SGI® Tempo System Administrator's Guide

007–4993–005

The SGI Tempo systems management software stack depends on several open source packages which require attribution. They are as follows:

c3:

conserver:

New Features in This Manual

This rewrite of the *SGI Tempo System Administrator's Guide* supports the SGI Tempo systems management software (v1.4).

Major Documentation Changes

Added the following:

- Updated information "Hostnames" on page 26
- Updated information in "InfiniBand Network" on page 28
- Updated information in "Installing Software on the System Admin Controller" on page 31
- "`crepo` Command" on page 105
- "`cinstallman` Command" on page 109
- "Using `cinstallman` to Install Packages into Software Images" on page 121
- "Using `yum` to Install Packages on Running Service or Leader Nodes" on page 121
- "Creating Compute and Service Node Images Using the `cinstallman` Command" on page 122
- "Installing a Service Node with a Non-default Image" on page 124
- "Using a Custom Repository for Site Packages" on page 124
- "Switching Compute Nodes to a `tmpfs` Root" on page 149
- "RAID Utility" on page 151
- Updated information in "InfiniBand Fabric Administrative Tools" on page 158
- "Configuring the InfiniBand Fat-tree Network Topology" on page 169
- "Node Replacement Procedure for Admin, Leader, and Service Nodes" on page 186
- Updated information in "Monitoring System Metrics with Performance Co-Pilot" on page 201

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)

Contents

Figures

Examples

Procedures

About This Guide

This guide is a reference document for people who manage the operation of SGI Altix ICE 8200 series systems running SUSE Linux Enterprise Server 10 Service Pack 2 with SGI ProPack 6 for Linux. It describes how to use SGI Tempo systems management software (v1.4) to perform general system discovery, installation, configuration, and operations on SGI Altix ICE 8200 series systems.

This manual contains the following chapters:

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.

For a list of manuals supporting SGI ProPack for Linux releases covering the following topics, see the *SGI ProPack 6 for Linux Start Here*:

- SGI documentation supporting SGI Altix ICE systems
- Novell documentation for SUSE Linux Enterprise Server 10 (SLES10)
- 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 5 for Linux 6 Start Here*, the SGI ProPack 6 release notes, which contain the latest information about software and documentation in this release, the list of RPMs distributed with SGI ProPack 6, and a useful migration guide, which contains helpful hints and advice for customers moving from earlier versions of SGI ProPack to SGI ProPack 6, can be found in the `/docs` directory on the SGI ProPack 6 Open/Free Source CD.

  The SGI ProPack 6 for Linux release notes get installed to the following location on a system running SGI ProPack 6: `/usr/share/doc/sgi-propack-5/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
  1140 East Arques Avenue
  Sunnyvale, CA 94085–4602

SGI values your comments and will respond to them promptly.

SGI Altix ICE 8200 Series System Overview

The SGI Altix Integrated Compute Environment (ICE) 8200 series 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 8200 series system and covers the following topics:

- "Hardware Overview" on page 1
- "Networks" on page 13
- "Network Interface Naming Conventions" on page 22

Hardware Overview

This section provides a brief overview of the SGI Altix ICE 8200 series system hardware and covers the following topics:

- "Basic System Building Blocks" on page 1
- "System Nodes" on page 8

For a detailed description, see the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

Basic System Building Blocks

The SGI Altix ICE 8200 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, as shown in Figure 1-1 on page 2. Each compute blade supports two either dual–core or quad-core Xeon processor sockets and eight fully-buffered, double-data-rate two (DDR2) memory dual in-line memory module (DIMMs). Four IRUs can reside in a custom designed 42U high rack.

One rack supports a maximum of 512 processor cores and 2TB of memory.

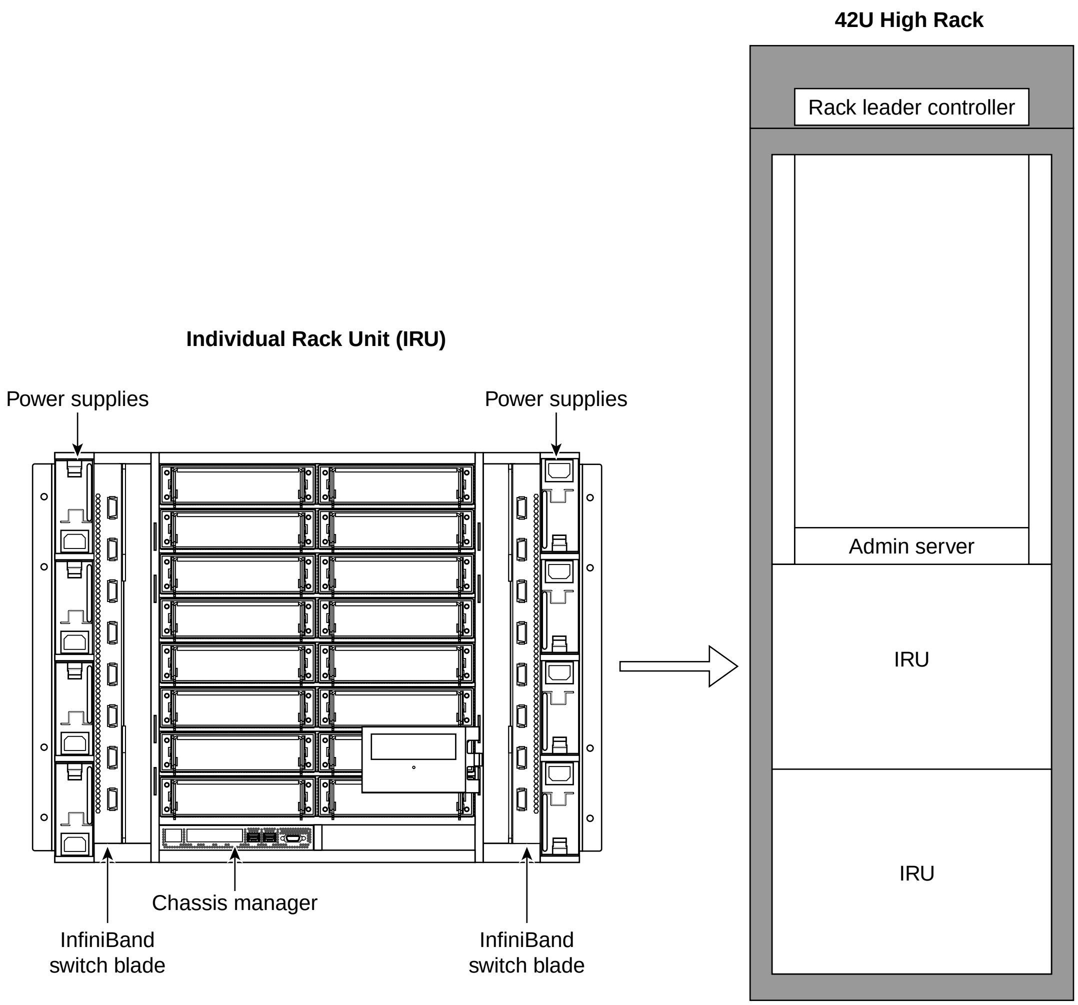

Figure 1-1 Basic System Building Blocks

This hardware overview section covers the following topics:

- "InfiniBand Fabric" on page 3
- "Gigabit Ethernet Network" on page 4
- "Individual Rack Unit" on page 4

InfiniBand Fabric

The SGI Altix ICE 8200 system topology is based on an InfiniBand interconnect. Internal InfiniBand switch ASICs of the IRU eliminate the need for external InfiniBand switches. The dual high-speed, low-latency double data rate (DDR) InfiniBand backplanes built into the IRUs provide for fast communication between nodes and racks.

An InfiniBand switch blade provides the interface between compute blades within the same chassis and also between compute blades in separate IRUs. Fabric management software monitors and controls the InfiniBand fabric. SGI Altix ICE 8200 systems are configured with two InfiniBand fabrics, designated as `ib0` and `ib1`. In order to maximize performance, SGI advises that the `ib0` fabric be used for all MPI traffic, in this case, for example SGI MPT. 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 157.

Note: The "`ib0` fabric" is a convenient shorthand for "the fabric which is connected to the `ib0` interface on most of the nodes". In the case of the storage service node, there are four interfaces called `ib0` through `ib3`, all of which are connected to the `ib1` fabric (see "Storage Service Node " on page 12 and "NAS Configuration for Multiple IB Interfaces" on page 89).

The SGI Altix ICE system is a distributed memory system as opposed to a shared memory system like that used in the SGI Altix 450 or SGI Altix 4700 high-performance compute servers. Instead of passing pointers into a shared virtual address space, parallel processes in an application pass messages and each process has its own dedicated processor and address space.

Just like a multi-processor shared memory system, an SGI Alitx ICE system can be shared among multiple applications. For instance, one application may run on 16 processors in the system while another application runs on a different set of eight processors. Very large systems may run dozens of separate, independent applications at the same time.

Typically, each process of an MPI job runs exclusively on a processor. Multiple processes can share a single processor, through standard Linux context switching, but this can have a significant effect on application performance. A parallel program can only finish when all of its sub-processes have finished. If one process is delayed because it is sharing a processor and memory with another application, then the entire parallel program is delayed. This gets slightly more complicated when systems have multiple processors (and/or multiple cores) that share memory, but the basic rule is that a process is run on a dedicated processor core.

Gigabit Ethernet Network

An Gigabit Ethernet connection network built into the backplane of the IRUs provides a control network isolated from application data. Traverse cables provide connection between IRUs and between racks. For more information on how the Gigabit Ethernet connection fabric is used, see "VLANs" on page 17.

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 8200 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 8200 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 17 and "Network Interface Naming Conventions" on page 22.

The rack leader controller (leader node) and system admin controller (admin node) are described in the section that follows ("System Nodes" on page 8).

Chassis Manager

Figure 1-2 on page 7 shows chassis manager cabling.

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 7 and Figure 1-3 on page 12 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 leader node in the rack. The system admin controller accesses the BMC on each compute node in the rack via VLAN running over a Gigabit Ethernet (GigE) connection (see Figure 1-7 on page 19).

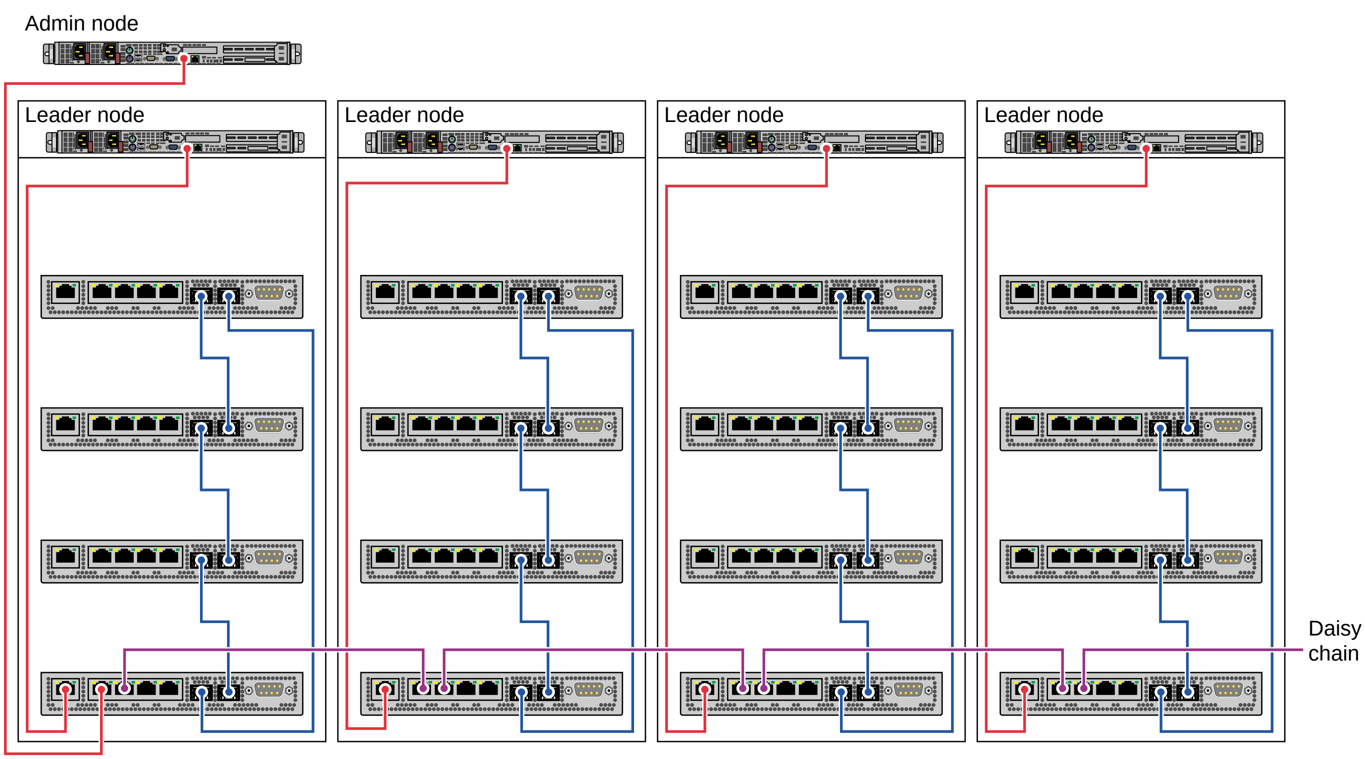

Figure 1-2 Chassis Manager Cabling

Figure 1-3 on page 12 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 8200 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 8200 system using SGI Tempo systems management software. There is only one system admin controller per SGI Altix ICE 8200 system, as shown in Figure 1-2 on page 7 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 admin node 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 1-2 on page 7. A GigE connection provides the network connection to other leader nodes and to first IRU within its rack as shown in Figure 1-3 on page 12 and Figure 1-4 on page 14. 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 the fabric management software to monitor and function the InfiniBand fabric on one or more leader nodes in your Altix ICE system
- Monitors, functions, and receives data from the IRUs within its rack
- Monitors, functions, and receives 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
- Provides a shared, read-only kernel image and initrd image and a root filesystem for the compute nodes in its rack
- Provides non-shared, read-write system storage (for `/var`, `/etc` and `/root`) and a minimal swap space for the compute nodes within its rack

The leader node can contain multiple images for the compute nodes. "Customizing Software On Your SGI Altix ICE System" on page 111 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 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).

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 1-4 on page 14. The compute nodes are controlled and monitored by the leader node for their rack as shown in Figure 1-2 on page 7. 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 1-5 on page 15). For more information on these jacks, see "Gigabit Ethernet (GigE) and 10/100 Ethernet Connections" on page 15.

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*.

Individual Rack Unit

The individual rack unit (IRU) is one of the basic building blocks of the SGI Altix ICE 8200 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 1-4 on page 14. 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 1-3 on page 12 shows a service node and a storage service node (NAS cube).

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 7 and Figure 1-3 on page 12 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.

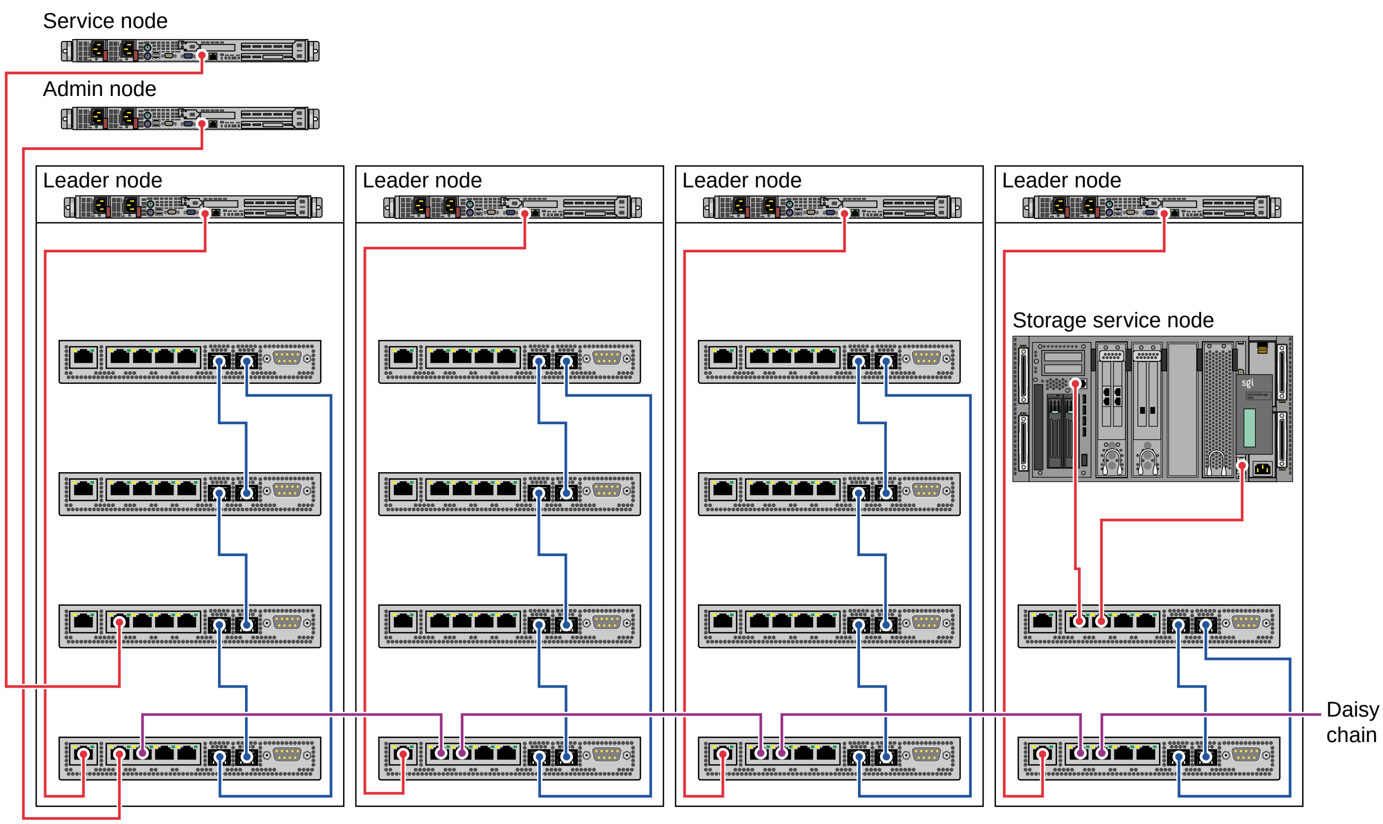

Figure 1-3 Service Nodes

Networks

This section describes the Gigabit Ethernet (GigE) and 10/100 Ethernet connections and the InfiniBand fabric in an SGI Altix ICE 8200 system and covers the following topics:

Networks Overview

This section describes the various network connections in the SGI Altix ICE 8200 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 1-4 on page 14. 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 1-6 on page 18, Figure 1-7 on page 19, and Figure 1-8 on page 20. 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 1-4 on page 14).

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 Cluster Command Control (C3) tool cexec(1) utility that enables the execution of any standard command on all Altix ICE system nodes.

