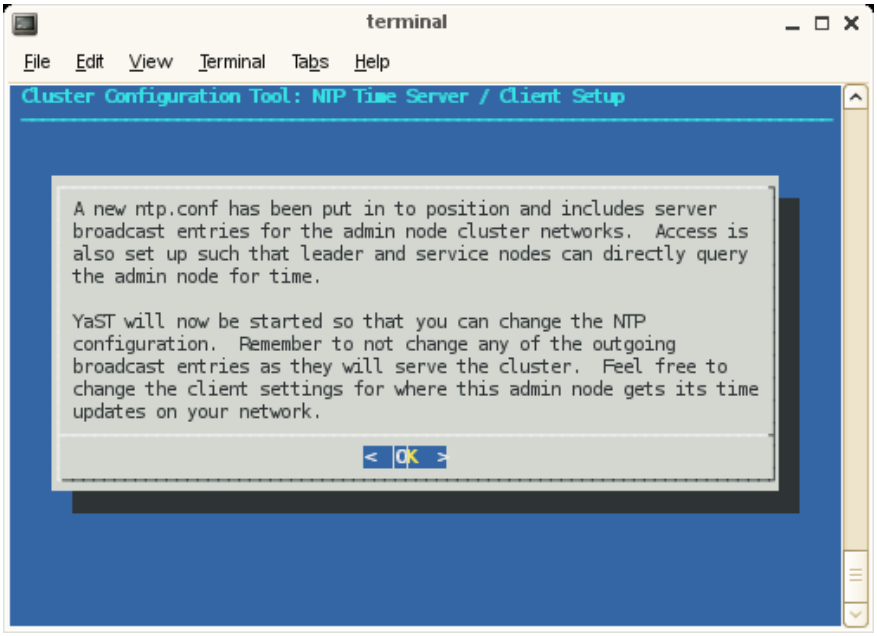

Figure 2-22 NTP Time Server/Client Setup Screen

14. Configure NTP time service as shown in Figure 2-23 on page 59. Click **Next** to continue.

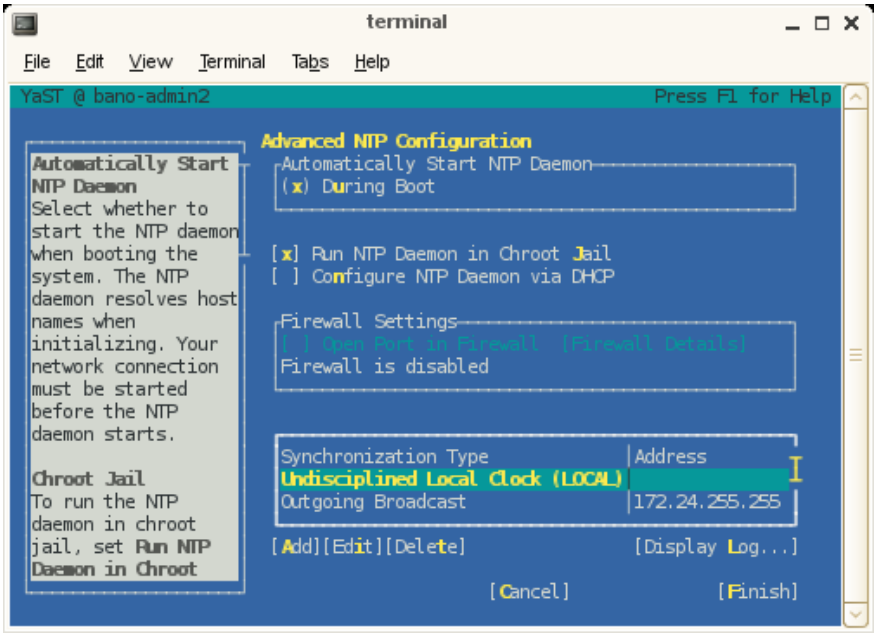

Figure 2-23 Advanced NTP Configuration Screen

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

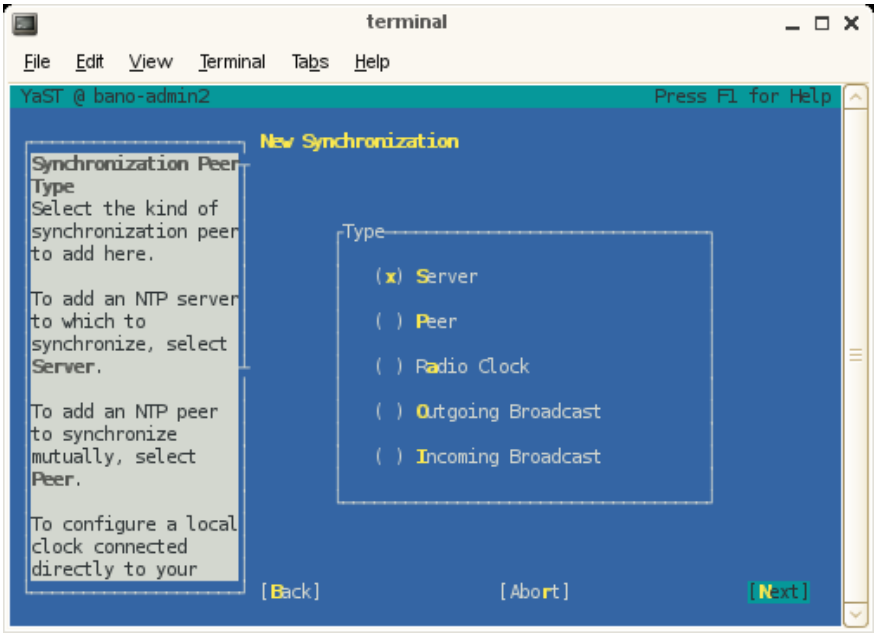

Figure 2-24 New Synchronization Screen

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

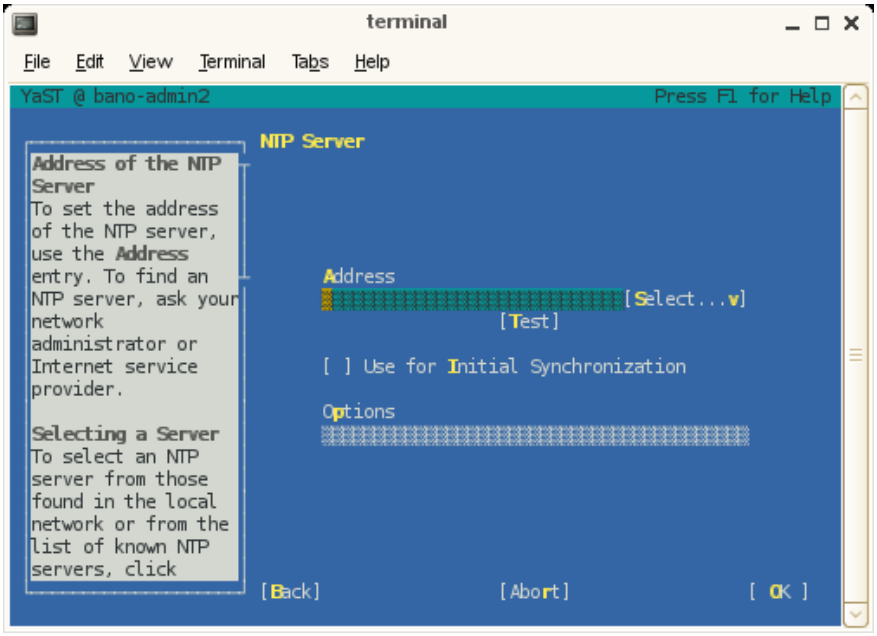

Figure 2-25 NTP Server Screen

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

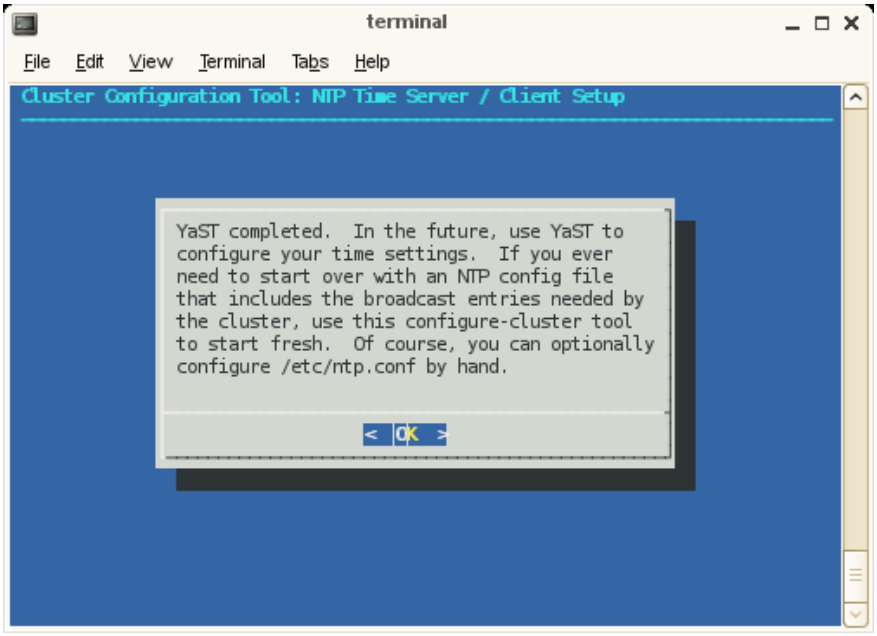

Figure 2-26 NTP Time Server/ Client Setup Screen Three

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 Altix ICE system.