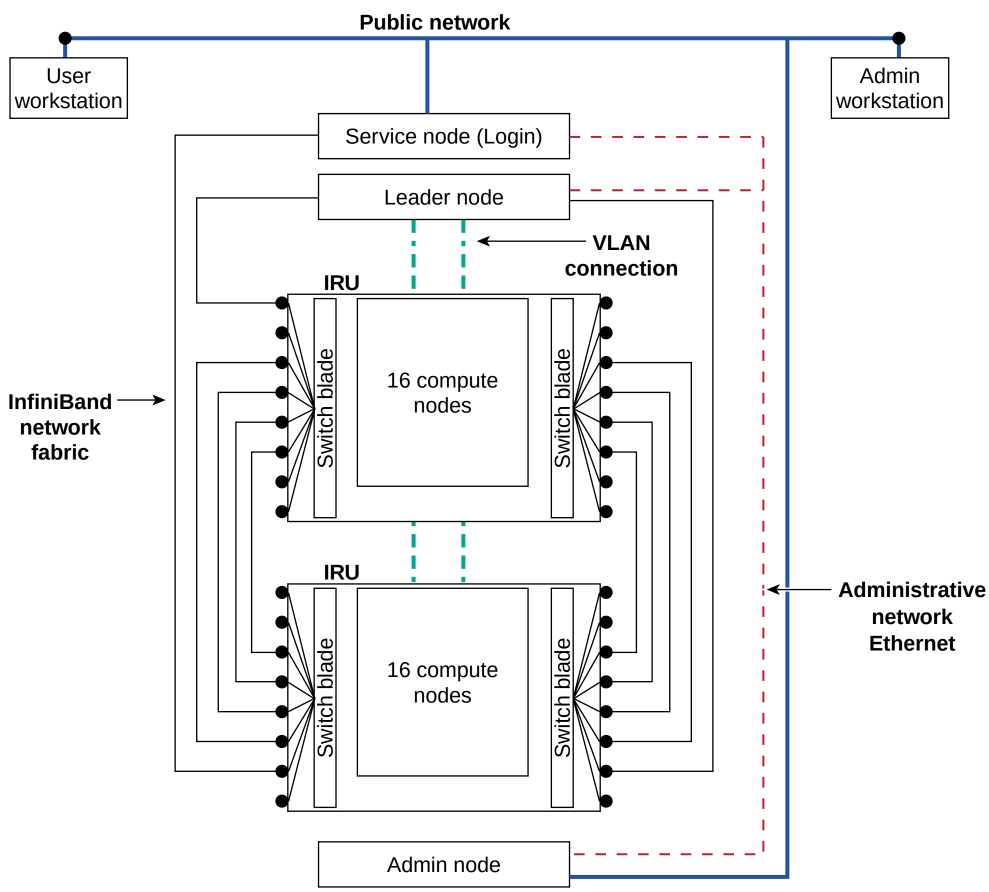

Figure 1-4 Network Connections In a System With Two IRUs

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 192.

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 1-5 on page 15 shows a more detailed view of the Chassis manager.

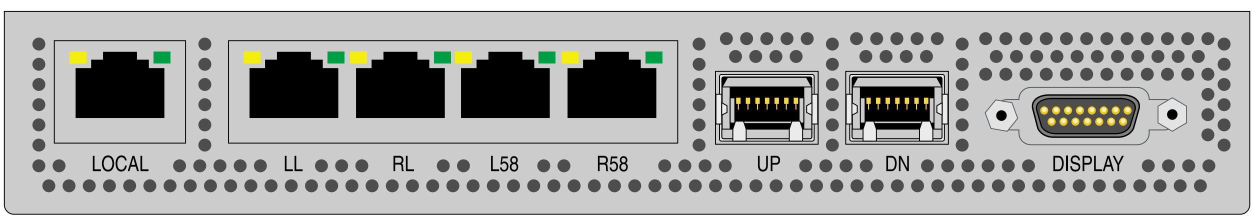

Figure 1-5 Chassis Manager

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 is "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 1-5 on page 15. The switches are stacked in a ring so failure of one link still allows traffic to flow in the opposite direction on the ring.

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 1-5 on page 15. 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 1-2 on page 7.

- **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 1-2 on page 7 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 1-2 on page 7 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 1-3 on page 12.

For small, one IRU configurations, the **L58** and **R58** ports (see Figure 1-5 on page 15) 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 1-6 on page 18.

- `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 1-6 on page 18.

`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 31. 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.

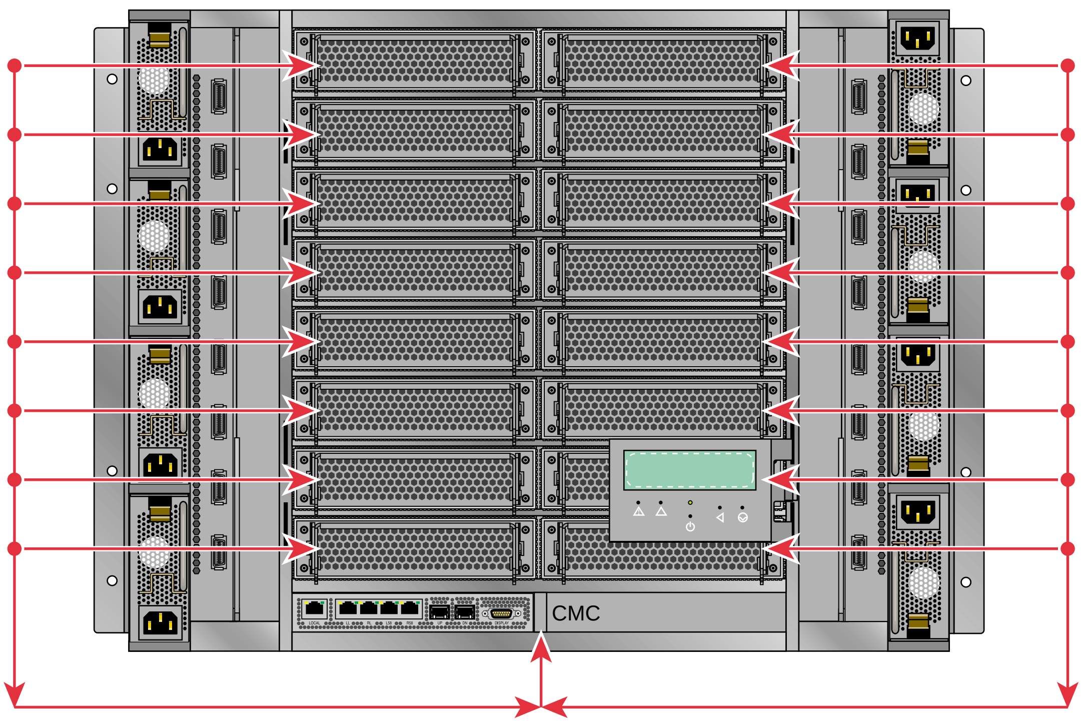

Figure 1-6 `VLAN_GBE` and `VLAN_BMC` Network Connections - IRU View

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).

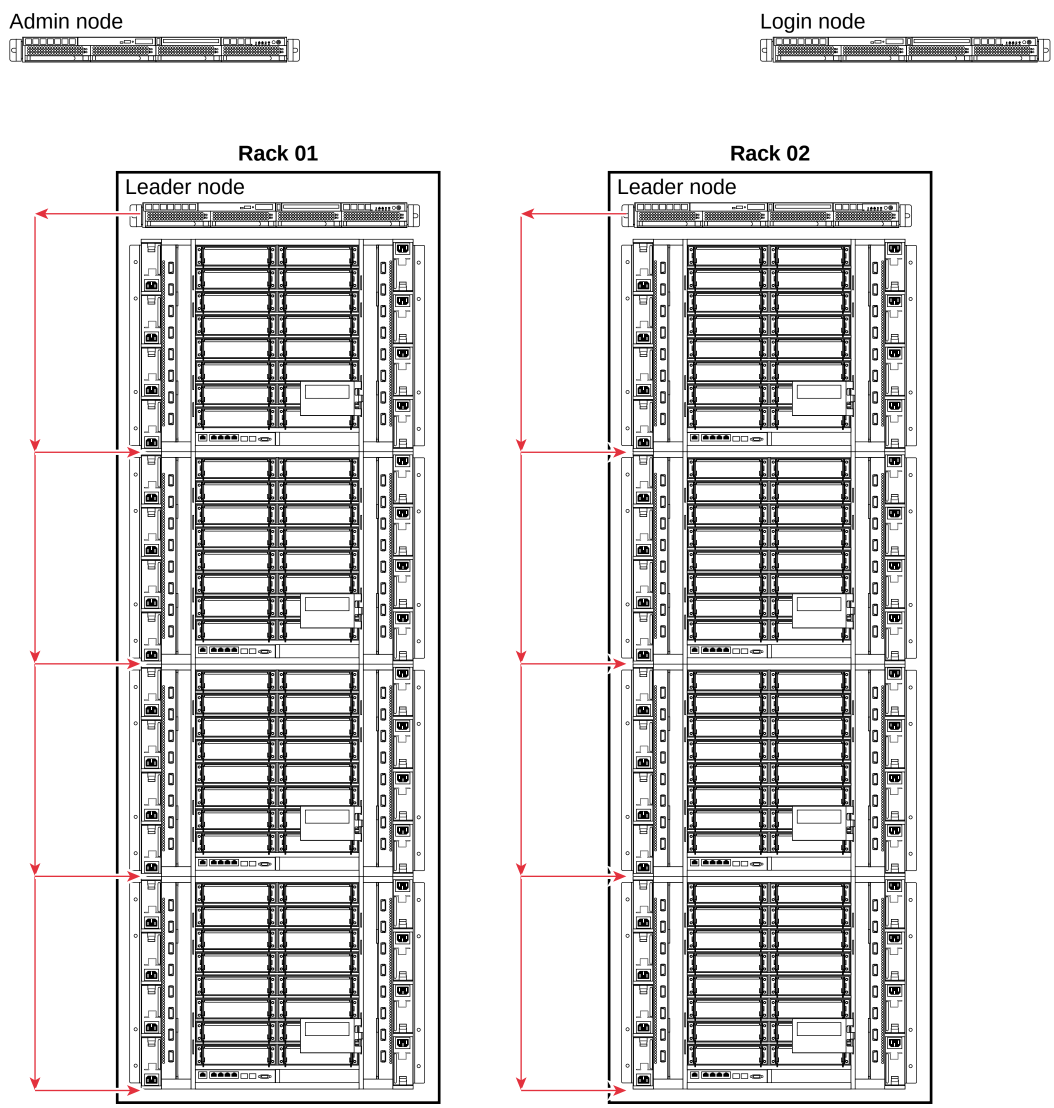

Figure 1-7 `VLAN_GBE` and `VLAN_BMC` Network Connections – Rack View

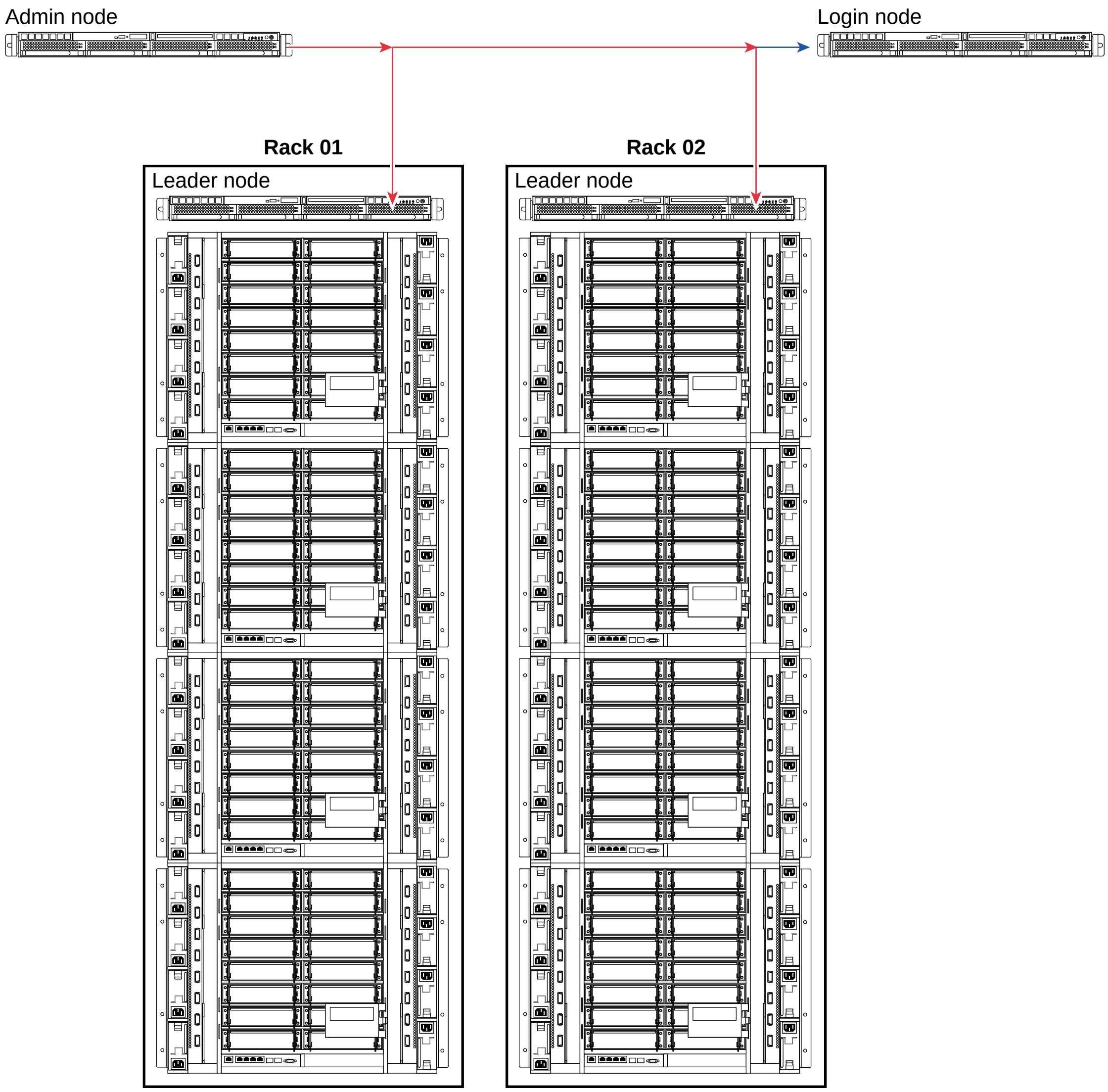

Figure 1-8 VLAN_HEAD Network Connections

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 1-6 on page 18, see chapter 1 and chapter 6 of the *SGI Altix ICE 8200 Series System Hardware User's Guide*.

InfiniBand Fabric

The InfiniBand fabric connects the service nodes, leader nodes, and the compute blades. It does not connect to the admin node or the CMCs. The InfiniBand network has two separate network fabrics, `ib0` and `ib1`. The host channel adapter (HCA) in the leader node has two ports that connect separately to the bottom IRU in the rack.

Each IRU has two 24-port switches (see Switch blade in Figure 1-9 on page 22). Each switch is on a separate fabric.

On each switch, 16 ports go to the 16 compute blades. Each compute blade has two, single port HCAs and each HCA connects to a fabric. Therefore, both switches connect to each blade.

Of the remaining eight ports on each switch, currently six of them are used to connect to either IRUs in the same rack or to IRUs in other racks. One port of one IRU in a rack (usually the first or 0th IRU) connects to the leader node in that rack.

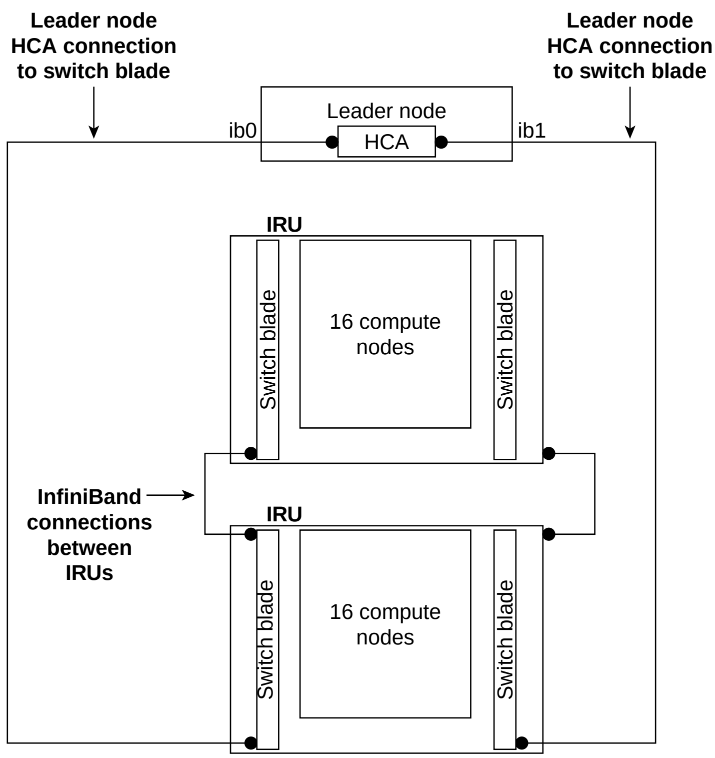

Figure 1-9 Two InfiniBand Fabrics in a System with Two IRUs

Network Interface Naming Conventions

As described in "Networks" on page 13, you can think of an SGI Altix ICE 8200 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 17, 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 23
- "`VLAN_Head` Network Connections" on page 23

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 8200 system (see "System Nodes" on page 8)

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 8200 system. These IDs are partly not routable. See "Non-resolvable Names" on page 26.

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 24 shows the VLAN_Head network connection names. See Figure 1-8 on page 20.

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 65.

VLAN_GBE Network Connections

Table 1-2 on page 24 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 1-6 on page 18). 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.

When more than one GigE interface is present, the names lead-eth1, iYnZ-eth1, and so on, may be used.

VLAN_BMC Network Connections

Table 1-3 on page 25 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 1-6 on page 18).

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 26 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, a blade (node), or a CMC needs to be uniquely identified within the Altix ICE system. Table 1-5 on page 26 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
Blade	rXiYnZ
CMC	rXiYc

Hostnames

Hostnames are distinct from the non-resolvable names and are shown in Table 1-6 on page 27. 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: Service hostnames need to match the node name, that is, serviceX.

For the SGI Tempo v1.4 release, 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.

For the SGI Tempo v1.4 release, 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, system admin controllers (leader nodes), and compute nodes, but not to the system admin controller (admin node) or CMCs. Table 1-7 on page 28 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 26).

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

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:

- "Configuring MFG-installed SGI Altix ICE System" on page 29
- "`configure-cluster` Command" on page 30
- "Installing Software on the System Admin Controller" on page 31
- "`discover` Command" on page 62
- "Installing Software on the Rack Leader Controllers and Service Nodes" on page 65
- "`blademond` Command For Automatic Blade Discovery" on page 68
- "Discovering Compute Nodes" on page 68
- "Service Node Installation and Configuration" on page 69
- "Configuring the Service Node" on page 70
- "Setting Up an NFS Home Server on a Service Node for Your Altix ICE System" on page 76
- "Service Node NFS Server Alternate: Re-exporting House NFS Servers" on page 82
- "Setting Up a NIS Server for Your Altix ICE System" on page 84
- "Installing SGI Tempo Patches and Updating SGI Altix ICE Systems " on page 92

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 92 and "Upgrading from SGI ProPack 5 SP5 to SGI ProPack 6" on page 101.

Configuring MFG-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 MFG-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 the Altix ICE cluster domain or any internal network ranges, 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" on page 30).

   b. Choose the **Reset Database** operation. Read the on-screen instructions.

   c. After the database has been reset, choose **Initial Setup Menu**.

   d. Start the options in this menu in order starting at **Configure Time Client/Server (NTP)**.

   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 62.

`configure-cluster` Command

The `configure-cluster` command launches a cluster configuration tool. 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 SLES10 SP2 and optionally SGI ProPack 6. 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)

Information on using this tool is described in the procedure in the following section, see "Installing Software on the System Admin Controller" on page 31.

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 8200 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 SLES10 SP2 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 192.

Procedure 2-2 Installing Software on the System Admin Controller

To install 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 32.

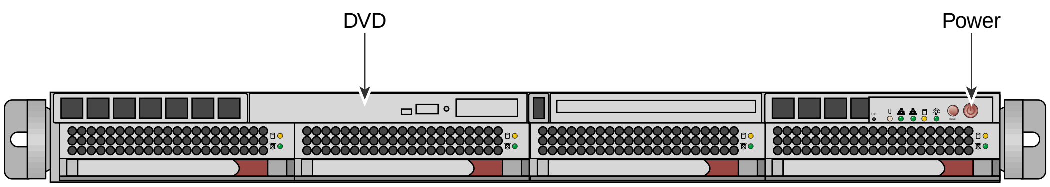

Figure 2-1 System Admin Controller Power On Button and DVD Drive

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 the 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 the you hit `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 a 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 32.

3. An autoinstall message appears on your console, as follows:

```
SGI Admin Node Autoinstallation DVD

This is the SGI Admin Node autoinstall DVD.
If you proceed, the entire system will be erased and re-installed.

You may install from the vga screen or from the serial console.
The default system console will match the console you used for installation.
Hit ENTER for the vga screen or type "serial" for serial.

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

Experts: You may choose to use the "auto" label (auto reboot and skip firstboot questions).
You may also append the "netinst" option
with an nfs path (hostname:/mntpoint/file.iso) to nfs mount the ISO.

Press ENTER to send autoinstallation output to the vga screen.
Type "serial" at the boot prompt to send autoinstallation output to the serial console.
```

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.

You can hit the **ENTER** button at the boot prompt. The boot `initrd.image` executes, the hard drive is patitioned 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 with the admin install DVD, any previous data on the disks is destroyed. This step takes several minutes. When the installation is complete, the system admin controller DVD drive automatically ejects the DVD.

---

4. Once installation of software on the system admin controller is complete, remove the DVD from the DVD drive.
5. Once the system has been installed, enter the `reboot` command to reboot your system.

Note: The output will go to the VGA screen unless you used **serial** for the admin install DVD earlier.

You will see messages about the system admin controller booting the kernel. You can ignore any messages about a few services that may fail to start.

Note: If you used the serial console for installation (**serial** is not the default), the console output and configuration questions from `yast2 firstboot` will go to the serial port. Pressing `Ctrl -l` will re-draw the `yast2 firstboot` screen when you are using the serial console.

---

6. After the reboot completes, the YaST first boot installation tool starts and a **`Welcome`** screen appears, as shown in Figure 2-2 on page 35. Click on the **Next** button to proceed.

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.

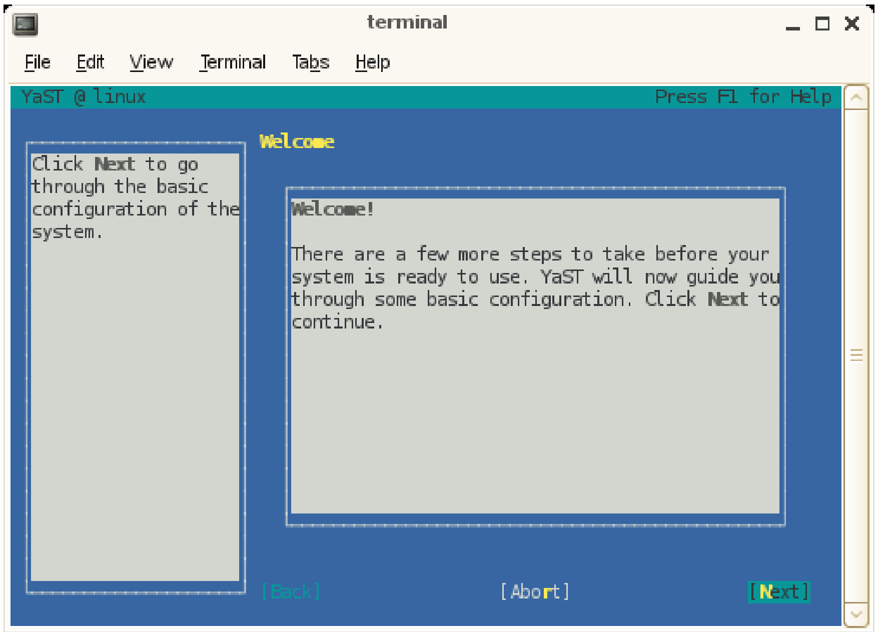

Figure 2-2 YaST **Welcome** Screen

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

7. From the **Hostname and Name Server Configuration** screen, as shown in Figure 2-3 on page 36, enter the hostname and domain name of your system in the appropriate fields. Make sure that **Change Hostname via DHCP** is unselected (no x should appear in the box). Click on the **Next** button to continue.

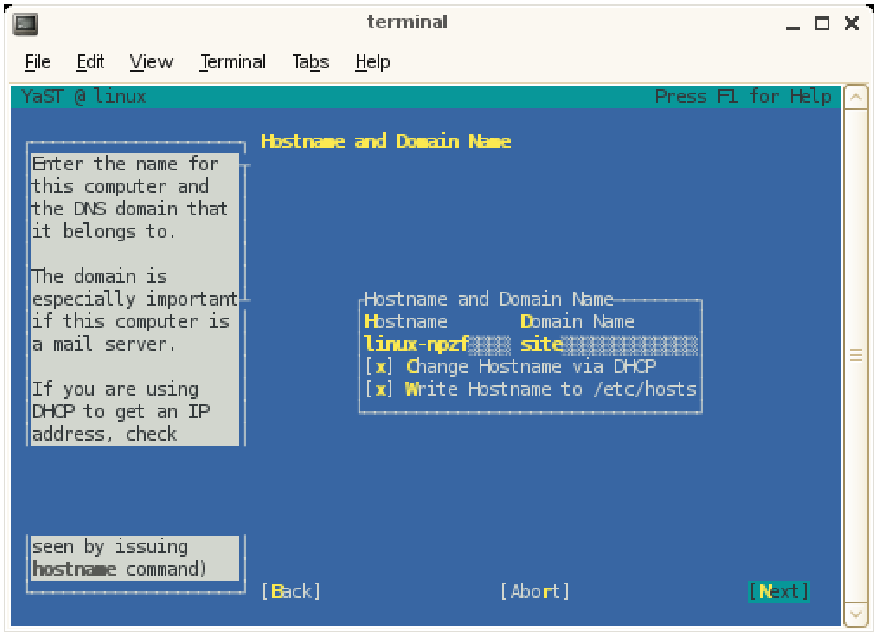

Figure 2-3 Hostname and Name Server Configuration Screen

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

---

8. From the **Network Card Configuration Interfaces** screen, shows the suggested configuration as shown in Figure 2-4 on page 37. Click **Next** to continue.

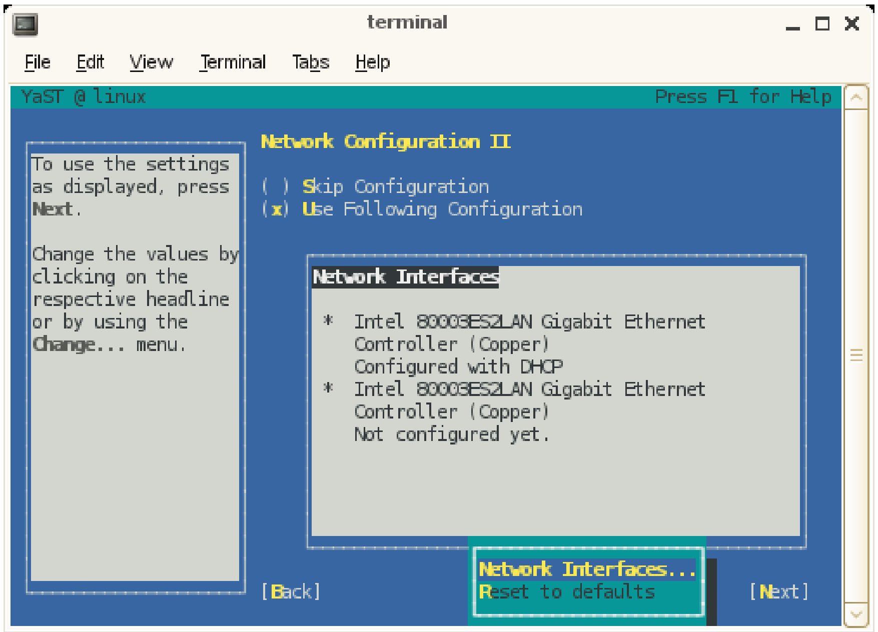

Figure 2-4 **Network Card Configuration Interfaces** Screen

9. From the **Network Card Configuration Overview** screen, configure the first card under **Name** to establish the public network (sometimes called the house network) connection to your SGI Altix ICE 8200 system.

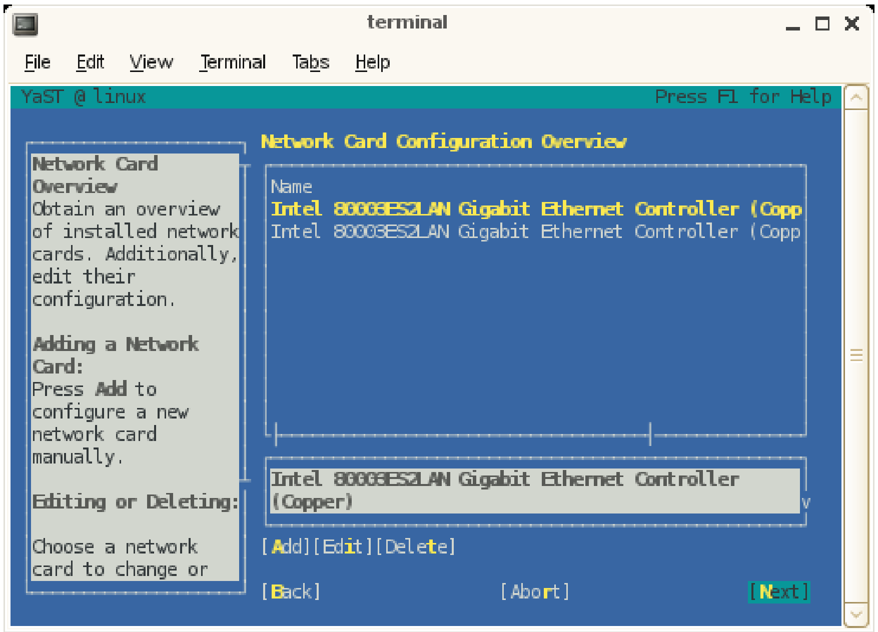

Figure 2-5 Network Card Configuration Overview Screen

Note: Do NOT configure the second interface at this time. A script will do this for you in a later step.

Click on the **Next** button to continue.

10. From the **Network Address Setup** screen, choose dynamic address setup via DHCP or enter the IP address for the system admin controller. This is your public/house network information. Click on the **Next** button to continue.

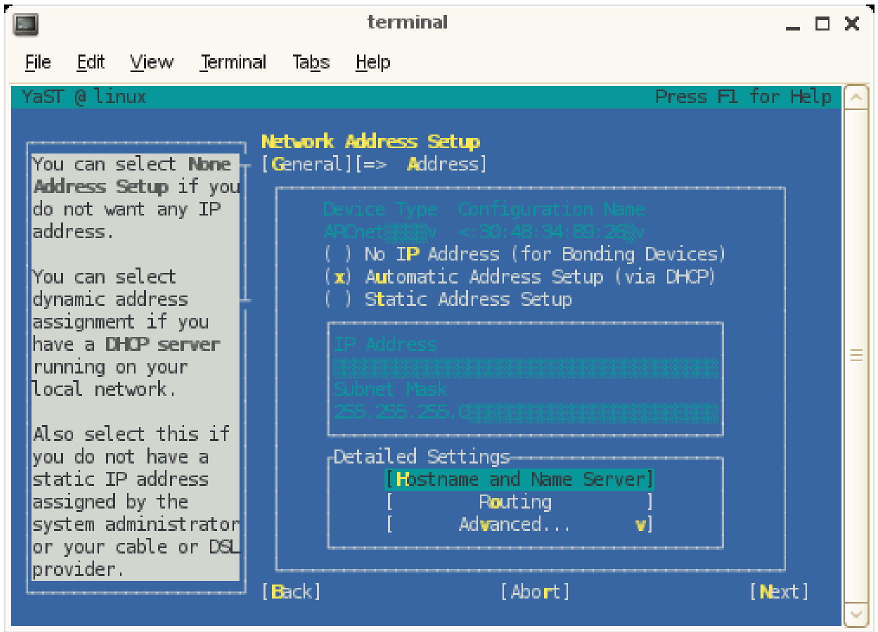

Figure 2-6 Network Address Setup Screen

11. From the **Hostname and Name Server Configuration** screen, enter the name and DNS domain name as shown in Figure 2-7 on page 40. Note that the hostname was entered in step 7.

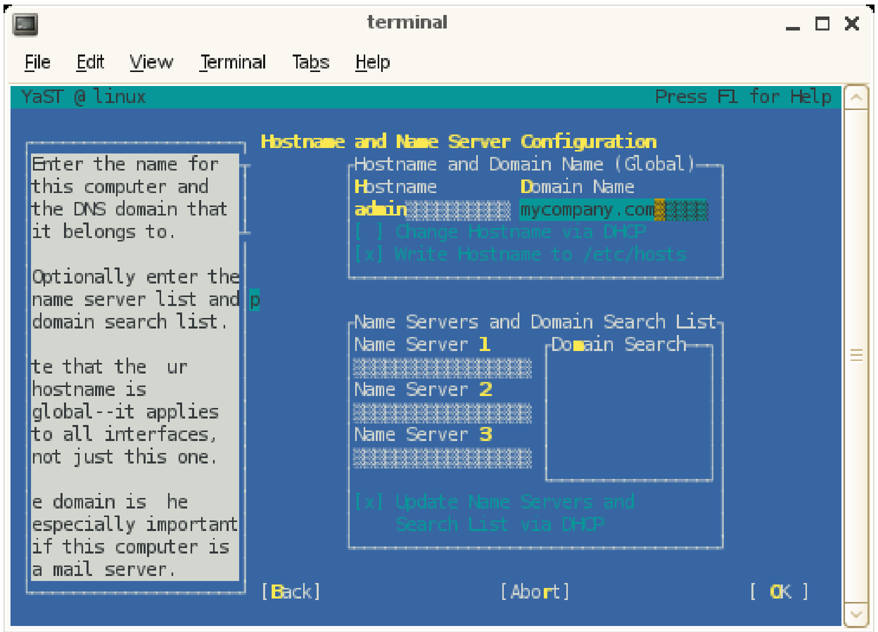

Figure 2-7 Hostname and Name Server Configuration Screen

12. From the **Routing Configuration** screen, enter the appropriate gateway address and netmask. Click on the **OK** button to continue.
13. From the **Clock and Time Zone** screen, select the appropriate region and time zone. Click on the **Next** button to continue.
14. From the **Password for the System Administrator "root"** screen, set the root password. Click on the **Next** button to continue.
15. From the **User Authentication Method** screen, select the authentication method to use for the users on your system. Click on the **Accept** button to continue.
16. Enter the user's full name, username, and user password in the **New Local User** screen. Click on the **Next** button to continue.
17. From the **Hardware Configuration** screen, select **Use Following Configuration**. Click on the **Next** button to continue.

18. An **Installation Competed** screen appears, as show in Figure 2-8 on page 41. Click on the **Finish** button.

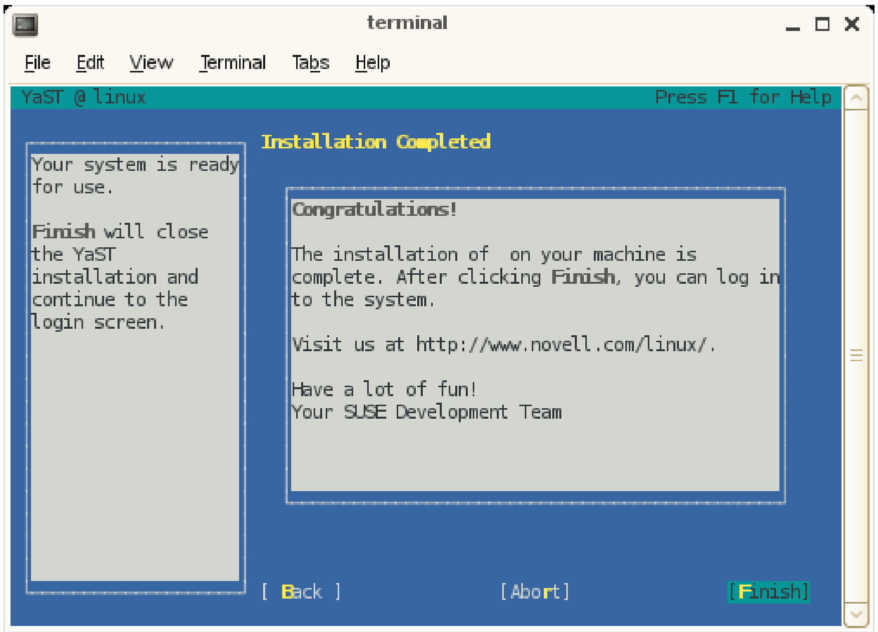

Figure 2-8 Installation Competed Screen

19. 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.

---

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

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

21. The **Cluster Configuration Tool: Initial Configuration Check** screen appears, as shown in Figure 2-9 on page 42. This tool provides instructions on the steps you need to take to configure your cluster. Click **OK** to continue.

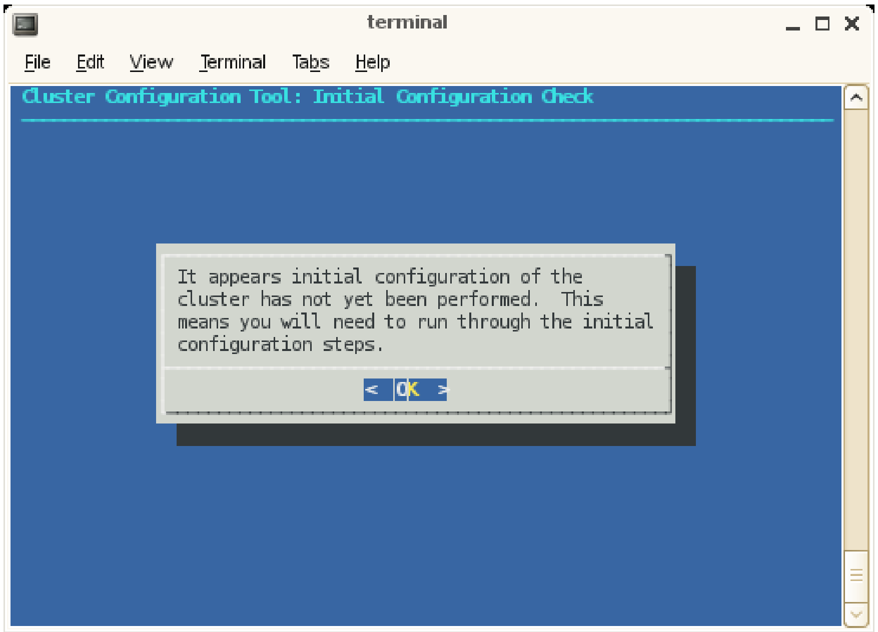

Figure 2-9 Cluster Configuration Tool: Initial Configuration Check Screen

22. The **Cluster Configuration Tool: Initial Cluster Setup** screen appears, as shown in Figure 2-10 on page 43. Read the notice and then click **OK** to continue.