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

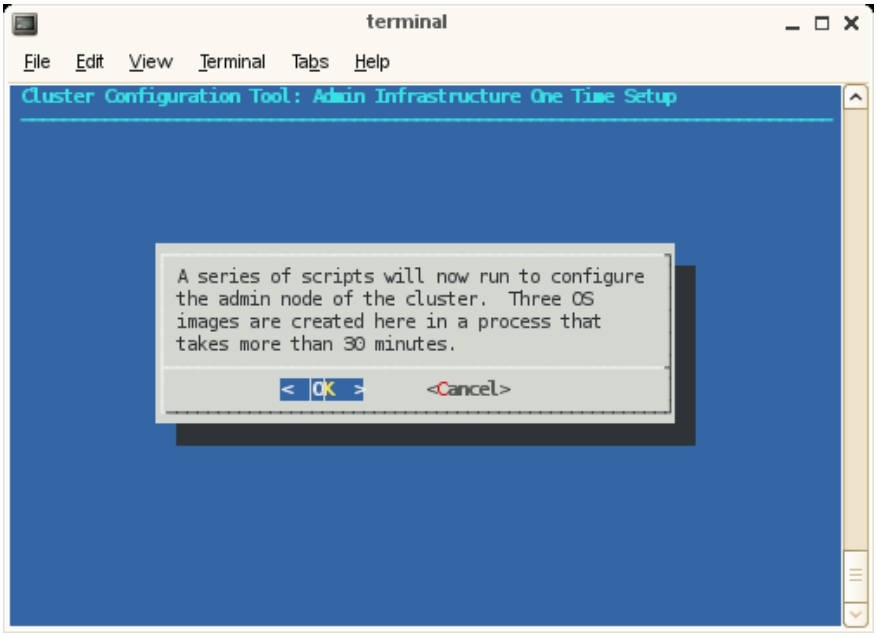

Figure 2-27 Admin Infrastructure One Time Setup Screen One

The root images for the service, rack leader controller, and compute nodes are then created. The output of the `mksiimage` 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 75. 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).

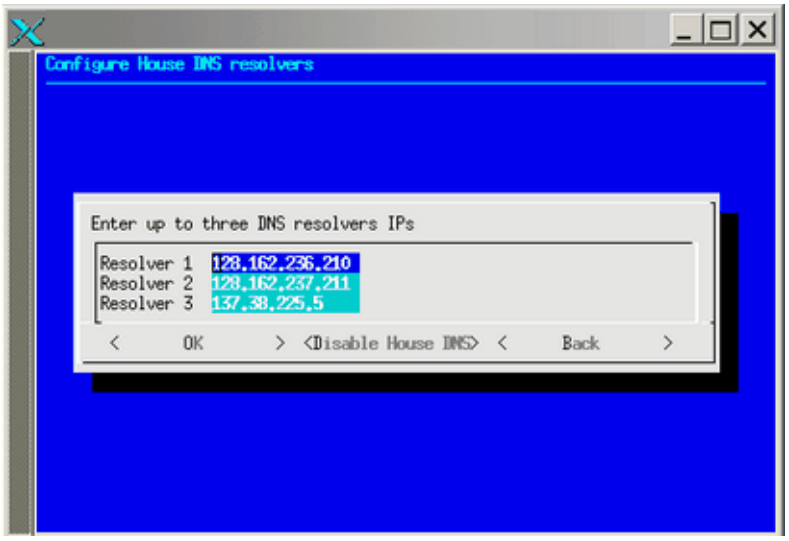

Figure 2-28 Configure House DNS Resolvers Screen

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

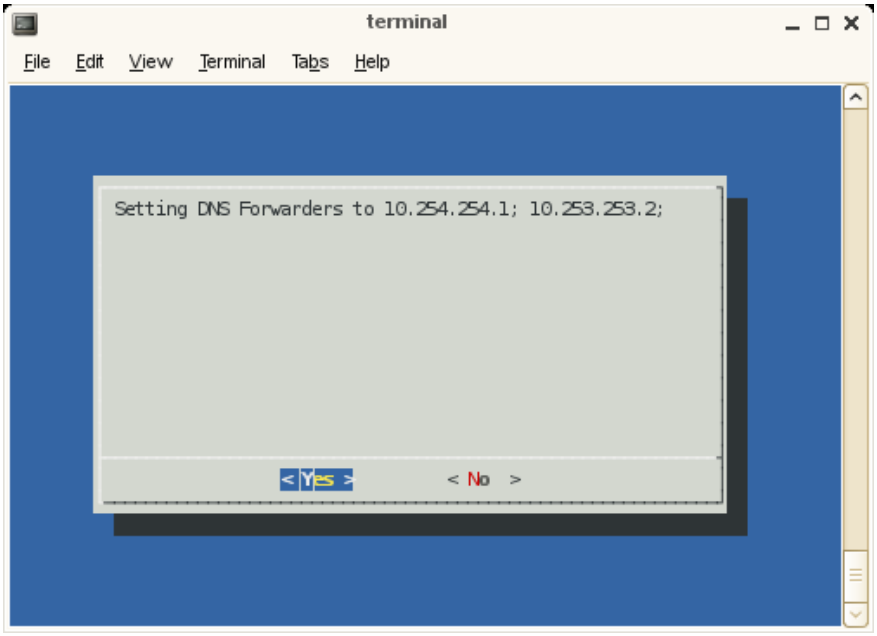

Figure 2-29 Setting DNS Forwarding Screen

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

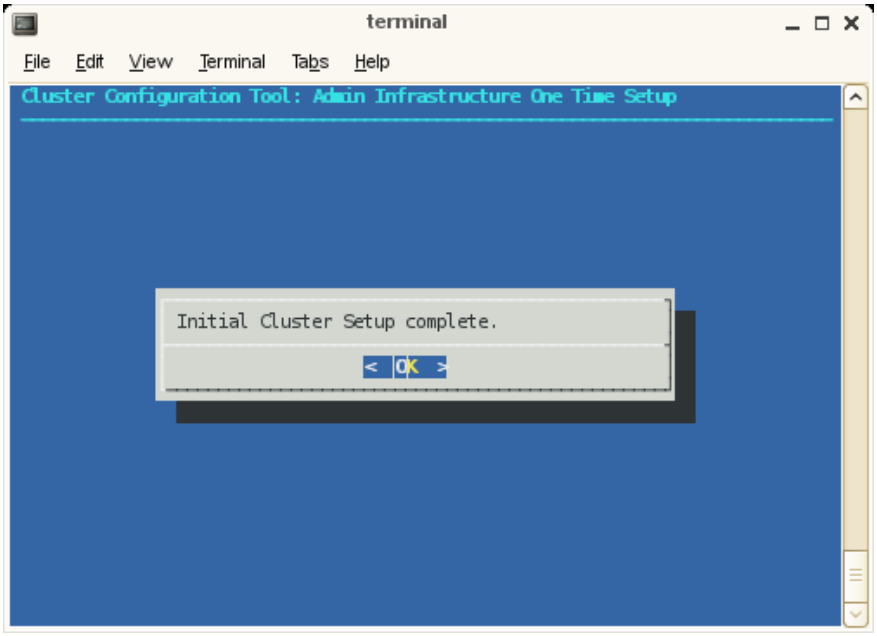

Figure 2-30 Cluster Configuration Tool: Admin Infrastructure One Time Setup Screen

22. Proceed to "Installing Software on the Rack Leader Controllers and Service Nodes" on page 71. 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.

`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 Altix 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 71).

For the Tempo 1.5 release (or later), 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 `cadmin --db-purge --node service0` command that will remove the node entirely from the database (for more information, see "`cadmin`: SGI Tempo Administrative Interface" on page 158). You can then reuse the "`myhost`" name.

Note: When you use the `discover` command to discover an SGI Altix 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:

```
admin ~# discover --h
Discover allows any number of racks, service nodes, or external switches to be
discovered in one command line.
/opt/sgi/sbin/discover --rack <#>[,]
/opt/sgi/sbin/discover --rackset ,[,]
/opt/sgi/sbin/discover --service <#>[,]
/opt/sgi/sbin/discover --switch [,]

--rack: discover a specific rack or set of racks (more than one --rack ok)
--rackset: discover count racks starting at start-number
--service: discover the specified service node
--switch: discover the specified external switch
--delrack: Mark rack leaders as deleted
--delservice: Mark a service node as deleted
--delswitch: Mark an external switch as deleted
--force: Not normally used.  Avoid sanity checks that require input.
```

```
--ignoremac : Not normally used.  See description below.
--macfile : Not normally used.  See description below.

This script is used to discover lead nodes and service nodes 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.
Switches are specified by name.

 is a list of comma separated options that modify how discover
proceeds for the associated node and sets it up for installation.  Hardware
types (see below) have no variable style naming with equal signs.  All
other option types take the form "name=value".

Options include:

 hardware type: A hardware model that affects how discover proceeds.  The list
 of options are at the end of this help message.  If a hardware type isn't
 specified, a default value is used.  Use the 'other' hardware type for a
 service node you supply and manage.  This mode will allocate IPs for you and
 print them to the screen.  It won't be managed by the cluster management
 software.  See the examples section for how to use the hardware type.

 image: You can specify an alternate image to install on to the target
 system.  See the examples for how to specify this.  Alternate image names
 can be supplied for managed service nodes and leaders.  You can later
 adjust the image assigned to the node with the cinstallman tool.

 console_device: Normally, the console device to use for the serial console
 is figured out based on the hardware type.  However, this offers a way to
 override the console device for a given type of hardware.

 net: ib0 or ib1, for external IB switches only.

 type: leaf or spine, for external IB switches only.

 redundant_mgmt_network: Can be used to override the cluster-wide default
 setting; valid values are 'yes' and 'no'.  A value of 'yes' indicates
 that the node should be configured with a redundant managment network.

If you wish to re-discover an existing service node or rack, simply run
```

```
the discover command in the same manner you would normally.  If you wish
to purge a rack or service node entirely -- never to be seen again -- use
--delservice / --delrack for this.  The same applies to external switches.

Expanded descriptions for other arguments:

--macfile:
Instead of discovering MACs by power cycling when instructed, consult the
file for the MAC instead.  This is not normally used.  All MACs to be
discovered must be in the 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

--ignoremac:
A MAC address to ignore during discover operations.  Not normally needed.
Multiple --ignoremac options may be specified.

Instructions on how to proceed with discover will be provided when you
perform a discover.
```

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 Altix 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 121.

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 Tempo v1.6 release (and later), 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 --switch 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 --switch 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 --switch command to list all such nodes in the cluster.

To remove a switch, perform the following:

```
admin:~ # discover --delswitch name
```
where *name* is that of a previously discovered switch.