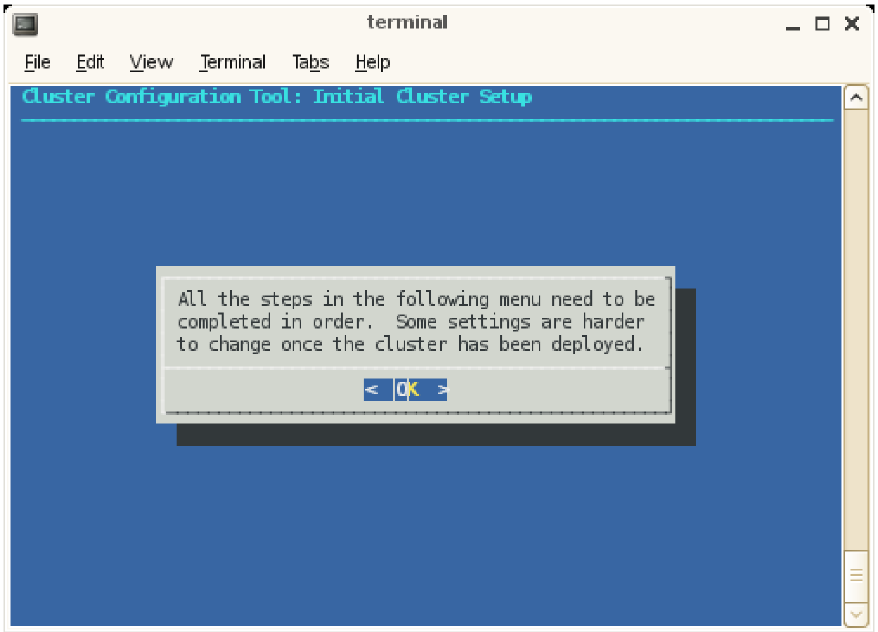

Figure 2-10 Cluster Configuration Tool: Initial Cluster Setup Screen

---

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.

---

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

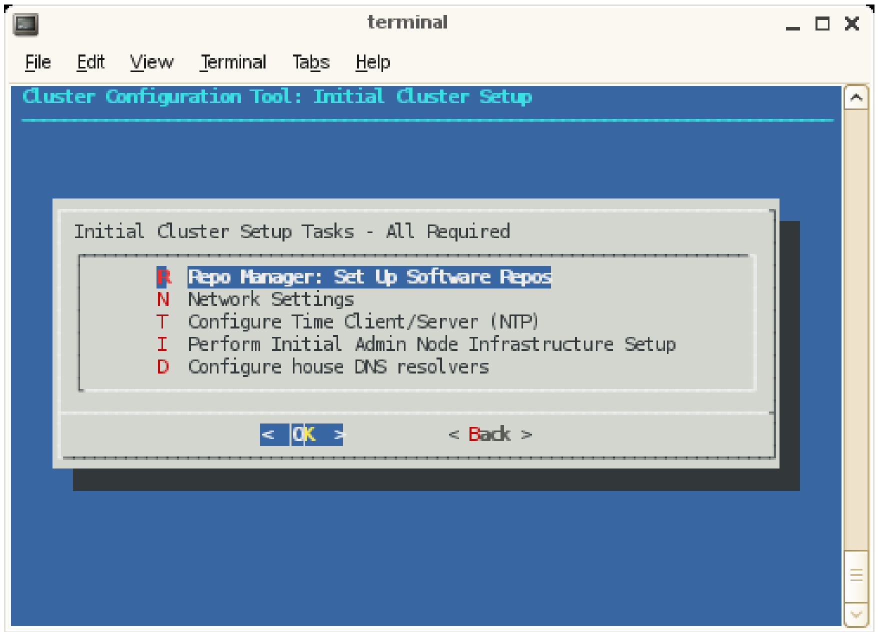

Figure 2-11 Initial Cluster Setup Tasks Screen

24.

Note: The next four screens use the `crepo` command to set up software repositories such as, SGI Foundation, SGI Tempo, SGI ProPack, and SLES10 SP2. For more information, see "`crepo` Command" on page 105.

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

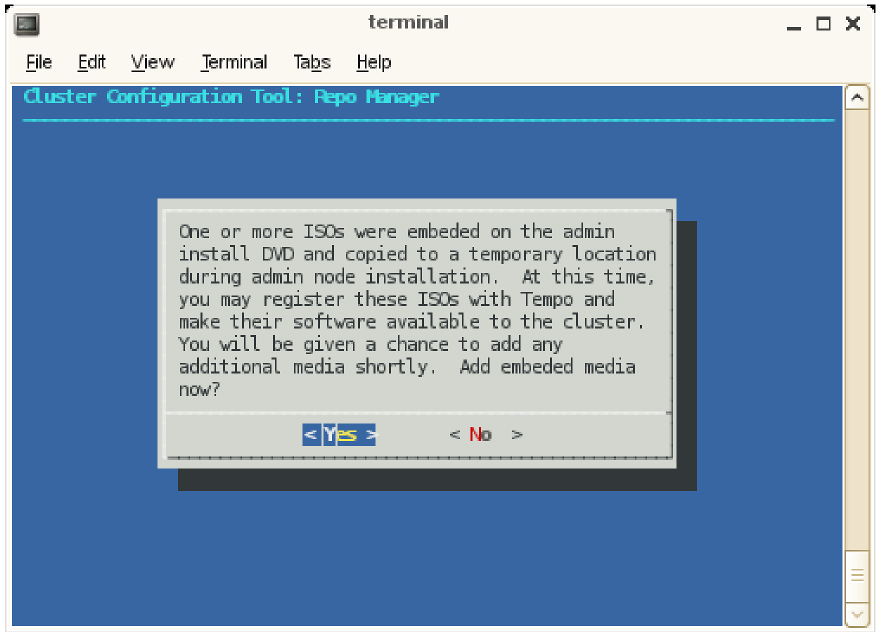

Figure 2-12 Cluster Configuration Tool: Repo Manager Screen One

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

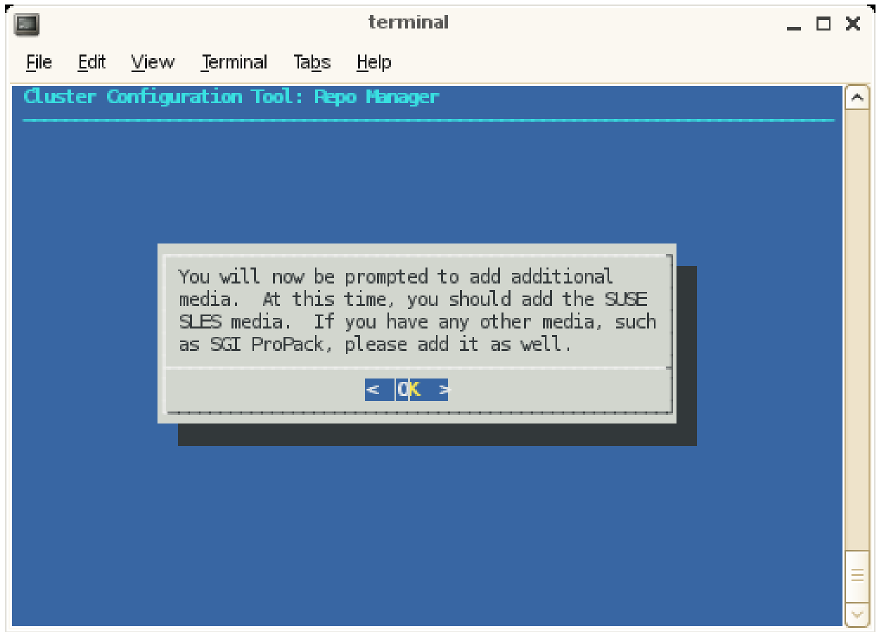

Figure 2-13 Cluster Configuration Tool: Repo Manager Screen Two

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

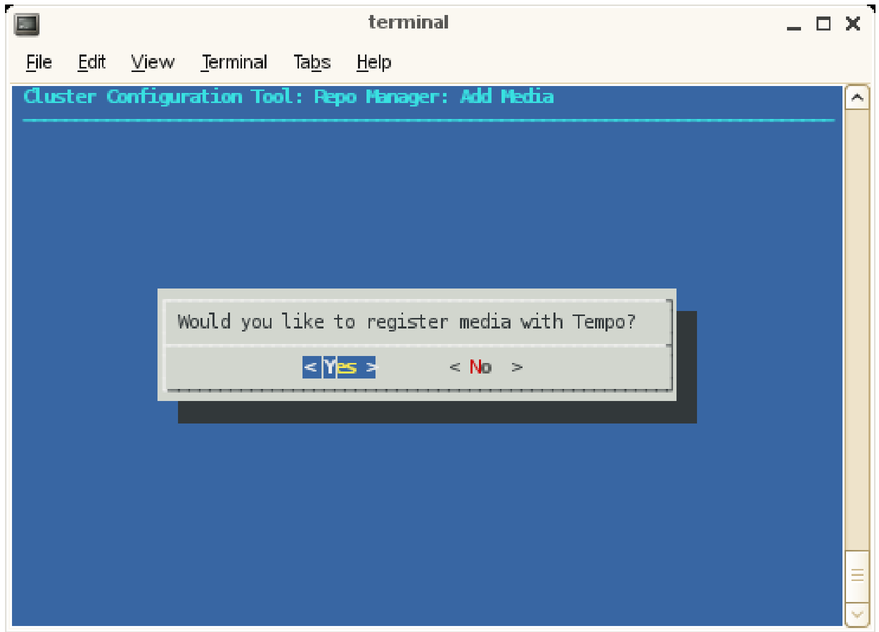

Figure 2-14 Cluster Configuration Tool: Repo Manager Screen Three

27. 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.

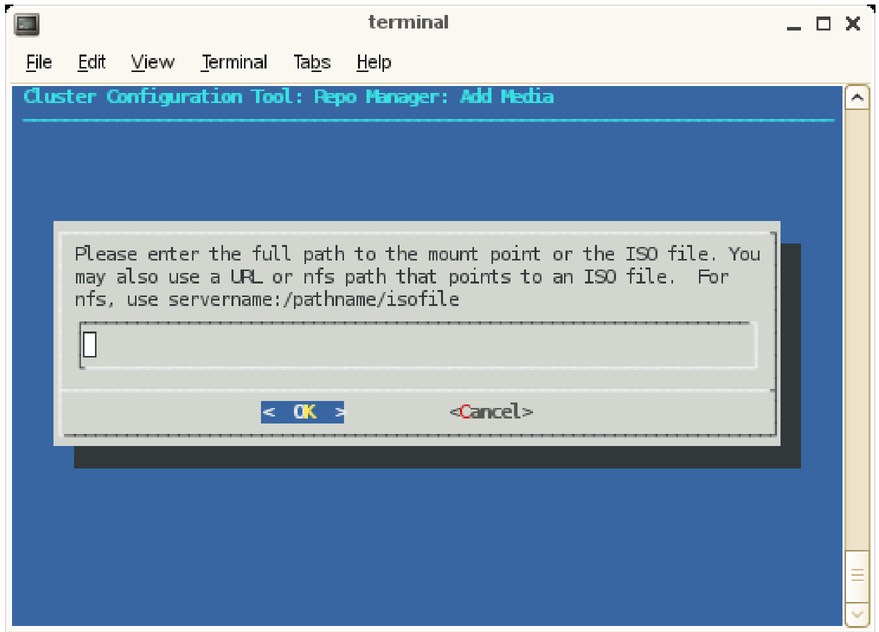

Figure 2-15 Cluster Configuration Tool: Repo Manager Screen Four

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

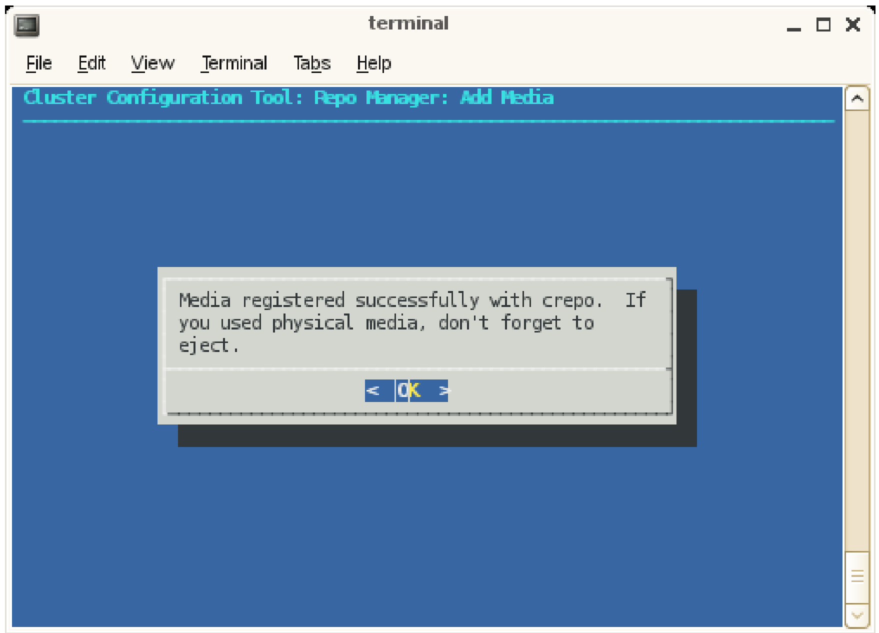

Figure 2-16 Cluster Configuration Tool: Repo Manager: Add Media Screen Four

---

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.

---

29. After choosing the **Network Settings** option, the **Cluster Network Setup** screen appears, as shown in Figure 2-17 on page 50.

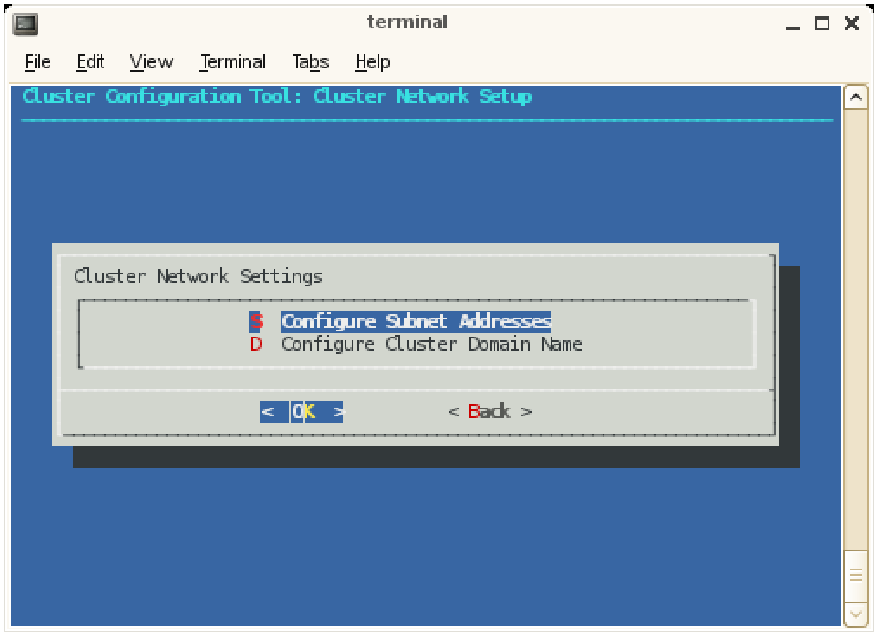

Figure 2-17 Cluster Network Setup Screen

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 2-18 on page 51.

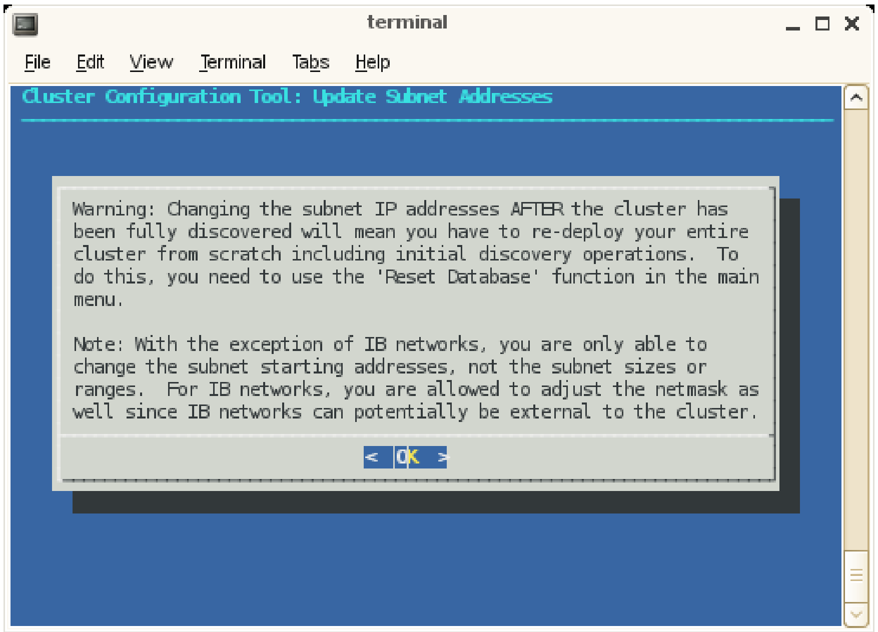

Figure 2-18 Update Subnet Address Warning Screen

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.

30. The **Update Subnet Addresses** screen appears, as shown in Figure 2-19 on page 52.

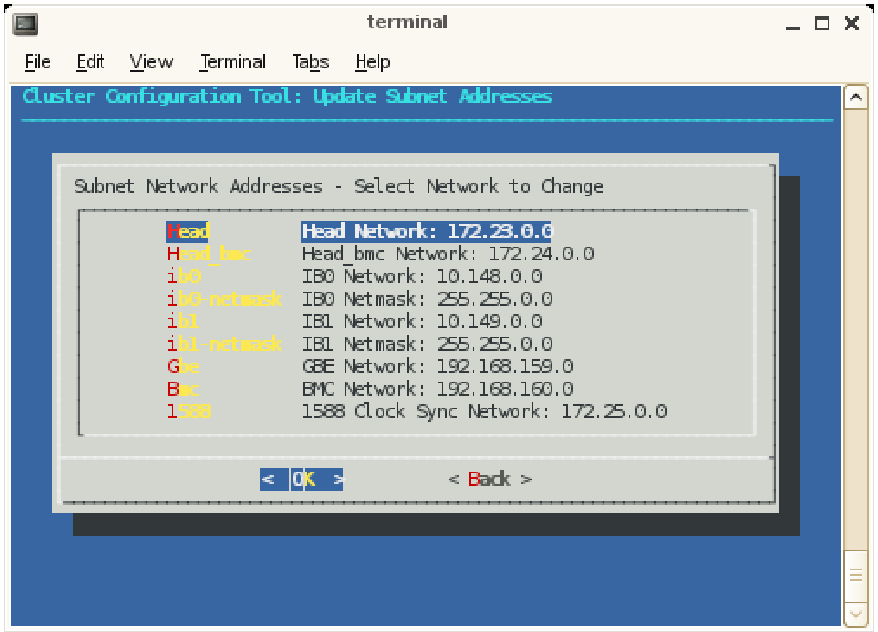

Figure 2-19 Update Subnet Addresses Screen

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.

31. Enter the domain name for your Altix ICE system, as shown in Figure 2-20 on page 53. Click **OK** to continue (this will be a subdomain to your house network, by default).