An example command is, as follows:

```
admin:~ # discover --delswitch 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:

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

Installing Software on the Rack Leader Controllers and Service Nodes

The `discover` command, described in "`discover` Command" on page 66, 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.

Procedure 2-5 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 161.

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 `blademond` process. This process discovers the IRUs and attached blades and sets them up for use. The `blademond` process is described in "`blademond` Command For Automatic Blade Discovery" on page 74, 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 firwmare 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 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, skippnig 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 74.

`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.

Discovering Compute Nodes

This section describes how to discover compute nodes in your Altix 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 `blademond` daemon that runs on the leader nodes (see "`blademond` Command For Automatic Blade Discovery" on page 74).

Procedure 2-6 Discovering Compute Nodes

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

1. Complete the steps in "Installing Software on the Rack Leader Controllers and Service Nodes" on page 71.
2. For instructions on how to configure, start, verify, or stop the InfiniBand Fabric management software on your Altix ICE system, see Chapter 4, "System Fabric Management" on page 177.

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 177.

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.

Unlike system admin controllers (admin nodes), `eth0` on the service node connects to the Altix ICE network (like rack leader controllers). If you wish to have the service node on your house network, you need to configure the second Ethernet interface (`eth1`).

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 on the service node, as follows:

- `eth1` is the house network that you should configure in firstboot.
- 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`.

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 44. 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 2-31.

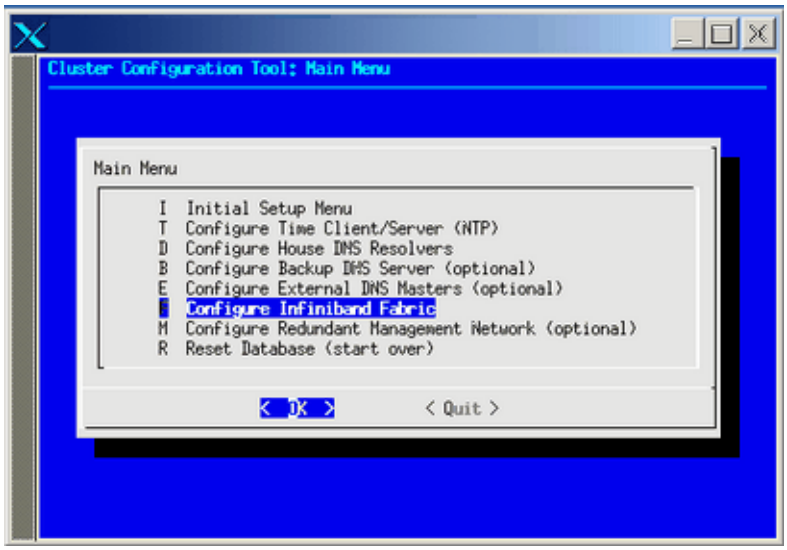

Figure 2-31 Configure InfiniBand Fabric from Cluster Configuration Tool

Select the **Configure InfiniBand Fabric** option, the InfiniBand Fabric Management tool appears, as shown in Figure 2-32.

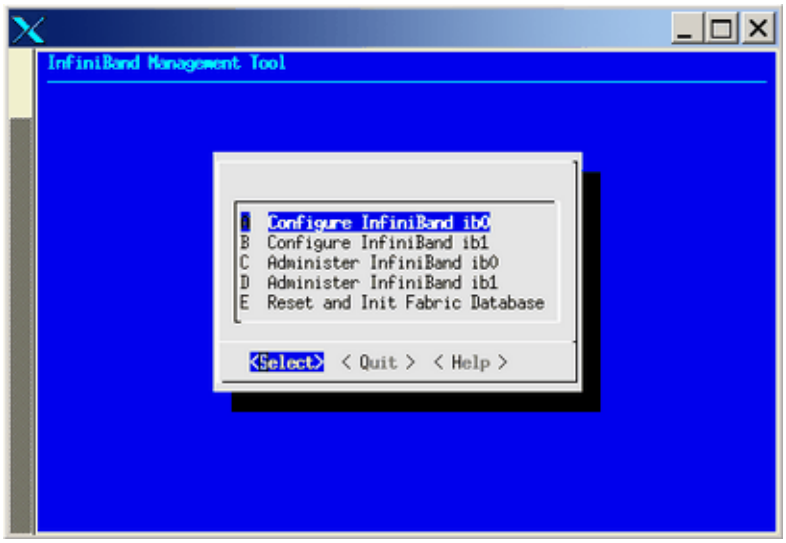

Figure 2-32 InfiniBand Management Tool Screen

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 2-33. Verify the status using the **Status** option.

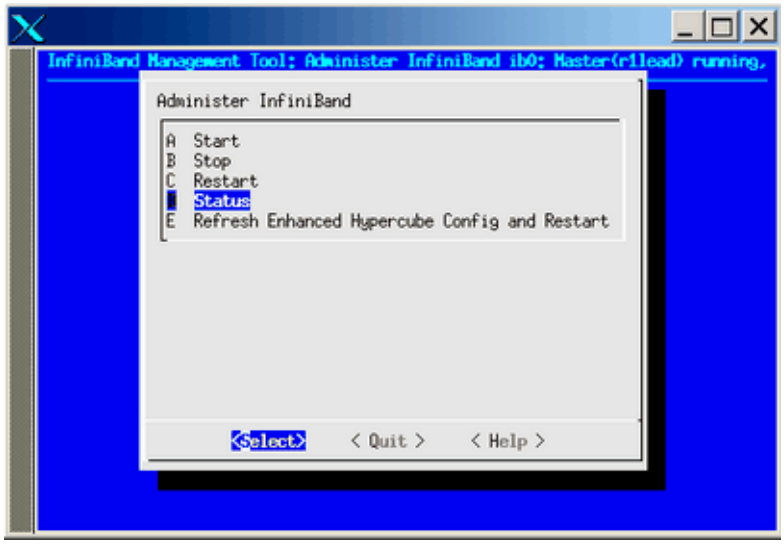

Figure 2-33 Administer InfiniBand GUI

Redundant Management Network

This section describes how to configure the redundant management network. For the SGI Altix 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**. For systems lacking a redundant Ethernet interface, such as, the SGI Altix 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 2-34 on page 79.

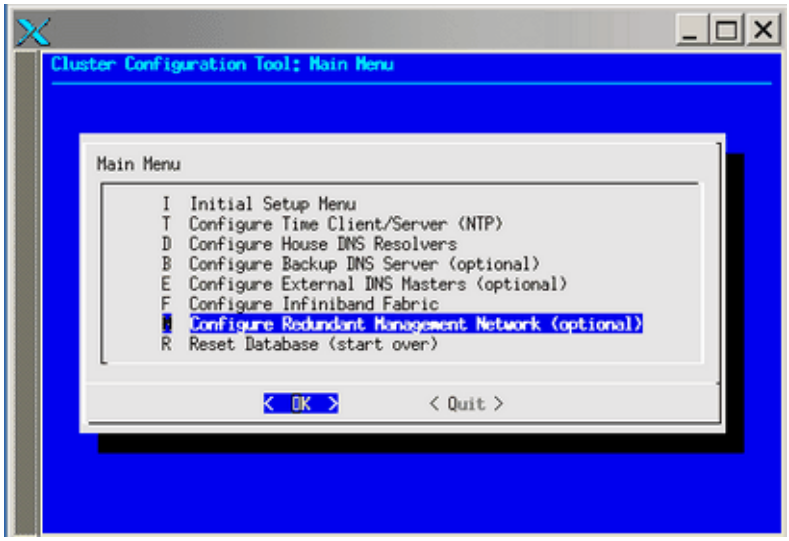

Figure 2-34 Configure Redundant Management Network Option Screen

Select the option and turn off or turn on the redundant network management value, as shown in Figure 2-35 on page 80.

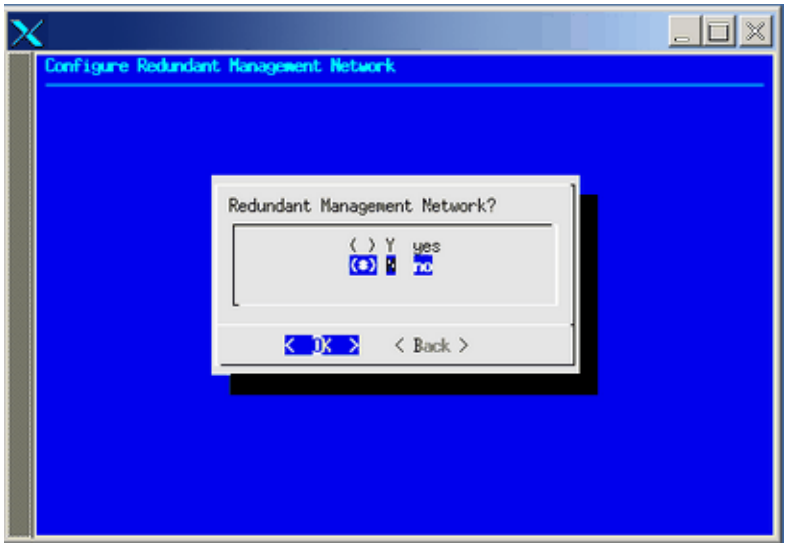

Figure 2-35 Redundant Management Network? Screen

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 a Rack Leader Controller

To turn on the redundant managment 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
```

Configuring the Service Node

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

Service Node Configuration for NAT

You may want to reach network services outside of your SGI Altix 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 2-7 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 132 and "Customizing Software On Your SGI Altix ICE System" on page 124.

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 81) 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 SGI Tempo software the IP addresses of the DNS servers on the house network that the SGI Tempo software can query for hostname resolution.

To do this, use the `configure-cluster` tool (see "`configure-cluster` Command Cluster Configuration Tool" on page 44). 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 SGI Tempo 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 Altix 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 81) 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 2-28 on page 64.

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 2-8 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 82.

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 2-9 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 95.

   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 2-10 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 2-9, page 85. 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 Altix ICE System

These 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 Altix ICE system could create network bottlenecks that the hierarchical design of Altix 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 85.

The instructions in this section assume you are using the service node image provided with the Tempo 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-3600508e000000000307921086e156100 -> ../../sda
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part5 -> ../../sda5
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part6 -> ../../sda6
lrwxrwxrwx 1 root root  9 2008-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a -> ../../sdb
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a-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-3600508e000000000307921086e156100 is in use by your system disk because it has a symbolic link pointing back to ../../sda. So do not consider that device.nThe other disk in the listing has ID scsi-3600508e0000000008dced2cfc3c1930a 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-3600508e0000000008dced2cfc3c1930a 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 2-11 Partitioning and Creating Filesystems for an NFS Home Server on a Service Node

The following example uses /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a 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 Altix ICE System", an example is provided that allows you to determine where your root disk is located so you can avoid accidently 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-3600508e0000000008dced2cfc3c1930a`. 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-3600508e0000000008dced2cfc3c1930a 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-3600508e0000000008dced2cfc3c1930a 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-3600508e0000000008dced2cfc3c1930a 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 128.

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 132.

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 Altix ICE System" on page 92. 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 Altix ICE System" on page 86. 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.

Setting Up a NIS Server for Your Altix ICE System

This section describes how to set up a network information service (NIS) server running SLES 11 for your Altix 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 85. This section covers the following topics:

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 2-12 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 100.

   ---

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 Altix 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 93.

Procedure 2-13 Setting Up a SLES Service Node a as 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 Altix 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 the in `/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 2-14 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 an 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 2-15 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 128.
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 2-16 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 95.

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 SGI Tempo Patches and Updating SGI Altix ICE Systems

This section describes how to update the software on an SGI Altix 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 103 and "Configuring the SMT Using YaST" on page 103).

Overview of Installing SGI Tempo Patches

SGI supplies updates to SGI Tempo 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 Altix ICE system set up a number of package repositories in the /tftpboot directory on the admin node. The SGI Tempo 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 Altix 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.

For additional information on updating your system, see "Upgrading from Prior SGI ProPack Releases to SGI ProPack 7 " on page 109.

There is a small amount of preparation required, in order to setup an SGI Altix 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.

This following sections describe these steps, as follows:

- "Update the Local Package Repositories on the Admin Node" on page 102
- "Installing Updates on Running Admin, Leader, and Service Nodes " on page 106

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 Altix ICE system.

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 SGI Tempo, SGI Foundation, SGI ProPack for Linux 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 "Update the SLES Package Repository" on page 103, 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.

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.

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. SGI Tempo software only uses the mirror abilities of this tool. Mechanisms within SGI Tempo 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 2-17 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 2-18 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
   ```

   Example output is, as follows:

```
admin:~ # smt-catalogs -e SLE11-SDK-Updates
TBD Need new output here
```

   In the example, above, select **7** because it is `x86_64` and `sles`, the others are not.

6. Use the `smt-catalogs -o` comand to only show the enabled catalogs. Make sure that it shows the channels you need to be set up for mirroring.

Warning: SGI Tempo 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 Tempo admin node to channels related to your SGI Tempo 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 106).

Note: Be advised that the first sync with the Novell server will take a very long time.

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 132).

  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 119 and "`cinstallman` Command" on page 121, respectively.

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 108 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 137. 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 146).
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.

Upgrading from Prior SGI ProPack Releases to SGI ProPack 7

For information on upgrading your system from a prior SGI ProPack release to SGI ProPack 7 for Linux, see the release notes. The SGI ProPack 7 release notes can be found in a file named `README.TXT` that is available in `/docs` directory on the SGI ProPack 7 for Linux CD.

The SGI ProPack 7 for Linux release notes get installed to the following location on a system running SGI ProPack 7:
`/usr/share/doc/packages/sgi-propack-7/README.txt`

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 110.

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	Extendedpartition, 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 110 shows a partition table with two available slots. Tempo 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 111. 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 110 or Table 2-2 on page 111. 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`

  When installing, re-partition the admin node to allow for the specified number of slots, default is 2.

- `install_slot`

  Details which root slot to install to for this session, a number from 1 to the number of slots available, default

- `destructive`

  If destructive is set to 1, potentially destructive operations are allowed. Some examples follow.

If an admin node is encountered with exactly one blank/virgin disk, and no parameters are provided, the admin node will be configured with a partition table for two slots and will install an operating system in to the first slot.

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 Tempo use, nothing destructive will happen unless the `destructive=1` parameter is passed.

If an `install_slot` is specified that appears to have been used for something once, it will not be reformatted unless `destructive=1` is supplied.

Note: This detection is simply parted detecting what filesystem was there. If you subsequently re-partitioned your disk using the same partition layout, then partitions from the past may appear to parted as having a valid filesystem when you might not expect that. In this case, just use `destructive=1` once you confirm it is okay to proceed.

If `re_partition_with_slots` is supplied, and a previous Tempo configuration is detected, it will error out 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 communication 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 Tempo 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 121) or by initial discovery with the `discover` command (see "`discover` Command" on page 66).

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 112.

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, peform 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: tempo 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: tempo 2.0 / sles11: first sles
slot 2: tempo 2.0 / sles11: my party
slot 3: tempo 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: tempo 2.0 / sles11: life is good
slot 2: tempo 2.0 / sles11: my party
slot 3: tempo 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
```

System Operation

This chapter describes how to use the SGI Tempo systems management software to operate your Altix ICE system and covers the following topics:

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, how to use `crepo` command to manage software image reposistories, and how to use the `cinstallman` command to create compute and service node images. It covers these topics:

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 Altix ICE System Configuration Framework" on page 142.

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, SGI Tempo, SGI ProPack, and the Linux distribution 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.

For the Tempo v2.0 release, the repositories are, as follows:

Repository	Description
`/tftpboot/sgi/*`	For SGI media
`/tftpboot/other/*`	For any media that is not from SGI
`/tftpboot/distro/*`	For Linux distribution repositories such as SLES
`/tftpboot/`*x*	Customer-supplied repositories

The directory and repository names are determined automatically for YaST compatible media including media supplied by SGI and the Linux `distro` distributors. For customer repositories, the customer supplies a name when pointing at a directory full of RPMs that comprises the repository. You need to populate the directory before pointing the `crepo` command at it. The `crepo` command collects information about update sources and provides that for the `sync-repo-updates` command. Since any media product may have an independent update URL, the `crepo` command has an interface that provides information about all available repositories to commands like `cinstallman`. This means you do not have to supply these URLs on the command line.

Additionally, the `crepo` command constructs default RPM lists based on the suggested packages on SGI media. For example, SGI ProPack may suggest certain packages be installed by default for service, compute, and leader nodes. The `crepo` command collects this information, along with suggested packages, from all other

repositories, and uses it to generate new suggested RPM lists in /etc/opt/sgi/rpmlists. 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:

For more information on rpmlist customization information, see "Creating Compute and Service Node Images Using the cinstallman Command" on page 137.

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/tempo media to the system repositories
       --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

--show                : show available add-on products

--show-distro         : like show, but only reports distro media like sles11

--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.

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.

Use --add for SGI/tempo media, to make the repos and rpms available. If the
supplied SGI/tempo media has suggested rpms from tempo 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
mksiimage commands.  It is assumed you will maintain your own default package
lists, perhaps using the sgi default package lists in /etc/opt/sgi/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 Tempo 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

Starting with the SGI Tempo v1.4 release, 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 --h
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 --rpmlist and --image
                          Optional flags  below:
       --clone          : Clone existing image, requires --source, --image.
                          Doesn't require --rpmlist.
       --recreate       : Like --del-image then --add-image, but preserves any
                          node associations.
                          Requires --image and --rpmlist
       --use-existing   : register an already existing image, doesn't
                          require --rpmlist
       --image {image}  : Specify the image to operate on
       --rpmlist {path} : Provide the rpmlist 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 packes 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 packes 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 Altix ICE System

This section discusses how to manage various nodes on your SGI Altix 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 SGI Tempo Patches and Updating SGI Altix ICE Systems " on page 101.

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 folowing 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.
#
# Modify per your sites requirements.
#
# This script is excecuted 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
```

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 excecuted 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 $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 128.

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 121.

1. Make a clone of the compute node image, as described in "Customizing Software Images" on page 128.
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 your 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 129.

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 121.

---

1. Use the `cinstallman` command to create your own version of the service node image. See "`cinstallman` Command" on page 121.

2. Change directory to the `images` directory, as follows:

   # **cd /var/lib/systemimager/images/**

3. From the system admin controller, copy the RPMs your 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 SGI Tempo.

Usage: cimage OPTION ...

Options
 --help                                 Usage and help text.
 --debug                                Output additional debug information.
 --list-images                          List images and their kernels.
 --list-nodes NODE                      List node(s) and what they are set to.
 --set [OPTION] IMAGE KERNEL NODE       Set node(s) to image and kernel.
    --nfs                               Use NFS roots (default).
    --tmpfs                             Use tmpfs roots.
 --set-default [OPTION] IMAGE KERNEL    Set default image, kernel, rootfs type.
    --nfs                               Use NFS roots (default).
    --tmpfs                             Use tmpfs roots.
 --show-default                         Show default image, kernel, rootfs type.
 --add-db IMAGE                         Add image and its kernels to the db.
 --del-db IMAGE                         Delete image and its kernels from db.
 --update-db IMAGE                      Short-cut for --del-db, then --add-db.
 --push-rack [OPTIONS] IMAGE RACK       Push or update image on rack(s).
    --force                             Bypass the booted nodes check, deletes.
    --update-only                       Skip files newer in dest, no delete.
    --quiet                             Turn off diagnostic information.
 --del-rack IMAGE RACK                  Delete an image from rack(s).
 --clone-image OIMAGE NIMAGE            Clone an existing image to a new image.
 --del-image [OPTIONS] IMAGE            Delete an existing image entirely.
    --quiet                             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 121.

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
r1lead: delete-image: mynewimage done.
```

Using `cinstallman` to Install Packages into Software Images

The packages that make up SGI Tempo, 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 132.
- 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 121.

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 SGI Tempo create these images during the `configure-cluster` installation step (see "Installing Software on the System Admin Controller").

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 119). 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` comand 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 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 132.

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 66. For more information on the `cinstallman` command, see "`cinstallman` Command" on page 121.

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/filetlOeP5:
   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 119).

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 `cinstallman` 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 will automatically be consulted by `cinstallman` command operations going forward including updating images, nodes, and creating images.

4. If you use `cinstallman` to create an image, you will want to add your custom package to the rpmlist you use with the cinstallman command (see "Using `cinstallman` to Install Packages into Software Images" on page 135).