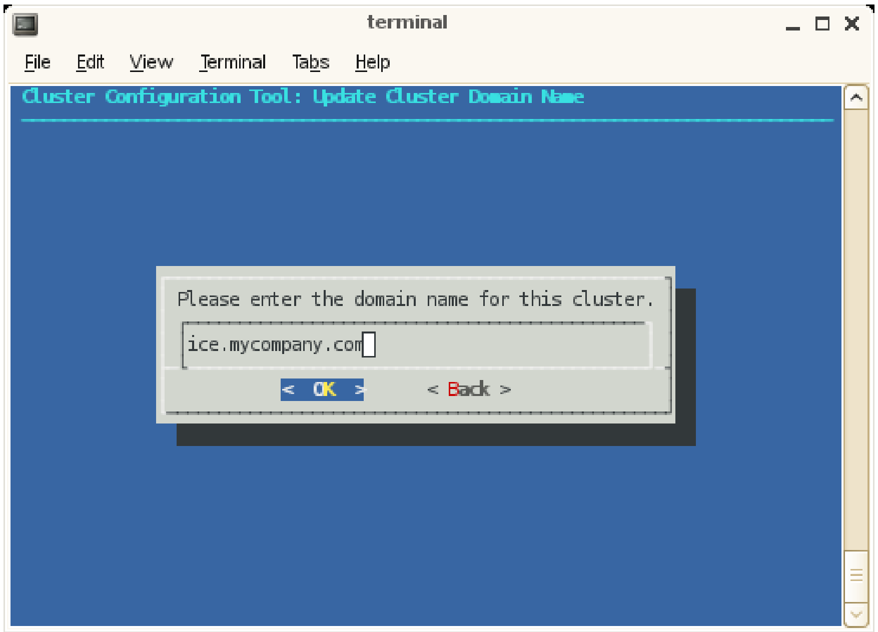

Figure 2-20 **Update Cluster Domain Name** Screen

32. 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.

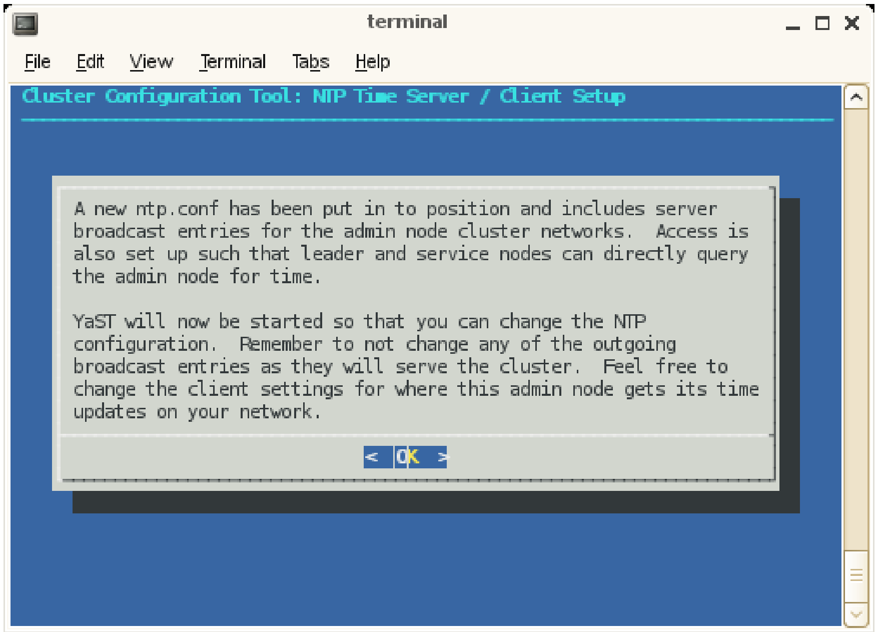

Figure 2-21 NTP Time Server/Client Setup Screen

33. Configure NTP time service as shown in Figure 2-22 on page 55. Click **Next** to continue.

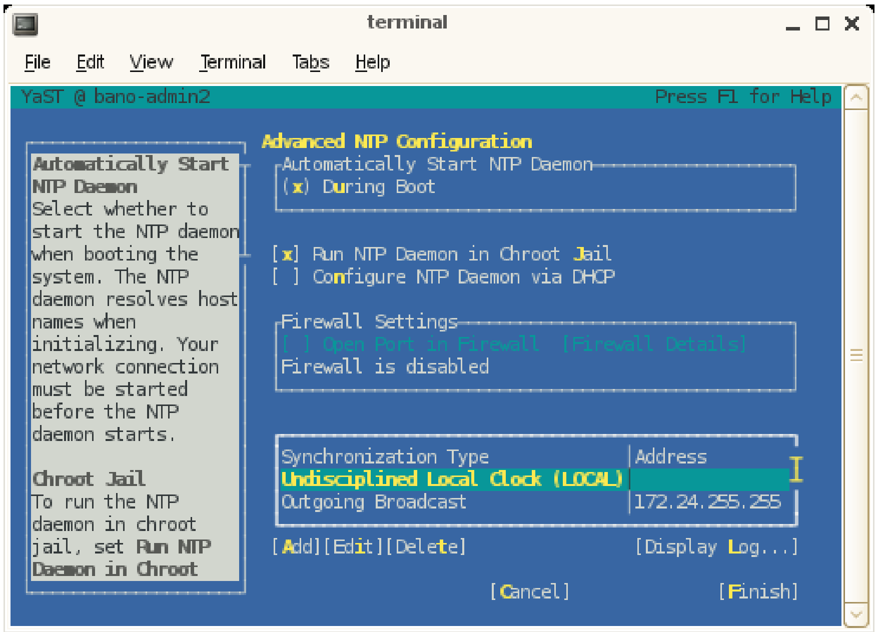

Figure 2-22 Advance NTP Configuration Screen

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

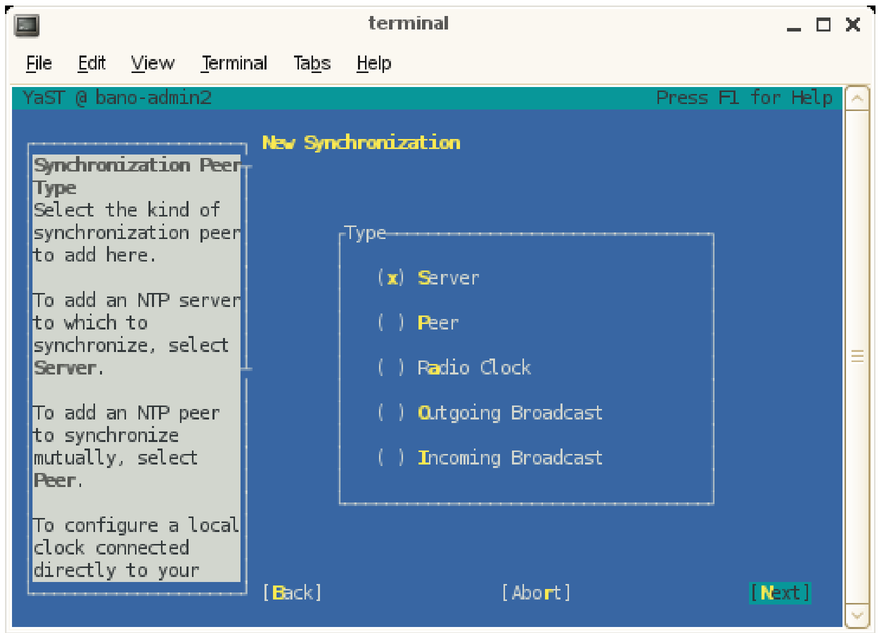

Figure 2-23 New Synchronization Screen

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

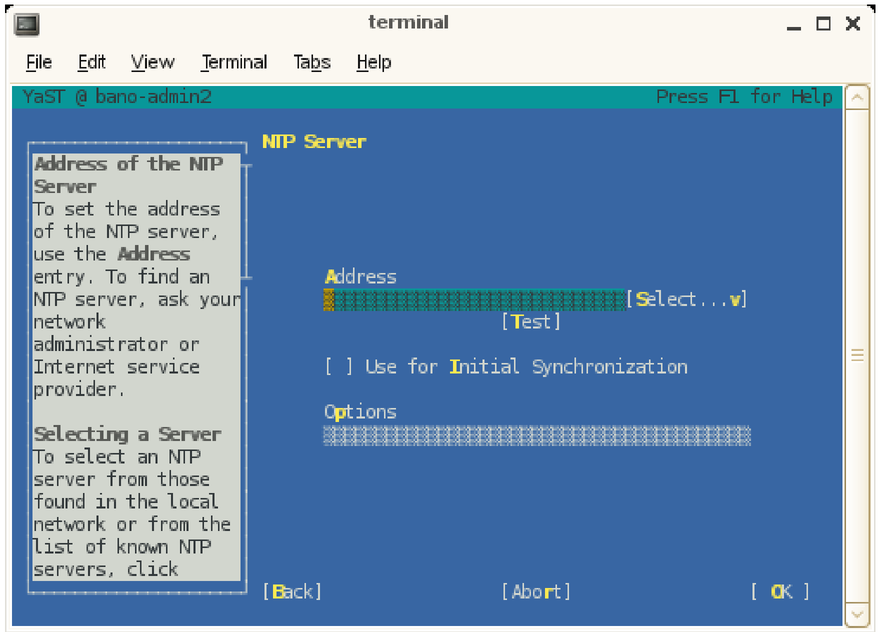

Figure 2-24 NTP Server Screen

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

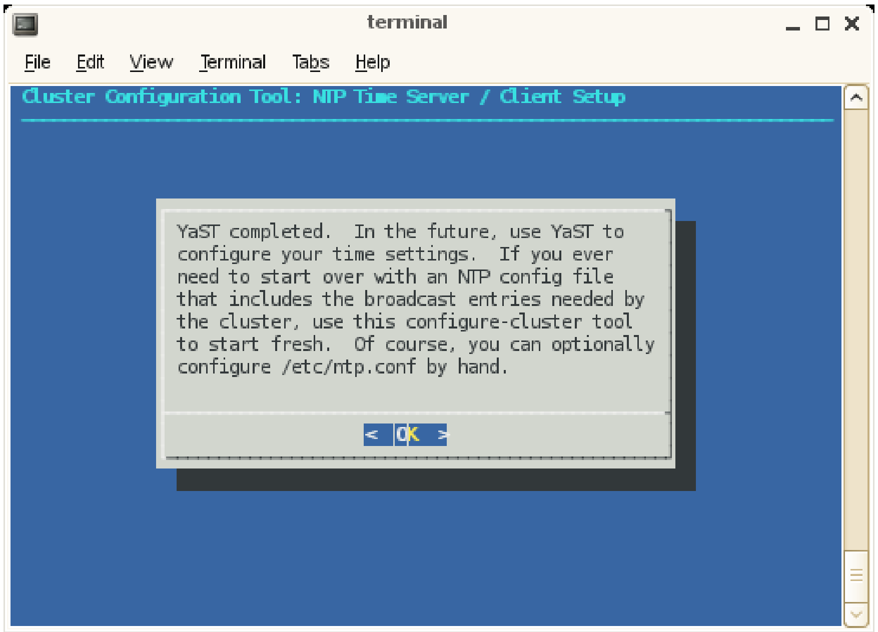

Figure 2-25 NTP Time Server/ Client Setup Screen Three

37. 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.

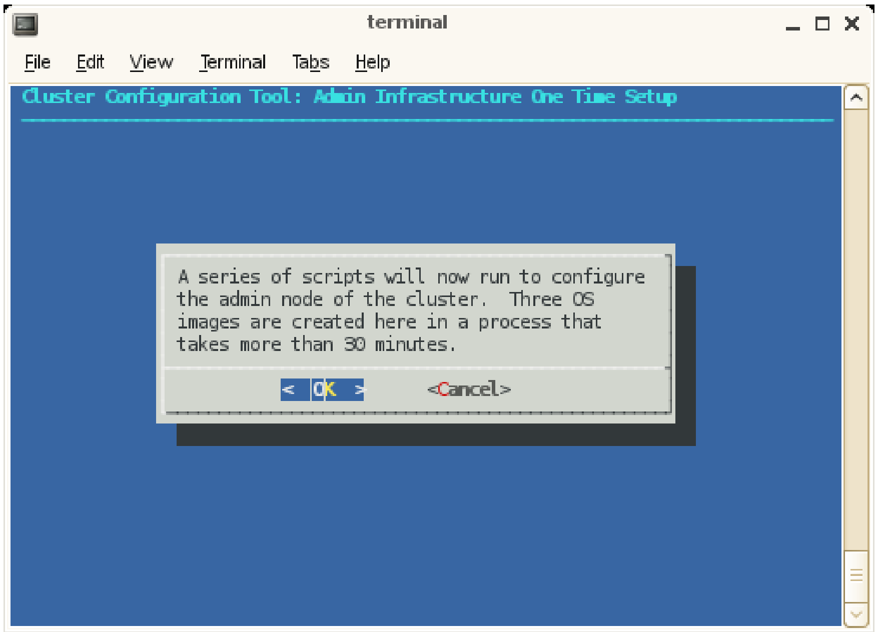

Figure 2-26 Admin Infrastructure One Time Setup Screen One

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 location similar to the following:

```
/tmp/mksiimage-cmds-out.12285
```

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.

38. 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 Installation and Configuration" on page 69. 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).

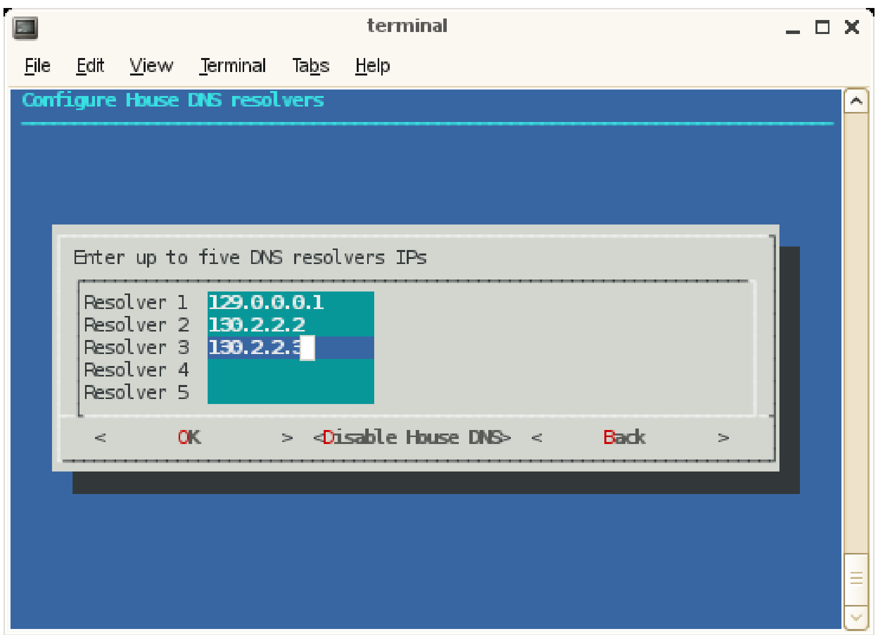

Figure 2-27 Configure House DNS Resolvers Screen

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

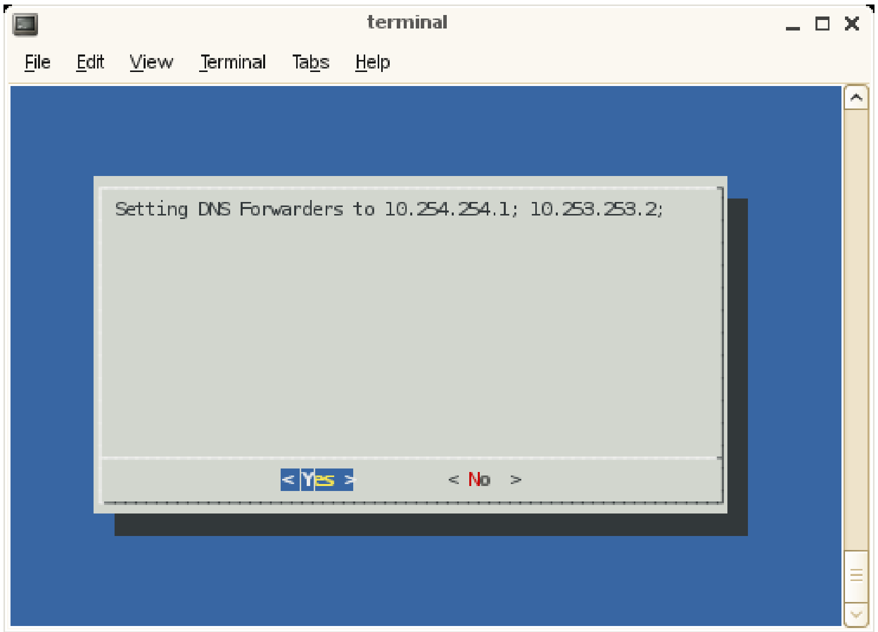

Figure 2-28 Setting DNS Forwarding Screen

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

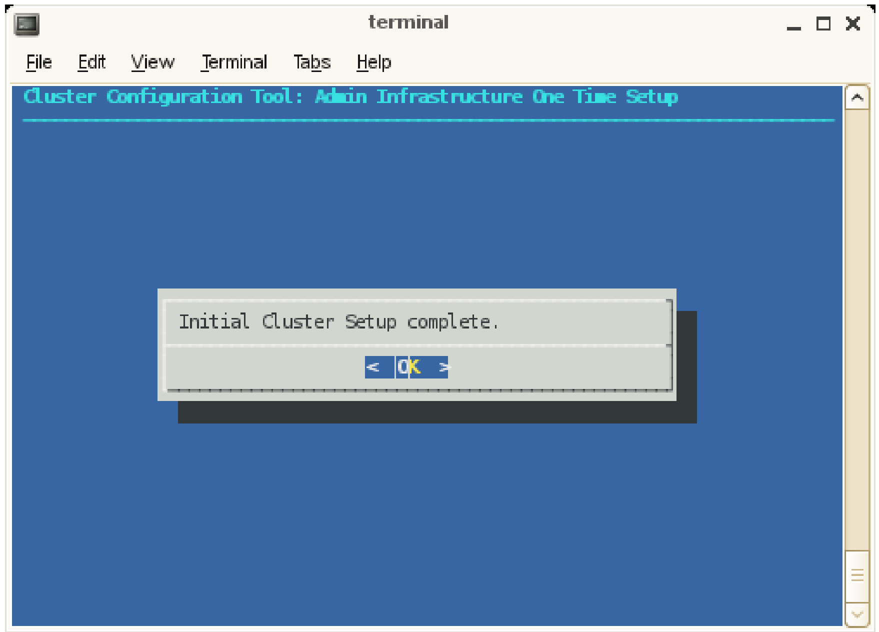

Figure 2-29 Cluster Configuration Tool: Admin Infrastructure One Time Setup Screen

41. Proceed to "Installing Software on the Rack Leader Controllers and Service Nodes" on page 65. 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 the 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 65).

The `discover` command is, as follows:

```
/opt/sgi/sbin/discover --rack <#>[,<hw-type>]
/opt/sgi/sbin/discover --rackset <start-number>,<count>[,<options>]
/opt/sgi/sbin/discover --service <#>[,<options>]
```

The `discover` command accepts the following options:

Option	Description
`--rack`	Discovers a specific rack or set of racks
`--rackset`	Discovers `count` racks starting at `start-number`
`--service`	Discovers the specified service node
`--force`	Use `--force` to avoid sanity checks that require input
`--ignoremac`	Ignores one or more MAC addresses
`--delrack`	Deletes racks and associated leaders and blades
`--delservice`	Deletes a service node
`--help`	Usage and help text

The *options* parameter is 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`".

The *options* paramater include the following:

- *hw-type*

  The *hw-type* parameter is a hardware model that affects how the `discover` command proceeds. If *hw-type* is not specified, a default value is used. Use the *other* hardware type for a service node you supply and manage. This mode allocates IP addresses for you and print them to the screen. This *other* type of service node is **not** managed by the Tempo systems management software.

  Valid hardware type specifiers are, as follows:

  – `ice-csn` (default type)

  – `xe210`

- xe240
- xe310
- altix450 (NAS cube)
- altix4000
- altix4700
- other

- image type

  For service nodes only, you can specify an alternate image to install on to the target system. See the examples for how to specify this.