SGI Altix ICE System Configuration Framework

All node types that are part of an SGI Altix 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 Altix 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 framwork 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 132.

- 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 configruation 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 would execute
- If you wish to make a custom version of an 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. SGI Tempo 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

The SGI Tempo 1.3 release included a new 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
 --temponames           modifier: return Tempo 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
r1i0n0
r1i0n1
r1lead
service0
```

To see a list of all compute nodes, perform the following:

```
[admin ~]# cnodes --compute
r1i0n0
r1i0n1
```

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
--noleader	Do not include leader nodes (valid with rack and system domains only).
--noservice	Do not include service nodes (valid with system domain only).
--force	When using wildcards in the target, disable all “safety” checks. Make sure you really want to use this command.
-n, --noexec	Displays, but does not execute, commands that affect power.
-v, --verbose	Print additional information on command progress

The <*target*> argument is one of the following:

`--node`	Applies the action to nodes. Nodes are compute nodes, rack leader controllers (leader nodes), system admin controller (admin node), and service nodes. [default]
`--iru`	Applies the action at the IRU level.
`--rack`	Applies the action at the rack level.
`--system`	Applies the action to the system. You must **not** specify a target with this type.

The <*action*> argument is one of the following:

`--status`	Show the power status of the target, including whether it is booted or not. [default]
`--up | --on`	Powers up the target.
`--down | --off`	Powers down the target.
`--reset`	Performs a hard reset on the target.
`--cycle`	Power cycles the target.
`--boot`	Boots up the target, unless it is already booted. Waits for all targets to boot.
`--reboot`	Reboots the target, even if already booted. Wait for all targets to boot.
`--halt`	Halts and then powers off the target.
`--shutdown`	Shuts down the target, but does not power it off. Waits for targets to shut down.
`--identify` <*interval*>	Turns on the identifying LED for the specified interval in seconds. Uses an interval of 0 to turn off immediately.
`-h, --help`	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 r1i*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 r1i*n*

# cpower --reset r1i*n*

# cpower --cycle r1i*n*

# cpower --identify 5 r1i*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 r1i*
```

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 --iru --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 Altix ICE system.

Shutting Down and Booting

Note: The `--shutdown --off` combination of actions were deprecated in the SGI Tempo v1.2 release. Use the `--halt` option in its place.

It 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 Altix ICE systems, this section remains intact. However, SGI recommends you use the `pdsh` and `pdcp` utilities described in "`pdsh` and `pdcp` Utilities" on page 157.

This section describes the cluster command and control (C3) tool suite for cluster administration and application support.

Note: The SGI Tempo 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
`cexec`(s)	Executes a given command string on each node of a cluster
`cget`	Retrieves a specified file from each node of a cluster and places it into the specified target directory
`ckill`	Runs `kill` on each node of a cluster for a specified process name
`clist`	Lists the names and types of clusters in the cluster configuration file
`cnum`	Returns the node names specified by the range specified on the command line
`cname`	Returns the node positions specified by the node name given on the command line
`cpush`	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 146).

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 Altix ICE system (for information on the hierarchical design of your Altix ICE system see "Basic System Building Blocks" on page 1).

To execute a C3 command on all blades within the default Altix 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 Altix 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 SGI Tempo software populates some `dshgroups` files for the various node types. On the admin node, SGI Tempo 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: SGI Tempo 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:

```
# 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

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 r1lead, service0, or r1i0n0.
[1] Only applies to service 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
[4] Use quoted, semi-colon separated list if more than one master
[5] Only applies to admin and service nodes
```

EXAMPLES

Example 3-6 SGI Tempo 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 Tempo 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 Tempo 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 Tempo 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
```

Console Management

SGI Tempo management systems software uses the open-source console management package called `conserver`. For detailed information on `consever`, see http://www.conserver.com/

An overview of the conserver package is, as follows:

- Manages the console devices of all managed nodes in an Altix 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 sysem 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 conserver(8)
conserver.passwd(5)	User access information for conserver(8)

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).


r1lead 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 SGI Tempo systems management software uses network time protocol (NTP) as the primary mechanism to keep the nodes in your Altix ICE system synchronized. This section describes this mechanism operates on the various Altix ICE components and covers these topics:

- "System Admin Controller NTP" on page 164
- "Rack Leader Controller NTP" on page 164

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 Altix 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 66) 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 Altix 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 Altix 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 excecuted 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 132.

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. For the SGI Tempo v1.8 release (or later), 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 Altix 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 150).
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.
4. 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
```

5. Reboot the all blades in the affected rack (see "Shutting Down and Booting" on page 150).
6. 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:

```
r1lead:~ # cexec --all free
************************ rack_1 ************************
--------- r1i0n0---------
             total       used       free     shared    buffers     cached
Mem:       2060140     206768    1853372          0          4      46256
-/+ buffers/cache:     160508    1899632
Swap:        49144          0      49144
--------- r1i0n1---------
             total       used       free     shared    buffers     cached
Mem:       2060140     137848    1922292          0          4      44200
-/+ buffers/cache:      93644    1966496
```

```
Swap:        49144          0      49144
--------- r1i0n8---------
             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 128.

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.

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:

```
r1lead:~ # 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] [-hnv] [-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] [-hnv] [-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 Altix 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 Altix XE250 or Altix XE270 systems with the 1068-based controller. They do not apply to Altix XE250 or Altix 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]
```

Backing up and Restoring the System Database

The SGI Tempo systems management software captures the relevant data for the managed objects in an SGI Altix 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 Altix ICE system and you need to back up the database on a regular basis.

Managed objects on an SGI Altix ICE include the following

- Altix 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.

System Fabric Management

The InfiniBand network on SGI Altix 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

InfiniBand fabric management on SGI Altix ICE systems is done using the OFED OpenSM software package and the `sgifmcli` tool (see "Fabric Component `sgifmcli` Command" on page 182). 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 Altix 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 22.

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 Altix 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 192.

On SGI Altix 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 23). 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.

The 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 Altix 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:~ # tempo-configure-fabric
```

The **InfiniBand Management Tool** GUI appears, as shown in Figure 4-1 on page 179.

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 44). For more information, see Figure 4-1.

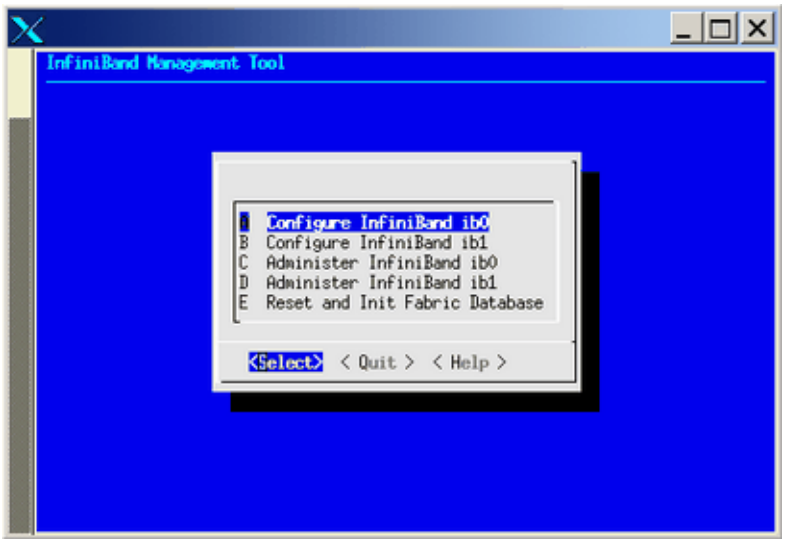

Figure 4-1 InfiniBand Management Tool Screen

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 `tempo-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 182) 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 179) 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 180. For more information, see "Network Topology" on page 188.

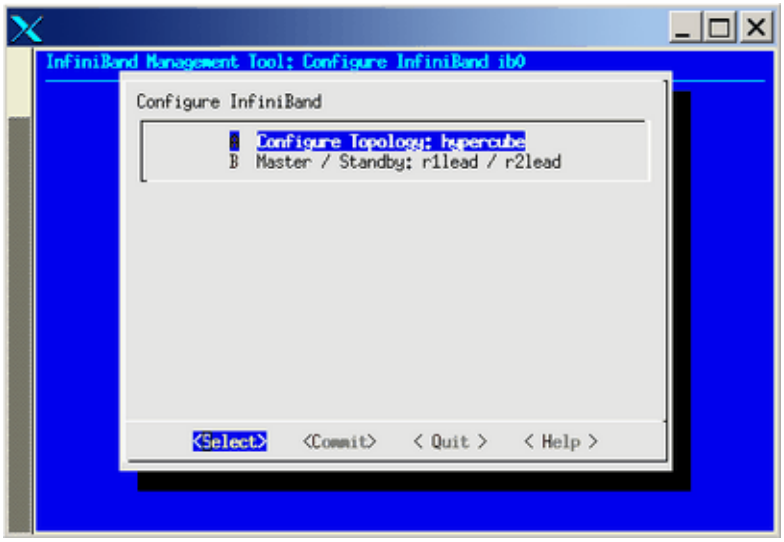

Figure 4-2 Configure Topology Screen

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 182.

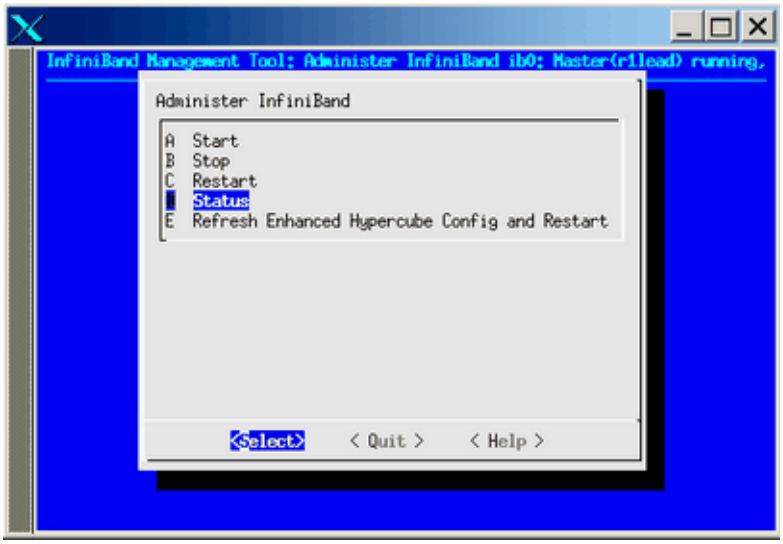

Figure 4-4 Administer InfiniBand Status Option

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 manaager (SM) and the optional standby SM for the specified fabric (see "InfiniBand Fabric Failover Mechanism" on page 192).

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 `tempo-configure-fabric` command (described in "The InfiniBand Management Tool Graphical User Interface" on page 178) 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
`voltaire-isr-9024`	Voltaire ISR 9024
`voltaire-isr-2004`	Voltaire ISR 2004
`voltaire-isr-2012`	Voltaire ISR 2012
`voltaire-isr-9096`	Voltaire ISR 9096
`voltaire-isr-9288`	Voltaire ISR 9288

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 `tempo-configure-fabric` (see "The InfiniBand Management Tool Graphical User Interface" on page 178). 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
`-h, --help`	Displays a help message and the exits

`-V, --version`	Shows the version number of the program
`-v, --verbose [DEBUG | INFO | ERROR]`	Select verbosity level (default: ERROR). Most the messages from `sgmifmcli` are written to a log file named `/var/log/sgifmcli.log`. The default level reports error messages only. `INFO` provides the user with details about the operation of `sgifmcli` 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**
`type`	The `type` option is one of the following: • `--mastersm` – 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 `tempo-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 188). This association is configured with the "`guid`" entry in the corresponding `opensm-ib[01].conf` file.

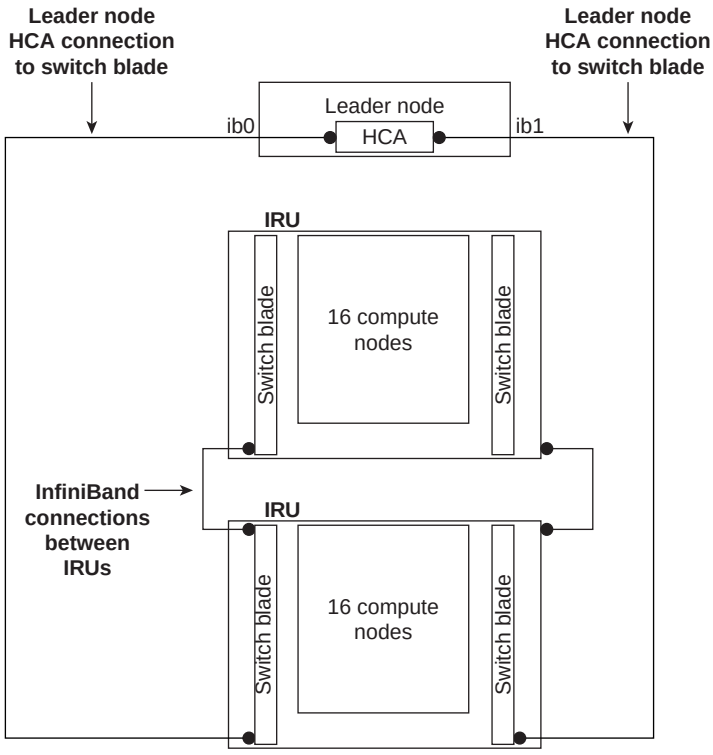

Figure 4-5 Two InfiniBand Fabrics in a System with Two IRUs

Network Topology

For SGI Altix 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 Altix 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 `tempo-configure-fabric` GUI to configure and administer the fabric (see "The InfiniBand Management Tool Graphical User Interface" on page 178).

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 192.

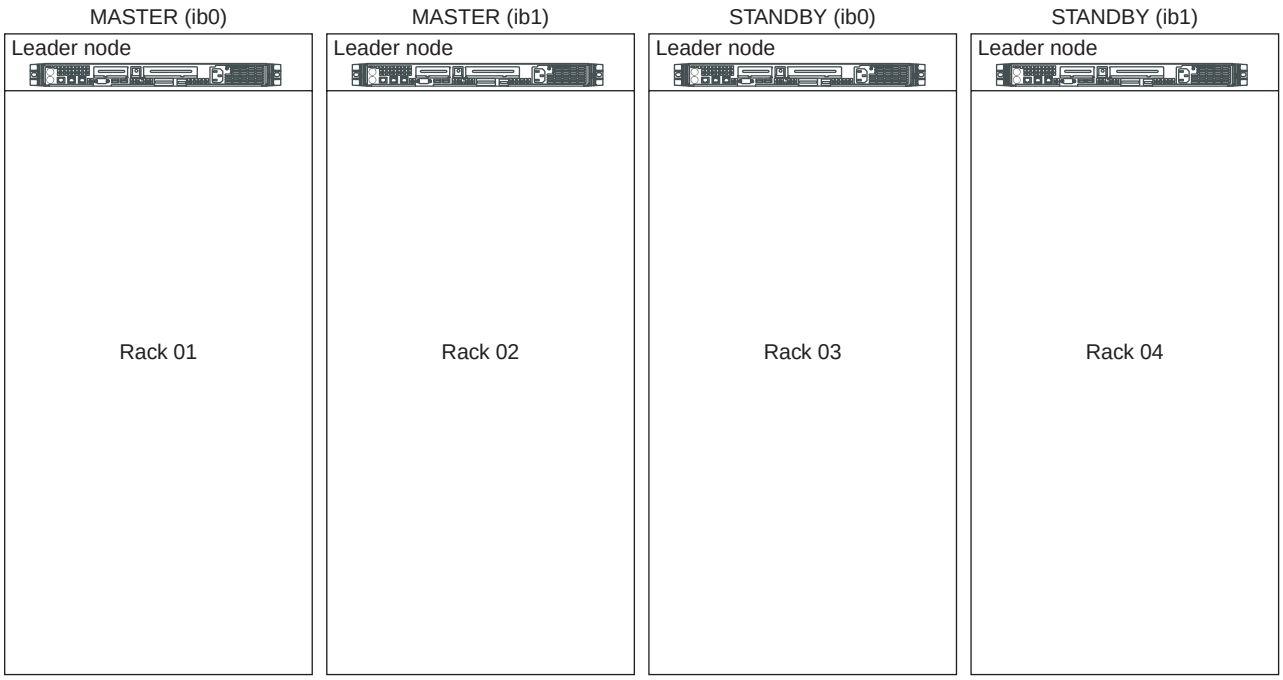

Figure 4-6 `opensm` Software Failover

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 `tempo-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
Toplogy = 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 182.

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 Altix 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.

SGI recommends that you use the Tempo `discover` command (see "`discover` Command" on page 66) 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.

Procedure 4-3 Configuring InfiniBand Fat-tree Network Topology

To configure the InfiniBand fat-tree network topology on an SGI Altix 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.

3. 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.

4. 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.

5. 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.

6. The switches related to a particular fabric can be listed, as follows:

```
# sgifmcli --switchlist --id <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 4-4 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 SGI Tempo cluster view is compared with the InfiniBand cluster view and potential differences are reported.

To verify the `ibo` 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

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 that `ibstat`) command to show the link rate, as follows:

```
r1lead:/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 find errors on a particular or number of HCA's 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:

```
r1lead:/opt/sgi/sbin # perfquery --help
Usage: perfquery [-d(ebug) -G(uid) -a(ll_ports) -r(eset_after_read) -C ca_name -P ca_port -R(eset_only)
 -t(imeout) timeout_ms -V(ersion) -h(elp)] [<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:

```
r1lead:/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:

```
r1lead:/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:

```
r1lead:/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"

r1lead:/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 "r1i0n8-ib0 HCA-1"
Ca       : 0x0008f10403988198 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"
Ca       : 0x0030487aa7840000 ports 1 devid 0x6274 vendid 0x2c9 "r1i0n1-ib0 HCA-1"
Ca       : 0x0030487aa79c0000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n0-ib0 HCA-1"
Ca       : 0x0030487aa7900000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n8-ib0 HCA-1"
```

```
Ca      : 0x0030487aa7980000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n1-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
SYNOPSYS
  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.fdbs   - A dump of the unicast forwarding tables of the fabric
                       switches
    ibdiagnet.mcfdbs - 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 then 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
  5 - Failed to use Topology File
  6 - Failed to load required Package
```

Output which shows no errors means the system is operating correctly:

```
r1lead:/opt/sgi/sbin # ibdiagnet
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdm1.2
-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:

```
r1lead:/opt/sgi/sbin # ibdiagnet -c 5000
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdm1.2
-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

This section describes some common maintenance procedures, as follows:

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 Altix 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 `blademond` daemon. See "Discovering Compute Nodes" on page 74, for more information.

6. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 132 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 Altix 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 `blademond` daemon. See "Discovering Compute Nodes" on page 74, for more information.

Add a New Blade

This section describes how to add a new blade to an Altix 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 `blademond` daemon. See "Discovering Compute Nodes" on page 74, for more information.
3. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 132 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 209.

It covers the following topics:

- "Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode" on page 219

Note: When ordering shelf spare systems from SGI, it is important to order spare nodes appropriate to or in conjunction with your SGI Altix ICE system. This is because the Altix 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 Altix 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 Altix 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 210, "Transfer Disks from Existing Server to the Cold Spare" on page 214, "Migrating to a Cold Spare: Importing the Disk Volumes" on page 214 and "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 216.

Note: If you are using multiple root slots (making use of cascading dual-boot as described in "Cascading Dual-Boot" on page 109) 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 duities 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 Altix `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 211.

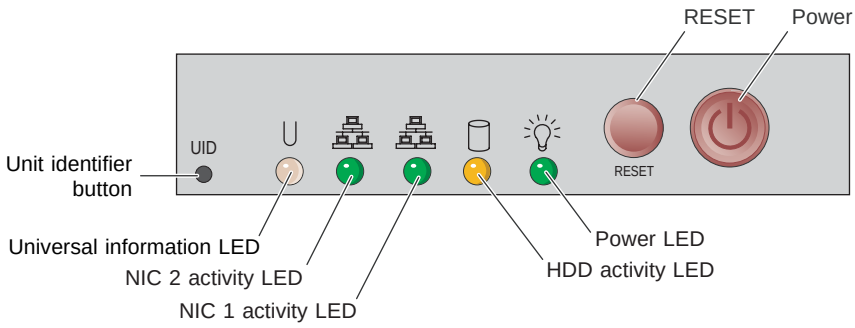

Figure 5-1 Admin/RLC Server Front Panel Controls and Indicator LEDs

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 213 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 214.

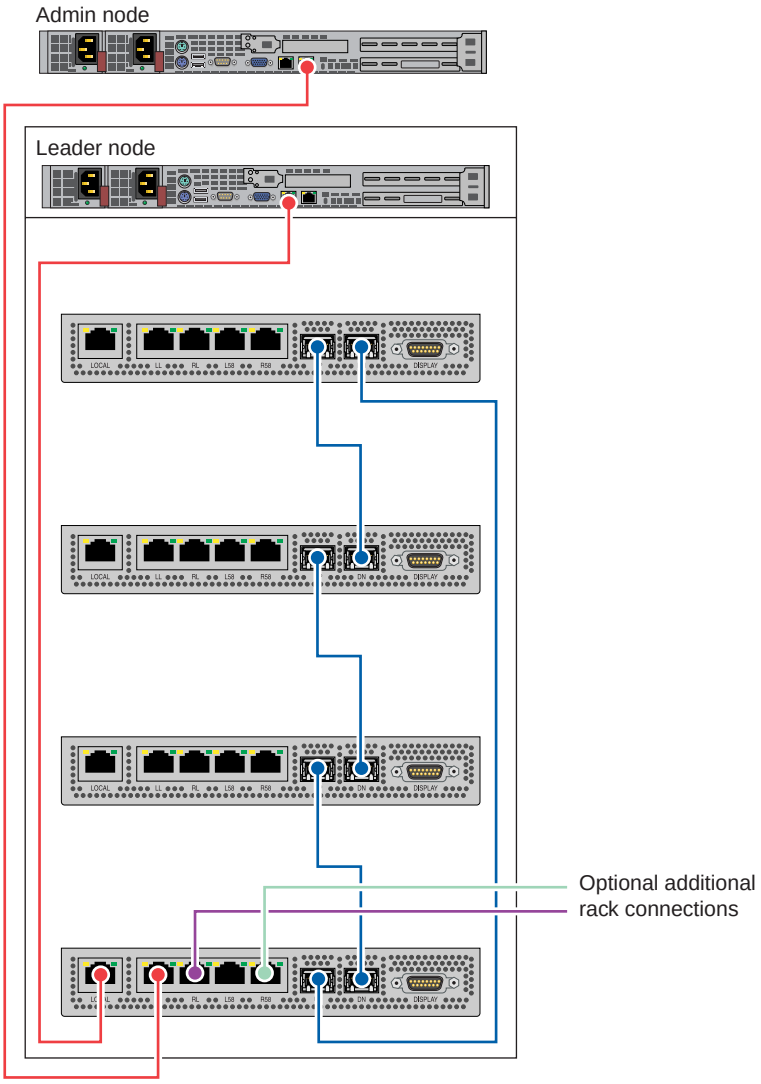

Figure 5-2 Admin/Leader to CMC Cable Examples

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 214.

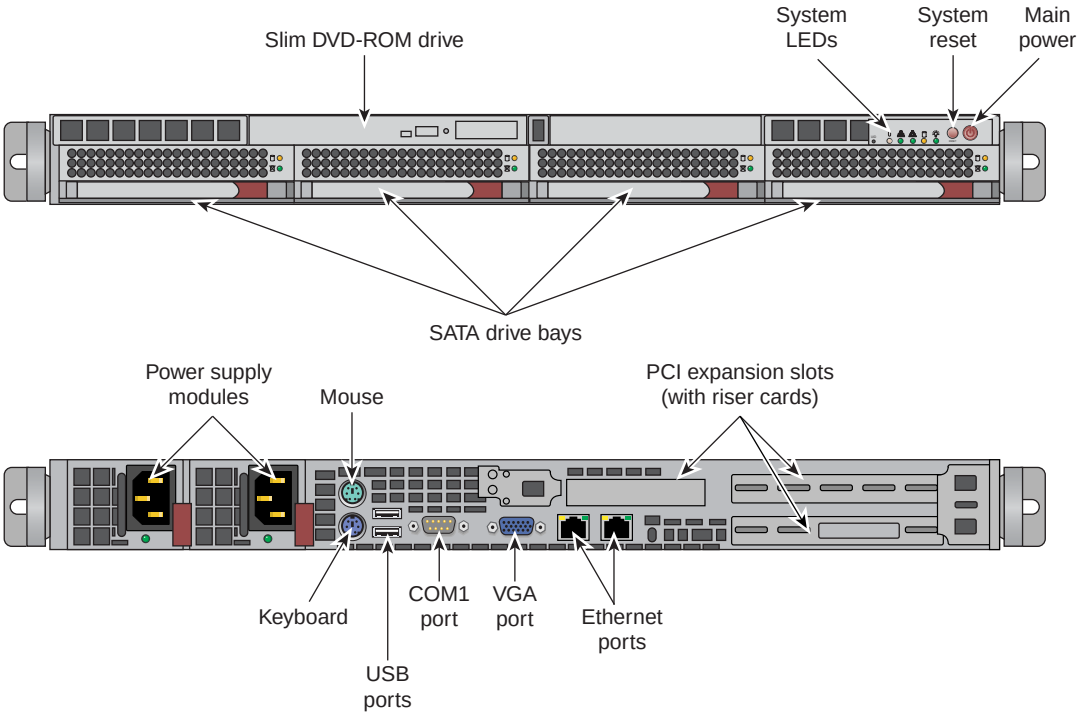

Figure 5-3 Admin/Leader Server Front Features and Rear Connector Locations

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 210.

Note: This section does not apply to SGI Altix XE250 systems with MegaRAID SAS/SATA storage hardware.

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 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 216.

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.

Starting with the Tempo 2.0 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.

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 Tempo 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 214).

   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 SGI Tempo 2.0, the highlighted choice should be : **SUSE Linux Enterprise Server 10 SP3**. 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 SGI Tempo 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 Tempo software.

   ---
10. Reboot the node and let it boot normally.

Procedure 5-8 Migrating to a Cold Spare: Service or Leader Nodes 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 217.

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 214).

   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 214). 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 Tempo 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 Tempo cluster at large when faulty systems are replaced.

Enable or Disable Auto Recovery Mode

Your site may prefer to disable the auto recovery mode. This can be done by using the `cadmin` command. These commands apply:

```
--enable-auto-recovery
--disable-auto-recovery
--show-auto-recovery
```

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 Tempo 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 66.

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 Altix 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 Altix 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 da
arrayd children. The default is "-17,0", hence resulting in the arrayd daem
selected as a candidate by the oom kernel killer thread and children selected a
dates. 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 Altix ICE systems with Tempo, 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 247.

- 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 SGI Altix ICE system running with Tempo software, 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 SGI Tempo 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 Software on the System Admin Controller" on page 34.

To make an inventory snapshot of an Altix 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
r1i0n0 has 8 CPUs
r1i0n1 has 8 CPUs
r1i0n8 has 8 CPUs
r1i1n0 has 8 CPUs
r1i1n1 has 8 CPUs
r1i1n8 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'
r1i0n0 up
r1i0n1 down
```

```
r1i0n8 up
r1i1n0 up
r1i1n1 down
r1i1n8 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'
r1i0n0 192.168.159.10
r1i0n1 192.168.159.11
r1i0n8 192.168.159.18
r1i1n0 192.168.159.26
r1i1n1 192.168.159.27
r1i1n8 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 Altix ICE system. It displays web browser-based, real-time (on demand) histograms of system metrics, as shown in Figure 5-4 on page 226.

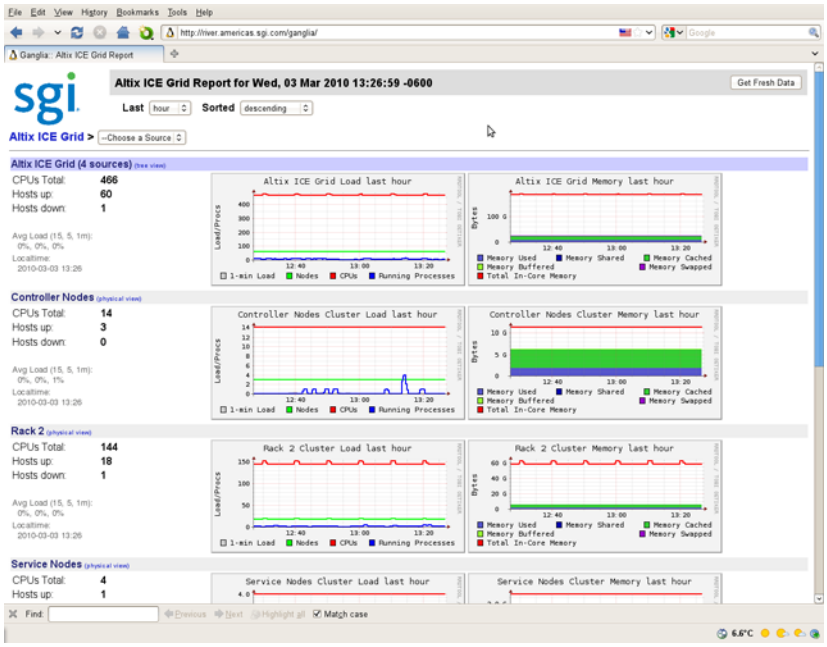

Figure 5-4 Ganglia System Monitor

Detailed information about the Ganglia monitoring system is available at: http://ganglia.info/.

SGI Tempo has devised a Ganglia model for the Altix 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 Altix 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 227 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.

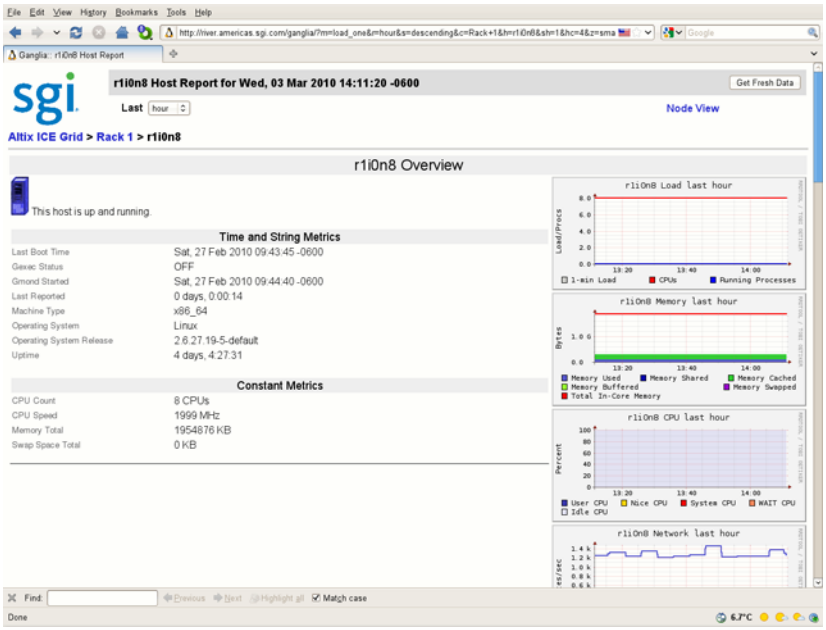

Figure 5-5 Ganglia System Monitoring Node View

System Monitoring Operation

This section describes the operation of the Ganglia system monitor and covers the following topics:

Accessing the Ganglia System Monitor

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. SGI TEMPO 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`
- Embedded Support Partner (ESP)

Whenever critical hardware event occurs, information is forwarded about the event to all three locations. You can observe a critical hardware event via `syslog`, via `sel.log` or using ESP. Furthermore, administrator-defined actions can be triggered via ESP, for instance sending an e-mail notification to the system administrator. For more information on ESP, see `esp`(5) man page and the *SGI Embedded Support Partner User Guide*.

All critical hardware events are summarized under the `BMC_CMC` event type. One particular event holds the following useful information:

`MSG ::=` <*syslog-prefix*> `TEMPO:`<*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 Altix ICE system is monitored by a lightweight daemon called `tempohbc`. 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-devined 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 the *SGI Embedded Support Partner User Guide*.

The HEARTBEAT event contains the following useful information:

`MSG ::=` <*syslog-prefix*> `TEMPO:`<*node*> `EVENT:HEARTBEAT APP:TEMPOHBD 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:

<*node*>	node name, for example, `r1i0n5`
<*date*>	date / time of the event

<text>	Description of event:
	'Heartbeat not detected'
	'Heartbeat lost'

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 Tempo 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 r1lead 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 r1lead cluster.control.suspend_monitorir
```