If you wish to re-discover an existing service node or rack, simply run the discover command in the same manner you normally would. If you wish to purge a rack or service node entirely, (never to be seen again), use --delservice and --delrack options.

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:

```
# /opt/sgi/sbin/discover --rack 1 --service 0,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:

```
# /opt/sgi/sbin/discover --rackset 1,5 --service 0,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-sles10sp2 (default), perform the following:

```
# /opt/sgi/sbin/discover --service 0,image=service-myimage
```

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

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

Installing Software on the Rack Leader Controllers and Service Nodes

The discover command, described in "discover Command" on page 62, 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 2-3 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 142.

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 68, 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 68.

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 theblademond daemon that runs on the leader nodes (see "blademond Command For Automatic Blade Discovery" on page 68).

Procedure 2-4 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 65.
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 157.

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 157.

Service Node 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-image` is used to install the service node.

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 `yast2-firstboot` software does not start automatically on the system console after the first boot after installation (unlike the admin node). This is because the service node installation is a somewhat automated process. A configuration script called `/opt/sgi/sbin/configure-service-node` is available. This script is very simple and simply pops up a couple of dialog windows and then forces `yast2-firstboot` to start up in the current shell session.

The pop-up dialog windows contain information about system management operations 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`.

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 8200 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 2-5 Service Node Configuration or 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. For service node running SUSE Linux Enterprise Server (SLES), there is documentation 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 service 0 # **`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 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 117 and "Customizing Software On Your SGI Altix ICE System" on page 111.

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 2-2 `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.

Service Node Configuration for Gateway Operation

You may chose to connect your compute nodes using resolvable addresses on the house network. This requires planning before the installation by reserving a large block of resolvable IP addresses on the house network and the correct steps early in installation.

Note: Placing a fabric on the house network does make it more susceptible to bandwidth and latency fluctuations due to undesired or unexpected network traffic.

Procedure 2-6 Service Node Configuration for Gateway Operation

To connect your compute nodes using resolvable addresses on the house network, perform the following steps:

1. Enter IP values into the `configure-cluster` script while you make sure to assign IP addresses in the resolvable range to the IB fabric(s) you desire.

   You can make either `ib0`, `ib1`, or both resolvable on the house network. Careful planning is required.

2. After house network addresses are assigned, you need to use the service node(s) operating system tools to enable IP forwarding and configure the house routers or network infrastructure to route addresses for the desired fabrics through the desired service nodes.

   All of these steps are extremely site specific, therefore, you need to rely on your network administrators to set up this type of configuration.

Service Node Configuration for DNS

For information on setting up DNS, see Figure 2-27 on page 60.

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 2-7 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. An example of the `sgi-fstab` file is, as follows:

```
sys-admin:/opt/sgi/share/per-host-customization/global # cat sgi-fstab
#!/bin/sh
#
# 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.  The full path to the per-host iru+slot directory is
# passed in as $1, e.g. /var/lib/sgi/per-host//i2n11.
#

iruslot=$1

cat <${iruslot}/etc/fstab
#               tmpfs           /tmp            tmpfs   defaults        0       0
EOF
```

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 71.

Service Node Configuration for NIS for the House Network

This section describes two different ways to configure a service node for NIS, 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

Assuming the installation has either Network Address Translation (NAT) or Gateway operations configured on one or more service nodes, the compute nodes can directly access the house NIS servers. Broadcast operations for discovering NIS servers do not typically work. Therefore, you need to configure the compute images with the IP address of the NIS server to which you want them to connect.

Procedure 2-8 Service Node Configuration for NIS with the Compute Nodes Directly Accessing the House NIS Infrastructure

To configure NIS on a compute node, perform the following steps:

1. Clone a compute image which you would like to extend to use NIS (see "`cimage` Command" on page 117 and "Customizing Software On Your SGI Altix ICE System" on page 111).

   Note: The default installation does not contain the `ypbind` package. You need to install it for use in your cloned image.

   ---

2. Install the `ypbind` package using the operating system package manager.
3. Use the operating system configuration tools to configure the `ypbind` software. See your operating system documentation for instructions on configuring `ypbind` for NIS operations and the `ypbind`(8) man page.
4. Push this new image out to the compute nodes and reboot the system to test the configuration.
5. If the compute blades fail to connect to the NIS server, use the technique for troubleshooting outlined in "Troubleshooting Service Node Configuration for NAT" on page 71.

Procedure 2-9 NIS with a Service Node as a NIS Slave Server to the House NIS Master

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

1. Make sure your network administrator has authorized the service node to act as a slave server.
2. Use the service node operating system tools to configure the NIS slave server on the service node.
3. Use the `ypwhich`(1) command to verify that it shows `localhost` as the current server and `ypcat`(1) `passwd` looks consistent with what you expect.

Note: You may have some issues with configuration tools, such as, removing parts of the host name or IP for the server. This can be solved by creating a `/etc/hosts` record.

---

4. Install the `ypbind` package using the operating system package manager.
5. Use the operating system configuration tools to configure the `ypbind` software. See your operating system documentation for instructions on configuring `ypbind` for NIS operations and the `ypbind`(8) man page.
6. Push this new image out to the compute nodes and reboot the system to test the configuration.
7. If the compute blades fail to connect to the NIS server, use the technique for troubleshooting outlined in "Troubleshooting Service Node Configuration for NAT" on page 71.

Note: Multiple service nodes can be used as NIS slave servers.

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 73.

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 2-10 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 mklabel msdos
```

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. Make sure the NFS server service is enabled, as follows:

```
# 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 113.

---

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-sles10sp2-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 before the "EOF" line in `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-sles10sp2-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 117.

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 84. 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 76. 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.

Service Node NFS Server Alternate: Re-exporting House NFS Servers

All operations are from the service node acting as the NFS proxy except where noted.

This procedure described in this section does not require the NAT/gateway operations and may be more efficient. This method does require that an unsupported package be installed. It is available from the SGI support page as described below.

Procedure 2-11 Service Node NFS Server Alternate: Re-exporting House NFS Servers

To set up a service node for re-exporting house NFS servers, perform the following steps:

1. Download the unsupported `nfs-server` RPM from the SGI support server:
   a. Login to Supportfolio (https://support.sgi.com/)
   b. Click on **Browse Collections**.
   c. Click on **Download Cool Software**.
   d. Find the `nfs-server` package.
2. Remove `nfs-utils` on the service node, as follows:

   ```
   # rpm -e nfs-utils
   ```
3. Install the newly downloaded `nfs-server` RPM, as follows:

   ```
   # rpm -Uvh /usr/src/packages/RPMS/x86_64/nfs-server-2.2beta51-246*.x86_64.rpm
   ```

4. Edit the `/etc/sysconfig/nfs` file and change the `REEXPORT_NFS` option to "yes"

5. Enable the NFS server at start-up, as follows:

   ```
   # chkconfig nfsserver on
   ```

6. Start it on the service node, as follows:

   ```
   # rcnfsserver start
   ```

7. Add the mount to the "house nfs server" on to the service node acting as the proxy for NFS. An example `fstab` line is, as follows:

   ```
   house-server:/mirror /mirror nfs defaults 0 0
   ```

8. Ensure the filesystem is mounted, as follows:

   ```
   # mount -a
   ```

9. Export the filesystem by adding a line to `/etc/exports` similar to the example. You also need to change the subdomain to match your site’s.

   ```
   /mirror *.ice.americas.sgi.com(ro,sync)
   ```

10. Now configure the compute blades to mount this directory from the service node acting as a proxy. In this example, it is assumed that `service0` is the node from which the blades will mount **`/mirror`**. To do this, add a line similar to this to the following before ’EOF’ in `/opt/sgi/share/per-host-customization/global/sgi-fstab` file. This file is located on the system admin controller (admin node).

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

11. Recall that the mount point for the compute blades needs to exist. Therefore, you might need to create a directory within the systemimager image on the admin node, for example, `mkdir /var/lib/systemimager/images/compute-sles10sp2/mirror`.

12. Tell NFS about the exports change, as follows:

    ```
    # rcnfsserver reload
    ```

13. Earlier, in this procedure, you changed the `sgi-fstab per-host customization` script and created a mount point within one or more compute

blade systemimager images. From the admin node, you need to push the images so they are available on the leader nodes serving your racks. The compute blades in the rack in question should be shut down prior to running this command. You should do this for all compute images you may have and for all racks.

```
# cimage --push-rack compute-sles10sp2 r1
```

14. Now you may boot up your compute blades. The filesystem will now be mounted on each one. When you access `/mirror` on a compute blade, the service node proxy NFS server then accesses its `/mirror`, which contacts the actual NFS server on the house network.

Setting Up a NIS Server for Your Altix ICE System

This section describes how to set up a network information service (NIS) server running SLES10 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 74. 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

- **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 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 SLES10.

Procedure 2-12 Setting Up a Service Node as a NIS master

To set up a 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 92.

---

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 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 SLES10.

Note: You do not do this on the NIS Master service node that you already configured as a client in "Setting Up a Service Node as a NIS Master" on page 85.

Procedure 2-13 Setting Up a 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 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 127.0.0.1 > /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 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 113.
2. Change the compute nodes to use the cloned image/kernel pair, as follows:

```
admin:~ # cimage --set compute-sles10sp2-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-sles10sp2-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-sles10sp2-clone/etc/yp.conf
```

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

```
admin:~# chroot /var/lib/systemimager/images/compute-sles10sp2-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-sles10sp2-clone/etc/group
admin:~# echo "+::::::" >> /var/lib/systemimager/images/compute-sles10sp2-clone/etc/passwd
admin:~# echo "+" >> /var/lib/systemimager/images/compute-sles10sp2-clone/etc/shadow
```

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

```
admin:~ # cimage --push-rack compute-sles10sp2-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 Service Node as a NIS Client" on page 86.

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.

Overview

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 the `/tftpboot/oscar` directory and SUSE Linux Enterprise Linux (SLES) packages are in the `/tftpboot/distro/sles-10-x86_64` directory.

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 `yum`(8) command. The `yum update` and `yum install` commands are used for all updates, including updates to existing images.

For additional information on updating your system, see "Upgrading from SGI ProPack 5 SP5 to SGI ProPack 6" on page 101.

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 then installed with the `yum` command.

This following sections describe these steps:

• "Update the Local Package Repositories on the Admin Node" on page 93

• "Installing Updates on Running Admin, Leader, and Service Nodes " on page 96

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.

If you installed and configured the yup tool as described in "Update the SLES Package Repository" on page 94, 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. If you fail to do this, the yum/yume/cinstallman command may not be able to see your new packages.

Update the SLES Package Repository

As described in "Update the SGI Package Repositories on the Admin Node" on page 93, it is possible to download updates for SUSE Linux Enterprise Server to the local SLES package repository on the admin node. Tools like YaST Online Update and rug(1) are designed to update a running system, but not well suited to managing a repository of packages for use within a clustered environment.

You can use the yup(1) tool to mirror the update packages from Novell's update servers.

The yup tool stores packages downloaded from the Novell update server in a directory structure that is not compatible with the local SGI Tempo, SGI ProPack and SLES package repositories located on the admin node. After yup is run, the packages need to be copied to the appropriate SLES repository.

If you plan to use the SGI-supplied /opt/sgi/sbin/sync-repo-updates script to keep repositories up to date, you will see that it copies the packages retrieved via yup in to the appropriate SLES package repository and then updates the repository metadata. If you plan to use your own customized scripts, please use the sync-repo-updates script, as an example.

Before you can download packages via yup, you must register the admin node with Novell. The following steps explain how to register the admin node with Novell using YaST, although the general steps are the same for graphical yast2:

1. Run the `/sbin/yast` command and select **Software -> Novell Customer Center Configuration**
2. You will be prompted to enter your email address and registration code, if you have it.

Note: If you do not have your registration code(s) handy, you may proceed and Novell will provide you with temporary access; however, you will need to register your SLES purchase in the Novell Customer Center (http://www.novell.com/center) in order to continue to get access to updates. If you already have a Novell account, use the email address associated with your Novell account here.

Sometimes the registration process will error out with a "failed to contact the server" message. This is due to an issue with unknown/untrusted key error caused by the manner in which the initial admin node images are created. Repeat step 3, you will not have to fill in the field forms again, and then `yast` will prompt you to import the keys. Once the keys are imported, you should be able to complete this step.

---

3. As part of the registration process, `yast` will add an install source for updates. DO NOT TRY TO USE YaST ONLINE UPDATE. At the end of the Novell Customer Center Configuration module, `yast` will display information on how to create a Novell account. If you do not already have a Novell account, go create one per the instructions.

Now that you have created a Novell account and registered your admin node with Novell, you can configure `yup` to mirror the SLES update packages. The following instructions cover only the essential steps required to mirror the SLES updates; the `yup` man pages have more detailed information on available options:

1. Edit `/etc/sysconfig/yup` file and modify the following variables:
   - `YUP_DEST_DIR`

     Set this variable to `/var/spool/yup-updates`. This value is referenced by the `/opt/sgi/sbin/sync-repo-updates` script.
   - `YUP_ID`

     Paste the contents of the `/etc/zmd/deviceid` file into this variable.
   - `YUP_PASS`

Paste the contents of the /etc/zmd/secret file into this variable.

- YUP_ARCH

  Set this variable to x86_64.

- YUP_SUBVERSIONS

  Set this variable to SP2.

2. Create the yup directory referenced above with the following command:

```
# mkdir /var/spool/yup-updates
```

   The yup command downloads packages to the /var/spool/yup-updates directory. The sync-repo-updates script moves the SLES packages from that directory into the /tftpboot/distro/sles-10-x86_64 directory so that the local SLES package repository on the admin node contains the updated packages.

3. Execute the /opt/sgi/sbin/sync-repo-updates script or run yup manuall.

   The sync-repo-updates script should now be able to download SLES updates directly from the update server of Novell and update the local SLES package repository on the admin node so that you can use the yum command to install or update SLES packages throughout the node.

Note: The variables in /etc/sysconfig/yup are important because by default you will only have access to the SLES updates for x86_64 by default. Setting YUP_ARCH or YUP_SUBVERSIONS variables incorrectly can lead to 403/permission denied errors. The yup may also error out when accessing SLES10-SP2-Online files; it is safe to ignore that error.

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.
- When using a node aggregation, for examle, the asterisk (*), 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 105 and "`cinstallman` Command" on page 109, 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 section "5.3 Additional Steps for Compute Image Kernel Updates" 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-sles10sp2
admin:~ # cinstallman --update-image --image service-sles10sp2
admin:~ # cinstallman --update-image --image compute-sles10sp2
```

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:

```
# cinstallman --yum-image --image compute-sles10sp2 clean all
```

Update Packages on Running Leader and Managed Service Nodes

It is possible to update live images running on leader and managed service nodes using the `yum` command.

Note: Changes made to live node images are not reflected in the node images in the `/var/lib/systemimager/images/{image}-sles10sp2` image directories. This means that changes made to live images may not get carried forward the next time you add or re-discover leader and/or service nodes.

The following examples show how to update a live leader node and/or managed service node using the `yum` command:

```
# ssh r1lead yum update   (update live leader node)
# ssh service0 yum update      (update live service node)
```

Update Packages Inside Images

SGI provides the `yum-image-wrapper` script, which makes using the `yum` command inside an image directory fairly straight forward. You can use the script directly, or use the script as a template for a more customized solution.

Note: Changes to the kernel package inside the compute image directory require some additional steps before the new kernel can be used on compute nodes (see "Additional Steps for Compute Image Kernel Updates" on page 100section for more details). This note does not apply to leader or managed service nodes.

The following examples show how to upgrade the packages inside three node images supplied by SGI:

```
admin:~ # yum-image-wrapper /var/lib/systemimager/images/lead-sles10sp2 update
admin:~ # yum-image-wrapper /var/lib/systemimager/images/service-sles10sp2 update
admin:~ # yum-image-wrapper /var/lib/systemimager/images/compute-sles10sp2 update
```

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 a leader or service node, it will already contain the updated packages.

Note: Before pushing the compute image to the leaders using the `cimage` command, it is good idea to clean the `yum` cache. This 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 leaders.

To clean the `yum` cache, from the system admin controller (admin node), perform the following:

```
admin:~ # yum-image-wrapper /var/lib/systemimager/images/compute-sles10sp2 clean all
```

It is also possible to use standard `rpm`(8) commands to update images. This can be quite cumbersome depending on the number of images you need to update and the number of packages you intend to update. The steps are as follows:

1. Copy the update packages you want to use in to the image directory in `/var/lib/systemimager/images/{image}-sles10sp2`.
2. Use the `chroot` command to start a shell within the root tree.
3. Install/update packages using standard `rpm` commands; certain packages require `/sys` to be mounted in the `chroot`, so you may have to mount `/sys` in order to work around any failures.

In general, SGI recommends using the `yum-image-wrapper` script to update the node images in the image directory.

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-sles10sp2` and that you have already updated the compute node image in the image directory per the instructions in "Update Packages Inside Images" on page 99. If you have named your compute node image something other than `compute-sles10sp2`, replace this in the example that follows:

1. Shut down any compute nodes that are running the `compute-sles10sp2` image (see "Power Management Commands" on page 128).
2. Push out the changes with the `cimage --push-rack` command, as follows:

   ```
   # cimage --push-rack compute-sles10sp2 r\*
   ```

3. Update the database to reflect the new kernel in the `compute-sles10sp2`, as follows:

   ```
   # cimage --update-db compute-sles10sp2
   ```

4. Verify the available kernel versions and select one to associate with the `compute-sles10sp2` image, as follows:

   ```
   # cimage --list-images
   ```

5. Associate the compute nodes with the new kernel/image pairing, as follows:

   ```
   # cimage --set compute-sles10sp2 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 SGI ProPack 5 SP5 to SGI ProPack 6

For information on upgrading your system from SGI ProPack 5 Service Pack 5 to SGI ProPack 6 , see the release notes. The SGI ProPack 6 release notes can be found in a file named `README.TXT` that is available in `/docs` directory on the SGI ProPack 6 for Linux CD.

The SGI ProPack 6 for Linux release notes get installed to the following location on a system running SGI ProPack 6:
`/usr/share/doc/packages/sgi-propack-6/README.txt`

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 SLES10 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

Currently, the compute nodes run the SUSE Linux Enterprise Server 10 (SLES10) Service Pack 1 (sp2) Linux distribution. To improve the performance of applications running MPI jobs on compute nodes, the following SLES10 services are turned off:

- acpid
- auditd
- boot.crypto
- boot.device-mapper
- boot.lvm
- boot.md
- cron
- earlykbd
- earlysyslog
- fbset

- irq_balancer
- kbd
- novell-zmd
- nscd
- postfix
- powersaved
- resmgr
- slpd
- splash
- splash_early
- suseRegister
- xdm

crepo Command

You can use the crepo command to manage software repositories, such as, SGI Foundation, SGI Tempo, SGI ProPack, and SLES10 SP2. 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 v1.4 release, the repositories are, as follows:

Repository	Description
/tftpboot/sgi/*	For SGI media
/tftpboot/other/*	For any YaST-style media that is not from SGI

`/tftpboot/distro/sles10sp2`

For SUSE Linux Enterprise Server 10 (SLES10SP2)

`/tftpboot/`*x*

Customer-supplied repositories

The directory and repository names are determined automatically for YaST compatible media including media supplied by SGI and SLES10SP2. 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. `crepo` 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 `sles10sp2` RPMs that SGI recommends. The `compute-sles10sp2.rpmlist`, `lead-sles10sp2.rpmlist`, and `service-sles10sp2.rpmlist` files are the ones that `crepo` created based on the `distro` RPM lists in this directory plus any suggested packages from any media added with the `--add` operation.