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 r1lead -s 1 -i 'r1i1n0' cluster.control.metrics

metric:    cluster.control.metrics
host:      r1lead
semantics: discrete instantaneous value
units:     none
samples:   1

                r1i1n0
"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 ::= Tempo 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` Value

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 5-9 Turning Off the `temperature.pmie` Value

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 234.

Adjusting `temperature.pmie` Values

This section describes how to adjust `temperature.pmie` values.

Procedure 5-10 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). 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` Value" on page 234.

Cluster Performance Monitor

You can use the Cluster Performance Monitor to monitor your Altix 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:

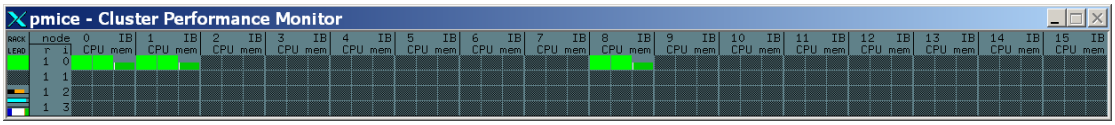

Figure 5-6 pmie- Cluster Performance Monitor

For a usage statement, use the `pmice --h` command, as follows:

```
admin:~ # pmice --h
/usr/bin/pmice: illegal option -- -
Info:
Usage: pmice [options] [pmgadgets 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
```

Setting up the Embedded Support Partner

The Embedded Support Partner (ESP) is a software suite to monitor events, set up proactive notification, and generate reports on SGI Altix systems. This section describes how to set it up on an SGI Altix ICE system. For detailed information about ESP, see *Embedded Support Partner User Guide.*

Procedure 5-11 Setting up the Embedded Support Partner

To set up ESP on an SGI Altix ICE system, perform the following steps:

1. From the admin node, use the `chkconfig` command to make sure that the state of ESP is `on`, as follows:

```
admin:~ # chkconfig --list | grep esp
esp                     0:on  1:on  2:on   3:on   4:on   5:on   6:on
        sgi-esphttp:        on
        sgi_espd:           on
```

ESP should already be running if its `chkconfig` flag is `on`. You can interact with ESP using a web interface or the command line (see Chapter 4, "Setting Up the ESP Environment" in the *Embedded Support Partner User Guide*.

2. From the admin node, create the default ESP user account, as follows:

```
admin:~ # espconfig -createadmin
```

3. Enable the hosts that will be allowed to access ESP with the following commands:

```
admin:~ # espconfig -enable ipaddr 127.0.0.0
admin:~ # espconfig -enable ipaddr 127.0.0.1
admin:~ # espconfig -enable ipaddr IP_address_of_client
```

4. From your laptop or PC system, point your browser to http://mymachine__-admin:5554 and log into ESP.

5. When the ESP login screen appears, login as administrator, use the password partner. After you login, the **System Information** screen appears (see Chapter 2, "Accessing ESP" *Embedded Support Partner User Guide*.

6. Now enter the **Customer Profile** information, as follows:

   a. **Select ESP Administration** from the menu.

   b. Click on **Customer Profile** (if not selected by default).

   c. Fill in the form and then click **Add**.

   d. Click **Commit**; or **Update** if already filled out.

7. Use ESP to **Examine Inventory**, as follows:

   a. Select **Reports Hardware Generate Report**.

   b. Select **Reports Software Generate Report**.

   c. You can search for individual packages by entering the name in the search box (below the system host name) and then selecting **GO** on the right hand side of the screen. You can also use the down arrow to select a package in this search box.

8. Use ESP to enable or disable **Performance Monitoring**, as follows:

   a. Select **Configuration** (from the top level menu) and then select **Performance Monitoring**.

b. Enable **PMIE**.

c. Disable the **PMIE** rule **cpu.util**.

d. Select **Commit**.

e. Select **Configuration System Monitoring** and enable the service `pmcd`.

f. Select **Update** and **Commit** (this may take a few minutes).

9. Use ESP to examine errors logs, as follows:

   a. From the top level menus, select **Report Events**.

   b. Then select **Last 30 days** and **All Classes** before clicking on **Generate Report**.

10. Use ESP to enable or disable **Notification**

    Notification of events is handled by `espnotify`. The notication can be of types e-mail, system console, or graphics console. The notifications are enabled or disabled by specific actions. So after configuring the notification action you can enable or disable the notification, as follows:

    a. Select **Configuration Actions** and click **Continue**.

    b. Decide on the notification format and then check and select **Continue** and **Commit**.

    c. Select **Enable**/**Disable** from the third level menu, and click to enable the notification you set up.

    d. Click **Commit**.

Troubleshooting

This section describes some troubleshooting tools and covers these topics:

dbdump Command

You can run the dbdump script to see an inventory of the Altix 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 Altix 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 }
...
```

`tempo-info-gather` Command

The `tempo-info-gather` command enables to collect vital system data especially when troubleshooting problems. The `tempo-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 Altix ICE nodes

To see a usage statement for the `tempo-info-gather` command, perform the following:

```
admin:/opt/sgi/sbin # tempo-info-gather  -h
 usage: tempo-info-gather [-h] [-P path] [-o file]
        tempo-info-gather -h            # Print this usage page
        tempo-info-gather -o file       # Tar and gzip the directories
into file (imply -n)
        tempo-info-gather -p path       # Directory to write the data
(default /var/tmp/tempo)
```

`cminfo` Command

The `cminfo` command is used internally by many of the SGI Tempo scripts that are used to discover, configure, and manage an SGI Altix 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:

```
r1lead:~ # 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 downlonad 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 1 8
^e c l 1 t       #traceback
^e c l 1 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 Altix ICE system comes preinstalled with the appropriate firmware. See your SGI field support person for any BMC, BIOS, and CMC firmware updates.

The SGI Altix 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 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 Altix 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
r1i0n8: 05/29/2007
r1i1n0: 05/29/2007
r1i1n1: 05/29/2007
r1i1n8: 05/29/2007


BMC versions:
-------------
admin: 1.31
r1lead: 1.31
service0: 1.31
r1i0n0: 1.29
r1i0n1: 1.29
r1i0n8: 1.29
r1i1n0: 1.29
r1i1n1: 1.29
```

```
r1i1n8: 1.29


CMC versions:
-------------
r1i0c: 0.0.9pre10
r1i1c: 0.0.9pre10


Infiniband versions:
--------------------
r1lead: 4.7.600
service0: 4.7.600
r1i0n0: 1.2.0
r1i0n0: 1.2.0
r1i0n1: 1.2.0
r1i0n1: 1.2.0
r1i0n8: 1.2.0
r1i0n8: 1.2.0
r1i1n0: 1.2.0
r1i1n0: 1.2.0
r1i1n1: 1.2.0
r1i1n1: 1.2.0
r1i1n8: 1.2.0
r1i1n8: 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 necessay 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
        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"
```

```
            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
```

```
#  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 epilouge 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 necessay 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
```

```
   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
```

```
#          - 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 SGI's ProPack bcfree command.
sub FreeBufferCache {
my $CMD_LINE;
my $RETURN;
my $BCFREE;
my $BCFREE_OPTS;

$BCFREE="/usr/bin/bcfree";
$BCFREE_OPTS="-a -s";

   if (-e "${BCFREE}") {
      $CMD_LINE="${BCFREE} ${BCFREE_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;
```

```
   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_lsmod=`lsmod | grep -c msr`;    # 0=not loaded, 1=msr loaded

   if ( $msr_lsmod == 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();
```

```
    &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