```
system-admin:/etc/opt/sgi/rpmlists # ls
compute-distro-sles10sp2.rpmlist  lead-sles10sp2.rpmlist
compute-sles10sp2.rpmlist         service-distro-sles10sp2.rpmlist
lead-distro-sles10sp2.rpmlist     service-sles10sp2.rpmlist
```

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

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

cimage --update-db mynewimagecinstallman 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 (calls mksiimage -L)

--update-image          : update pkgs in image to latest pkgs avail 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 args 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 args 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
                          reintall/reimage?  Requires --node
```

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)

With the SGI Tempo 1.4 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:

```
system-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
```

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-sles10sp2.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 describe 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 92.

Compute Node Per-Host Customizations

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.

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-hostname` is, as follows:

```
#!/bin/sh
  #
  # Copyright (c) 2007 Silicon Graphics, Inc.
  # All rights reserved.
  #
  # Set the compute node's hostname to the cluster unique name
  #
  # 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//i2n11.
  #

  # sanity checks
  . /opt/sgi/share/per-host-customization/global/sanity.sh

  iruslot=$1

  # source cluster management information
  . ${iruslot}/etc/opt/sgi/cminfo

  # set hostname of blade to cluster unique name
  echo ${NAME} > ${iruslot}/etc/HOSTNAME
```

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 to SGI ProPack 5 SP3 if you were running SGI ProPack 5 SP2 previously). 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 Clone a Compute Node Image

To clone a compute node image, perform the following steps:

1. From the system admin controller, create a clone of the compute node image, as follows:

   ```
   # cinstallman --create-image --clone --source compute-sles10sp2 --image new
   ```

   After that command is complete, you will have a new image located in `/var/lib/systemimager/images/new` on the system admin controller.

2. To see that the image is now available, perform the following command:

   ```
    # cimage --list-images
   image: compute-sles10sp2
   kernel: 2.6.16.60-0.17-smp

   image: new
   kernel: 2.6.16.60-0.17-smp
   ```

For RPM lists, the default RPM lists are located in `/etc/opt/sgi/rpmlists` on the system admin controller. SGI suggests you never change these files, but rather, create your own versions using the ones supplied by SGI as a base.

Please note, it is important that certain packages be in the `rpmlist` for a given node. For example, an `rpmlist` used for compute nodes should have packages `sgi-compute-node` and `sgi-cluster`. Service nodes must have `sgi-service-node` and `sgi-cluster`.

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 109.

---

1. Make a clone of the compute node image, as described in "Customizing Software Images" on page 113.

2. Determine what images and kernels you have available now, as follows:

```
# cimage --list-images
  image: compute-sles10sp2
         kernel: 2.6.16.60-0.17-smp

  image: compute-sles10sp2-new
         kernel: 2.6.16.60-0.17-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:

```
# cp /newrpm.rpm new/tmp
```

5. The new RPMs now reside in `/tmp` direcory in the image named `new`. To install them into your new compute node image, perform the following commands:

```
# chroot new bash
```

   And then perform the following:

```
# rpm -Uvh /tmp/newrpm.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 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 3-3, page 116.

Procedure 3-3 Creating a Simple Compute Node Image Clone

Note: Always work from a clone image, see "Customizing Software Images" on page 113.

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-sles10sp2 --image compute-sles10sp2-new
```

2. To see the images and kernels in the list, perform the following:

```
# cimage --list-images
image: compute-sles10sp2
       kernel: 2.6.16.60-0.17-smp

image: compute-sles10sp2-new
       kernel: 2.6.16.60-0.17-smp
```

3. To change the compute nodes to use the cloned image/kernel pair, perform the following:

```
# cimage --set compute-sles10sp2-new 2.6.16.60-0.17-smp "r*i*n*"
```

Procedure 3-4 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 109.

---

1. Use the `cinstallman` command to create your own version of the service node image. See "`cinstallman` Command" on page 109.

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 /newrpm.rpm my-service-image/tmp
   ```

4. The new RPMs now reside in `/tmp` direcory in the image named `my-service-image`. To install them into your new compute node image, perform the following commands:

   ```
   # chroot new 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 re-installed with the image. If you wish to install the package on running systems, you can copy the `rpm` to the running system and use `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:

```
sys-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).
 --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-sles10sp2
       kernel: 2.6.16.60-0.17-smp
```

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-sles10sp2 2.6.16.60-0.17-smp
r1i0n1: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n2: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n3: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n4: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n5: compute-sles10sp2 2.6.16.60-0.17-smp
```

```
r1i0n6: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n7: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n8: compute-sles10sp2 2.6.16.60-0.17-smp
[...snip...]
```

To set the `r1i0n0` compute node to boot the `2.6.16.60-0.17-smp` kernel from the `compute-sles10sp2` image, perform the following: :

```
# cimage --set compute-sles10sp2 2.6.16.60-0.17-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-sles10sp2 2.6.16.60-0.17-smp
r1i0n1: compute-sles10sp2 2.6.16.60-0.17-smp
r1i0n2: compute-sles10sp2 2.6.16.60-0.17-smp
[...snip...]
```

To set all nodes in all racks to boot the `2.6.16.60-0.17-smp` kernel from the `compute-sles10sp2` image, perform the following:

```
# cimage --set compute-sles10sp2 2.6.16.60-0.17-smp r*i*n*
```

To set two ranges of nodes to boot the 2.6.16.60-0.17-smp kernel, perform the following:

```
# cimage --set compute-sles10sp2 2.6.16.60-0.17-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the compute-sles10sp2 image to a new image (so that you can modify it) , perform the following:

```
# cinstallman --create-image --clone --source compute-sles10sp2 --image mynewimage
Cloning compute-sles10sp2 to mynewimage ... done
```

The clone process adds the image and its kernels to the database

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//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-sles10sp2
        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, SLES10 SP2, 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.

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-sles10sp2 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 117.
- 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 109.

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-sles10sp2.rpmlist (see "crepo Command" on page 105). 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.

Procedure 3-5 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-sles10sp2.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.

   This command may take a long time and has a large amount of output. The output is automatically directed to a temporary file which you can review and delete as you wish. The command gives you the location of the output file when it runs.

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 3-6 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-sles10sp2.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.

   This command may take a long time and has a large amount of output. The output is automatically directed to a temporary file which you can review and delete as you wish. The command gives you the location of the output file when it runs.

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 117.

Installing a Service Node with a Non-default Image

After you have updated or created a service node image, you can install that image on to a managed service node, such as a login node.

Note: Re-installing the service node using the `discover` process will destroy everything previously on the root drive.

By default, `discover` uses the SGI default `service-sles10sp2` image. For example:

```
# discover --service 2,image=my-service-node-image
```

The image above directs the installation of the described operating system image.

For more information on the `discover` command, see "`discover` Command" on page 62.

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 105).

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 3-7 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 121).

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 117.

- 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.

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.
- There is also a SUSE configuration script in `/sbin/conf.d`, called `SuSEconfig.00cluster-configuration`, that calls the framework. This is in case of you are using YaST to install or upgrade packages.
- 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.

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).
`--ipmi`	Uses `ipmitool` to communicate. [default]
`--ssh`	Uses `ssh` to communicate.
`--intelplus`	Uses the `-o intelplus` option for `ipmitool` [default] Note that you do not usually need to specify this.
`--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

Note: The command will fail if the target contains any wild cards, unless the `--all` option is specified.

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 defaultd 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*
```

The `cpower` command also

This example gives you the power status and boot status of all the compute blades in rack 1. This command is equivalent to **`cpower --node --status r1i*`**.

This command issues an `ipmitool` `power off` command to all of the nodes specified by the wildcard, as follows:

```
# cpower --off r2i*
```

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***
- # **cpower --reset r1i***
- # **cpower --cycle r1i***
- # **cpower --identify 5 r1i***

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 r1*
```

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 it's 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 3-2 `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

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`, `power-iru`, `power-rack`, and `power-system` commands rather than `cshutdown` (see "Power Management Commands" on page 128).

EXAMPLES

Example 3-3 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 employee.

Example 3-4 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>`

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 --help
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 [1]
--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
```

```
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
```

EXAMPLES

Example 3-5 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
```

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:

  ```
  system-admin # /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 3-8 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:

```
system-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:

```
system-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:

```
system-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 144
- "Rack Leader Controller NTP" on page 144
- "Managed Service, Compute, and Leader BMC Setup with NTP" on page 145
- "Service Node NTP" on page 145
- "Compute Node NTP" on page 145
- "NTP Work Arounds" on page 145

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 62) 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 3-9 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, as shown in the example below:

```
#!/bin/sh
#
# Copyright (c) 2007 Silicon Graphics, Inc.
# All rights reserved.
#
# 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//i2n11.
```

```
#

# sanity checks
. /opt/sgi/share/per-host-customization/global/sanity.sh

iruslot=$1

cat <${iruslot}/etc/fstab
#               tmpfs           /tmp            tmpfs   size=48m        0       0
EOF
```

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 117.

Disabling Swap Space

This section describes how to disable swap space on your Altix ICE system.

Procedure 3-10 Disabling Swap Space

To disable swap space, from the admin node, perform the following steps:

1. Turn off swapping, as follows:

```
admin:~# chroot /var/lib/systemimager/images/compute-sles10sp2 chkconfig iscsiswap off
```

2. Push the new image out to the compute nodes, as follows:

   ```
   admin:~# cimage --push-rack compute-sles10sp2 r\*
   ```

3. Power on or reboot the compute nodes (see "Shutting Down and Booting" on page 132).

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 132).
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 132).
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-sles10sp2 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 113.

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-sles10sp2 2.6.16.60-0.21-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-sles10sp2 2.6.16.60-0.21-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-sles10sp2 2.6.16.60-0.21-smp r1i0n0
```

4. You can change the view the change back to NFS root, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles10sp2 2.6.16.60-0.21-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 systems with the 1068-based controller. They do not apply to Altix XE250 systems that have the LSI Megaraid controller.

Example 3-6 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. 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 8200 series systems uses Open Fabrics Enterprise Distribution (OFED) software. This section describes the InfiniBand fabric and how to manage it. For background information on OFED, see http://www.openfabrics.org.

InfiniBand Fabric Management

This section describes the InfiniBand fabric and covers the following topics:

- "InfiniBand Fabric Overview" on page 157
- "InfiniBand Fabric Administrative Tools" on page 158
- "InfiniBand Fabric Management Configuration and Operation Overview" on page 163
- "Useful Utilities and Diagnostics" on page 173

InfiniBand Fabric Overview

Fabric management on SGI Altix ICE 8200 series systems uses the OFED OpenSM software package. 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. The InfiniBand network has two separate network fabrics, `ib0` and `ib1` (see "InfiniBand Fabric" on page 21) with the following characteristics:

- Each network fabric has its own subnet manager (SM).
- For a system with two racks or more, one rack leader controller (leader node) runs an instance of SM to manage the `ib0` fabric and a second leader node runs an instance of SM to manage the `ib1` fabric. A database on the admin node keeps a record of which rack leader nodes are running the fabric management software for either `ib0` or `ib1`, respectively. The `smadmin` command has the logic to place `opensm` on the appropriate rack leader controller. If one of the rack leader controllers becomes unavailable, management of fabric can be assigned to another available rack leader node in the system.

- On a system with a single rack, both instances of opensm run on the same rack leader node.
- Each instance of SM on the rack leader controller is controlled by the /etc/ofa/opensm-ib[01].conf configuration file. For more information, see "smconfig Automatic Fabric Configuration Tool" on page 159.
- Rack leader controllers run the opensm daemon for each fabric over separate HCA ports (see Figure 1-9 on page 22).

Note: For this release, after a system reboot, you need to manually restart the opensm daemons running on the InfiniBand fabric. If the opensm daemons are allowed to start automatically, as the leader nodes boot, you will not know which leader is the Master and it is highly likely that the fabric will NOT be routed correctly. After a system reboot, use the smadmin command to restart the fabric. For more information, see "smadmin InfiniBand Fabric Administration Tool" on page 160 and "Fabric Management and Rebooting" on page 163.

- Each fabric is addressed by a global unique identifier (GUID) and unique HCA port.

  The GUID and HCA port is set in the configuration file.

Note: Currently, the InfiniBand fabric ib0 is reserved for MPI and the InfiniBand fabric ib1 is reserved for storage.

InfiniBand Fabric Administrative Tools

The InfiniBand fabric is not started automatically on your Altix ICE system because if the fabric is started too early when the system is being discovered and installed, the InfiniBand fabric will not be discovered correctly. This section describes how to configure and administer you InfiniBand fabric and covers these topics:

smconfig Automatic Fabric Configuration Tool

SGI Tempo provides the smconfig tool that automatically configures the fabric for you. "Configuring and Initializing the InfiniBand Fabric Manually" on page 170 describes how to manually configure a fabric and provides more detailed information on how fabric configuration works.

The smconfig command is, as follows:

```
/opt/sgi/sbin/smconfig
```

It accepts the following options:

Option	Description
-f [ib0 or ib1]	Selects fabric ib0 or ib1 (Required)
-r [dor updn]	Use dor variable for hypercube topology. Use updn variable for fat-tree topology. (Required) For more information, see "Network Topology" on page 168.
-o [rack lead IP's]	OSM hosts list (overrides the default of autoconfigure)
-l [1234 etc]	Select (individual) rack lead (default is ALL rack leads)

The command line arguments allow you to override the default behavior which is to auto-configure the fabric management. SGI recommends you allow the tool to auto-configure fabric management.

Procedure 4-1 Using the smconfig Command to Automatically Configure the InfiniBand Fabric

To automatically configure the ib0 and ib1 InfiniBand fabrics on your system, perform the following:

1. From the system admin controller (admin node), perform the following command:

   ```
   admin:~ # smconfig -f ib0
   Configuring r1lead
   Configuring r2lead
   Configuring r3lead
   Configuring r4lead
   ```

2. Repeat the command for the ib1 fabric, as follows:

```
admin:~ # smconfig -f ib1
Configuring r1lead
Configuring r2lead
Configuring r3lead
Configuring r4lead
```

smadmin InfiniBand Fabric Administration Tool

SGI Tempo provides the smadmin tool that allows you to start up or stop the ib0 and ib1 InfiniBand fabrics. You can also use this tool to restart a fabric or get the status of a fabric. Use this command after your Altix ICE system has been discovered and is powered up (see "smconfig Automatic Fabric Configuration Tool" on page 159).

The smadmin command is, as follows:

```
/opt/sgi/sbin/smadmin
```

It accepts the following options:

Option	Description
-f	Fabric ib0 or fabric ib1 (Required)
-u	Start fabric management
-d	Stop fabric management
-r	Restart fabric management
-s	Get opensmd status (see "InfiniBand Fabric Management Configuration and Operation Overview" on page 163)
-m	Find opensmd MASTER node
-c	Attempt a fabric cleanup
-e [dor updn]	Select routing engine. For more information on routing engine variables, see "Network Topology" on page 168.

Procedure 4-2 Using the smadmin Command to Administer the InfiniBand Fabric

The opensm instance for each fabric is run on different rack leader nodes. For example, the first rack leader controller discovered runs opensm for ib0, the second rack leader controller discovered runs opensm for ib1. The smadmin command has the logic to place opensm on the appropriate rack leader controller.

1. From the system admin controller (admin node), to start fabric management on the `ib0` fabric, perform the following:

```
admin:~ # smadmin -f ib0 -u
Running start of ib0
opensm is stopped
Starting opensm on r1lead
opensm start [ OK ]
smagent-rack: opensm configuration r1lead: opensmd started on fabric ib0
Started opensm for fabric ib0 on r1lead
```

2. From the admin node, to start fabric management on the `ib1` fabric, perform the following:

```
admin:~ # smadmin -f ib1 -u
Running start of ib1
Another fabric has opensm (pid 1253) running...
smadmin notice : Another opensm is already running on r1lead
Proceeding to next rack lead opensm is stopped Starting opensm on r2lead opensm start [ OK ]
smagent-rack: opensm configuration r2lead: opensmd started on fabric ib1
Started opensm for fabric ib1 on r2lead
```

Note: The output for the command looks somewhat different because fabric `ib0` is already running and the fabric management software detects this.

If a fabric fails to start, you will see output similar to the following:

```
Running start on r1lead
smadmin: smadmin error : Invalid configuration on r1lead - Re run /opt/sgi/sbin/smconfig for r1le
```

To fix this run the `smconfig` command on rack 1 lead, as follows:

```
admin:~ # smconfig -f ib0 -l 1
Configuring r1lead
```

You should now be able to start fabric `ib0` (admin:~ # **`smadmin -f ib0 -u`**)

3. If the both fabric managers started ok, you should be able to ping various `-ib0` and `-ib1` host names in your system (use the `ifconfig`(8) command to get the

IP address). From one of the rack leader controllers, `ping` the `service0 ib0` interface, as follows:

```
r1lead# ping -c 1 10.148.0.67
PING 10.148.0.67 (10.148.0.67) 56(84) bytes of data.
64 bytes from 10.148.0.67: icmp_seq=1 ttl=64 time=0.013 ms

--- 10.148.0.67 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.013/0.013/0.013/0.000 ms
```

If you are not able to ping a system node at this point, it is most likely a cabling issue.

4. To stop the fabric management software on a fabric, perform the following:

```
admin:~ # smadmin -f ib0 -d
Running stop of ib0
opensm is running with pid of 1253...
......
opensm shutdown [ OK ]
smagent-rack: opensm configuration r1lead: opensmd stopped on fabric ib0
```

5. The fabric manager for each fabric runs on a different rack leader controller node. There is one MASTER node and no standby. From the admin node, to find the MASTER node, perform the following:

```
admin:~ # smadmin -f ib0 -m
smagent-rack: opensm configuration (from r1lead): opensmd master for ib0 is r1lead
```

6. To determine the status of the fabric management software running on your system, perform the following:

```
admin:~ # smadmin -f ib0 -s
Running status of ib0 on r1lead
opensm is running with pid of 30761...
Running status of ib0 on r2lead
Another fabric has opensm (pid 30263) running...
Running status of ib0 on r3lead
opensm is stopped
Running status of ib0 on r4lead
opensm is stopped.
```

Procedure 4-3 Troubleshooting the InfiniBand Fabric

If the fabric management software dies or exits incorrectly, a state may exist that will prevent it form being re-started on that fabric until a cleanup of the fabric management database is performed, as follows:

1. Perform this set of commands from the system admin controller (admin node):

```
admin:~ # /opt/sgi/sbin/smadmin -f ib0 -d
admin:~ # /opt/sgi/sbin/smadmin -f ib0 -c
admin:~ # /opt/sgi/sbin/smadmin -f ib0 -u
```

2. Repeat for `ib1` fabric if necessary.

Fabric Management and Rebooting

Although the fabric management software can detect changes in the fabric, like the rebooting of a single blade, it not designed to cope with major changes in the fabric, such as, the loss of a switch, rebooting of a whole rack, or rebooting of all of the compute blades. If a reboot of a single rack or all racks or all blades occurs, it is necessary to retart the fabric management software for each fabric. Use the `smadmin` command, as described in "`smadmin` InfiniBand Fabric Administration Tool" on page 160.

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 by setting is in the `SWEEP` variable in the `opensm-ib0.conf` and `opensm-ib1.conf` configuration files located in the `/etc/ofa` directory on the rack leader node.

Note: SGI highly recommends that you do **NOT** change this variable.

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.

A sample `opensm-ibx.conf` configuration file is, as follows:

Example 4-1 `opensm-ib0.conf` and `opensm-ib.conf` Configuration Files

```
# DEBUG mode
#  This option specifies a debug option.
#  These options are not normally needed.
#  The number following -d selects the debug
#  option to enable as follows:
#  OPT   Description
#  ---    -----------------
#  0  - Ignore other SM nodes.
#  1  - Force single threaded dispatching.
#  2  - Force log flushing after each log message.
#  3  - Disable multicast support.
#  4  - Put OpenSM in memory tracking mode.
#  10.. Put OpenSM in testability mode.
#  none, no debug options are enabled.
DEBUG=none

# LMC
#  This option specifies the subnet's LMC value.
#  The number of LIDs assigned to each port is 2^LMC.
#  The LMC value must be in the range 0-7.
#  LMC values > 0 allow multiple paths between ports.
#  LMC values > 0 should only be used if the subnet
#  topology actually provides multiple paths between
#  ports, i.e. multiple interconnects between switches.
#  OpenSM defaults to LMC = 0, which allows
#  one path between any two ports.
LMC=0

# MAXSMPS
#  This option specifies the number of VL15 SMP MADs
#  allowed on the wire at any one time.
#  Specifying -maxsmps 0 allows unlimited outstanding SMPs.
#  Without -maxsmps, OpenSM defaults to a maximum of
#  one outstanding SMP.
MAXSMPS=0

# REASSIGN_LIDS
#  This option causes OpenSM to reassign LIDs to all
#  end nodes. Specifying "REASSIGN_LIDS=yes" on a running subnet
```

```
#  may disrupt subnet traffic.
#  With "REASSIGN_LIDS=no", OpenSM attempts to preserve existing
#  LID assignments resolving multiple use of same LID.
REASSIGN_LIDS="yes"

# SWEEP
#  This option specifies the number of seconds between
#  subnet sweeps.  Specifying SWEEP=0 disables sweeping.
#  OpenSM defaults to a sweep interval of 10 seconds.
SWEEP=10

# TIMEOUT
#  This option specifies the time in milliseconds
#  used for transaction timeouts.
#  Specifying -t 0 disables timeouts.
#  Without -t, OpenSM defaults to a timeout value of
#  200 milliseconds.
TIMEOUT=200

# OSM_LOG
#  This option defines the log to be the given file.
#  By default the log goes to /tmp/osm.log.
#  For the log to go to standard output use OSM_LOG=stdout.
OSM_LOG=/var/log/osm-ib0.log

# VERBOSE
#  This option increases the log verbosity level.
#  The "-v" option may be specified multiple times
#  to further increase the verbosity level.
#   "-V" option sets the maximum verbosity level and
#   forces log flushing.
#   The "-V" is equivalent to "-vf 0xFF -d 2".
VERBOSE="none"

# ROUTING_ENGINE
#  This option chooses the routing engine instead of
#  the Min Hop algorithm which is default.
#  Valid routing engines are :-
#         Min Hop, dor, updn, file, ftree, lash
#  To switch to different routing engine set the engine
#  name in ROUTING_ENGINE (i.e.  ROUTING_ENGINE=lash).
```

```
#  For Min Hop use ROUTING_ENGINE="none" or ROUTING_ENGINE=
ROUTING_ENGINE="dor"

# GUID_FILE
#  This option only allowed when UPDN algorithm is activated
#  It specifies the guid list file from which to fetch the guid list
#  The file contain in each line only one valid guid
GUID_FILE="none"

#  This option specifies the local port GUID value
#  with which OpenSM should bind.  OpenSM may be
#  bound to 1 port at a time.
#  If GUID given is 0, opensmd use PORT_NUM parameter.
#  Without -g (GUID="none"), OpenSM trys to use the default port.
#  example GUID="0x0005ad00000517c9"
GUID="none"

# OSM_HOSTS
#  The list of all SM's IP addresses in InfiniBand subnet
#  Used to handover mechanism
#  example OSM_HOSTS="128.162.246.221 128.162.246.42"
OSM_HOSTS="none"

# OSM_CACHE_DIR
OSM_CACHE_DIR="/var/cache/osm/ib0"

# CACHE_OPTIONS
#  Cache the given command line options into the file
#  /var/cache/osm/opensm-ib0.opts for use next invocation
#  The cache directory can be changed by the environment
#  variable OSM_CACHE_DIR
#  Set to '--cache-options' or '-c' in order to enable
CACHE_OPTIONS="-c"

# HONORE_GUID2LID
#  This option forces OpenSM to honor the guid2lid file,
#  when it comes out of Standby state, if such file exists
#  under OSM_CACHE_DIR, and is valid.
#  Set to '--honor_guid2lid' or '-x' to enable.
#  By default this is FALSE. Will be set automatically to '--honor_guid2lid'
#  if OSM_HOSTS includes list of more then one IP addresses.
```

```
HONORE_GUID2LID="-x"

# RCP
#  This option osed by SLDD daemon for handover mechanism
#  to copy local cache file to remote computer
RCP=/usr/bin/scp

# RSH
#  This option osed by SLDD daemon for handover mechanism
#  to execute commands on remote computer
RSH=/usr/bin/ssh

# RESCAN_TIME
#  This option osed by SLDD daemon for handover mechanism
#  Time between sweep of sldd daemon in seconds
RESCAN_TIME=60

# PORT_NUM
#  This option defines HCA's port number which OpenSM should bind
PORT_NUM=1

# ONBOOT
#  To start OpenSM automatically set ONBOOT=yes
ONBOOT=yes

# MULTI_FABRIC
# Allow multiple fabrics (and copies of OpenSM) on the same SM host
MULTI_FABRIC=yes
```

Each fabric is addressed by a global unqiue identifier (GUID) and unique HCA port (see Figure 4-1 on page 168). Each fabric has a unique `GUID` set in its respective configuration file.

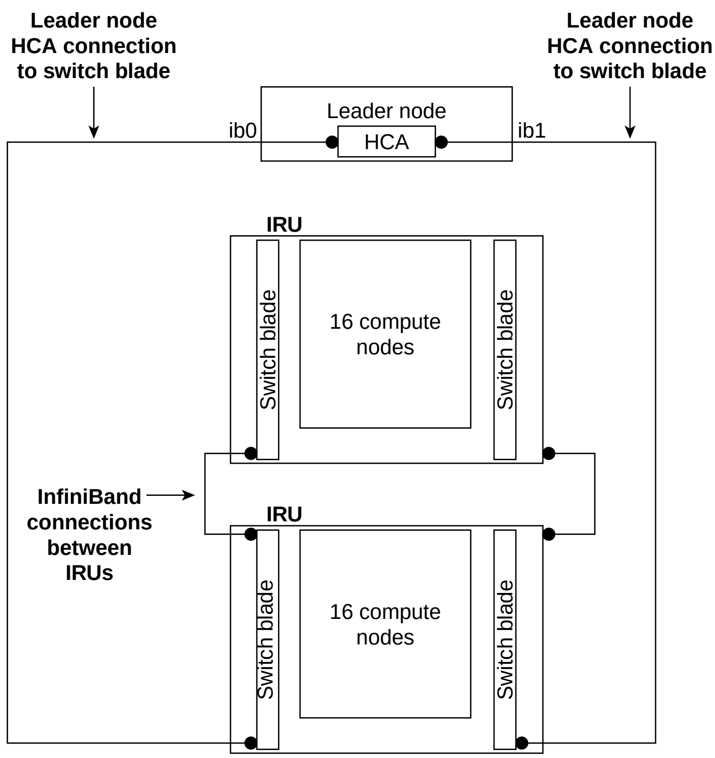

Figure 4-1 Two InfiniBand Fabrics in a System with Two IRUs

Network Topology

For SGI Altix ICE systems with a hypercube topology, SGI requires `ROUTING_ENGINE="dor"` as the default variable (dimension order routing 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 requires `ROUTING_ENGINE="updn"` as the default variable. 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 file to show the GUID of the SM master.

Configuring the InfiniBand Fat-tree Network Topology

This section describes how to configure InfiniBand fat-tree network topology.

Procedure 4-4 Configuring InfiniBand Fat-tree Network Topology

To configure the InfiniBand fat-tree network topology on an SGI Altix ICE 8200 series system, perform the following steps:

1. Create the following files on the system admin controller (admin node):

   ```
   /opt/sgi/var/smadmin/ext_switch-ib0
   /opt/sgi/var/smadmin/ext_switch-ib1
   ```

2. With your favorite text editor, specify the external switches on the ib0 and ib1 fabrics and their model names in colon-separated lists. For example, for the /opt/sgi/var/smadmin/ext_switch-ib0 file add two entries, one for each external switch and its model name, as follows:

   ```
   service52020:ISR2012

   service51020:ISR2012
   ```

   Perform this step for the /opt/sgi/var/smadmin/ext_switch-ib1 file.

3. From the admin node, run the following command:

   ```
   # smconfig -f ib0 -r updn -z -a /opt/sgi/var/smadmin/ext_switch-ib0
   ```

4. From the admin node, run the following command:

   ```
   # smadmin -f ib0 -u
   ```

Repeat these two steps for the ib1 fabric

5. From the admin node, run the following command:

```
# smconfig -f ib1 -r updn -z -a /opt/sgi/var/smadmin/ext_switch-ib1
```

6. From the admin node, run the following command:

```
# smadmin -f ib1 -u
```

Configuring and Initializing the InfiniBand Fabric Manually

This section describes the changes you need to make to the /etc/opensm-ib0.conf or /etc/opensm-ib1.conf configuration file to configure opensm software, how to start the opensmd-ib0 and opensmd-ib1 daemons, and verify the fabric is operating. For an overview of fabric configuration and management, see "InfiniBand Fabric Management Configuration and Operation Overview" on page 163.

Procedure 4-5 Configuring and Initializing the InfiniBand Fabric Manually

To configure, initialize, and verify the InfiniBand farbic, perform the following steps:

1. From the admin node, connect to the leader node or rack 1, as follows:

```
# ssh r1lead
```

Note: Before you attempting to initialize the InfiniBand fabric, make sure all compute nodes are booted and operational.

2. From the admin node, determine and record the IP addresses of the leader nodes, as follows:

```
# ping -c 1 r1lead
PING r1lead.ice.americas.sgi.com (172.16.0.2) 56(84) bytes of data.
64 bytes from r1lead.ice.americas.sgi.com (172.16.0.2): icmp_seq=1 ttl=64 time=0.127 ms

--- r1lead.ice.americas.sgi.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.127/0.127/0.127/0.000 ms
# ping -c 1 r2lead
PING r2lead.ice.americas.sgi.com (172.16.0.3) 56(84) bytes of data.
64 bytes from r2lead.ice.americas.sgi.com (172.16.0.3): icmp_seq=1 ttl=64 time=0.089 ms

--- r2lead.ice.americas.sgi.com ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.089/0.089/0.089/0.000 ms
# ping -c 1 r3lead
PING r3lead.ice.americas.sgi.com (172.16.0.4) 56(84) bytes of data.
64 bytes from r3lead.ice.americas.sgi.com (172.16.0.4): icmp_seq=1 ttl=64 time=0.129 ms

--- r3lead.ice.americas.sgi.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.129/0.129/0.129/0.000 ms
# ping -c 1 r4lead
PING r4lead.ice.americas.sgi.com (172.16.0.5) 56(84) bytes of data.
64 bytes from r4lead.ice.americas.sgi.com (172.16.0.5): icmp_seq=1 ttl=64 time=0.136 ms

--- r4lead.ice.americas.sgi.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.136/0.136/0.136/0.000 ms
```

3. From the leader node, issue an `ibstat` command to determine the `Port GUID` values, as follows:

```
r1lead:/ # ibstat
CA 'mthca0'
        CA type: MT23108
        Number of ports: 2
        Firmware version: 3.3.3
        Hardware version: a1
        Node GUID: 0x0008f1040397b03c
        System image GUID: 0x0008f1040397b03f
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 10
                Base lid: 1
                LMC: 0
                SM lid: 1
                Capability mask: 0x02510a6a
                Port GUID: 0x0008f1040397b03d <—<< goes into opensm-ib0.conf
        Port 2:
                State: Initializing
                Physical state: LinkUp
                Rate: 10
                Base lid: 0
```

```
LMC: 0
SM lid: 0
Capability mask: 0x02510a68
Port GUID: 0x0008f1040397b03e <—<< goes into opensm-ib1.conf
```

Note: Get usage information on the ibstat command, as follows:

```
r1lead:/ # ibstat --help
Usage: ibstat [-d(ebug) -l(ist_of_cas) -s(hort) -p(ort_list) -V(ersion)]  [portnum]
        Examples:
                ibstat -l         # list all IB devices
                ibstat mthca0 2 # stat port 2 of 'mthca0'
```

4. From the leader node, change directory to the /etc, as follows:

   ```
   r1lead:/ # cd /etc
   ```

5. Using your favorite editor, open the opensm-ib0.conf file and enter the Port GUID: value, in this example, 0x0008f1040397b03d, as follows:

   ```
   GUID="0x0008f1040397b03d"
   ```

6. Using your favorite editor, open the opensm-ib1.conf file and enter the Port GUID: value, in this example, 0x0008f1040397b03e, as follows:

   ```
   GUID="0x0008f1040397b03e"
   ```

7. There are two routing options available based on fabric topology, as follows:

   - For fat-tree network topology, use updn.
   - For hypercube network topology, use dor.

   For more information, see "Network Topology" on page 168.

   For hypercube network topology, set the ROUTING_ENGINE variable in both configuration files to dor (dimension order routing) , as follows:

   ```
   ROUTING_ENGINE="dor"
   ```

   For fat—tree network topology, set the ROUTING_ENGINE variable in both configuration files to updn (Unicast routing algorithm) , as follows:

   ```
   ROUTING_ENGINE="updn"
   ```

8. To initialize the ib0 fabric, start the opensmd-ib0 daemon, as follows:

   ```
   # /etc/init.d/opensmd-ib0 start
   ```

9. To initialize the ib1 fabric, start the opensmd-ib1 daemon, as follows:

   ```
   # /etc/init.d/opensmd-ib1 start
   ```

10. Use the the ibnetdiscover command to verify the fabric, as follows:

```
r1lead:/ # ibnetdiscover -l
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"
Ca       : 0x0030487aa7940000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa78c0000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0008f10403988198 ports 2 devid 0x6278 vendid 0x8f1 "service0-ib0 HCA-1"
Ca       : 0x0030487aa7840000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa79c0000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa7900000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa7980000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0008f104039881a8 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"
```

Note: Get usage information on the ibnetdiscover command, as follows:

```
r1lead:/ # ibnetdiscover --help
Usage: ibnetdiscover [-d(ebug)] -e(rr_show) -v(erbose) -s(how) -l(ist) -g(rouping) -H(ca_list) -S(witch_
 -V(ersion) -C ca_name -P ca_port -t(imeout) timeout_ms --switch-map switch-map] []
--switch-map  specify a switch-map file
```

11. Exit the rack leader controller (leader node) and return to the system admin controller (admin node), you should be good to go now.

Useful Utilities and Diagnostics

The openib-diags 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/bin directory, as follows:

```
r1lead:~ # cd /usr/bin
r1lead:/usr/bin # ls ib*
ibaddr          ibcheckstate     ibdiscover.pl       ibnetdiscover     ib_rdma_bw   ibstatus        ...
```

```
ibcheckerrors     ibcheckwidth     ibdmchk              ibnlparse         ib_rdma_lat  ibswitches      ...
ibcheckerrs       ibclearcounters  ibdmsh               ibnodes           ib_read_bw   ibsysstat       ...
ibchecknet        ibclearerrors    ibdmtr               ibping            ib_read_lat  ibtopodiff      ...
ibchecknode       ib_clock_test    ibfindnodesusing.pl  ibportstate       ibroute      ibtracert       ...
ibcheckport       ibdiagnet        ibhosts              ibprintca.pl      ib_send_bw   ibv_asyncwatch  ...
ibcheckportstate  ibdiagpath       ibis                 ibprintswitch.pl  ib_send_lat  ibv_devices     ...
ibcheckportwidth  ibdiagui         iblinkinfo.pl        ibqueryerrors.pl  ibstat       ibv_devinfo
```

This section covers the following topics:

- "ibstat and ibstatus Commands" on page 174
- "perfquery Command" on page 176
- "ibnetdiscover Command" on page 177
- "ibdiagnet Command" on page 178

ibstat and ibstatus Commands

You can use the ibstat command to see the current status of the host channel adapaters (HCA) in your InfiniBand fabric incluing 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, 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" on page 183
- "Node Replacement Procedure for Admin, Leader, and Service Nodes" on page 186
- "Inventory Verification Tool" on page 192
- "System Monitoring Overview" on page 195
- "System Monitoring Operation" on page 198
- "Monitoring System Metrics with Performance Co-Pilot" on page 201
- "Setting up the Embedded Support Partner" on page 203
- "Troubleshooting" on page 205
- "`kdump` Utility" on page 209
- "System Firmware" on page 210

Maintenance Procedures

This section describes some common maintenance procedures, as follows:

- "Temporarily Take a Node Offline for Maintenance" on page 183
- "Permanently Replace a Failed Blade" on page 184
- "Permanently Remove a Blade " on page 185
- "Add a New Blade" on page 186

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. In the Tempo 1.3 release, 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 68, for more information.

6. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 117 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. In the Tempo 1.3 release, 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 68, 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. In the Tempo 1.3 release, 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 68, for more information.
3. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 117 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 Admin, Leader, and Service Nodes

This section describe how to install and configure a spare admin, leader, or managed service node. It covers the following topics:

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 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.

Shelf Spare Admin or Leader Node Availability

A shelf spare node is like an existing admin or leader node, but it sits on a shelf to be used in an emergency.

If the admin or leader node should fail, the shelf spare can be swapped in to position to take over the duities of the failed node.

If you wish to make use of shelf 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.

Migrating to a Shelf Spare: Installing the Hardware

If you find that an admin node or leader node has failed and you need to replace it with a shelf 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 Migrating to a Shelf 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 the failed node from AC power.
3. Remove the two system disks from the failed node and set them aside for later reinstallation.
4. Remove the Ethernet cables. Label the cables to avoid confusing them. It is important that they stay in the same jacks in the new node.
5. Remove the system from the rack.
6. Install the shelf spare system into the rack.
7. Install the system disks you set aside in step 3 (from the system you are replacing).

8. Connect the Ethernet cables in the same way they were connected to the replaced node.
9. Connect AC power.
10. Connect a keyboard and VGA monitor (and mouse if you like).
11. Do **NOT** power up the system just yet. Proceed to "Migrating to a Shelf Spare: Importing the Disk Volumes" on page 189.

Migrating to a Shelf Spare: Importing the Disk Volumes

This section describes how to import the disk volumes into the new node installed in "Migrating to a Shelf Spare: Installing the Hardware" on page 188.

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 Shelf Spare: Booting for the First Time on the Migrated Node" on page 190.

Migrating to a Shelf Spare: Booting for the First Time on the Migrated Node

This section provides details on booting the system for the first time on the replacement node. These instructions include some special things you need to do with GRUB boot loader to ensure, for this boot only, that the console output goes to the VGA screen. This important because on leader nodes, there is no way to connect to the BMC with IPMI at this moment to use the IPMI serial console. The `console` command will not work for the leader node until the system is configured as described in this section. The network will not be properly configured until the end of this procedure either.

Procedure 5-7 Migrating to a Shelf Spare: Booting for the First Time on the Migrated Node

To boot for the first time on a migrated node, perform the following steps:

1. 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 Shelf Spare: Importing the Disk Volumes" on page 189).

On leader nodes, the node will attempt to PXE boot as it comes up. This is normal. The PXE boot will fail and this is normal. On the admin node, no PXE will be attempted. In either case, the system will eventually try to boot from disk.

Note: If it is not booting from disk, the wrong volume may be selected as the boot disk in the LSI BIOS tool. See "Migrating to a Shelf Spare: Importing the Disk Volumes" on page 189.

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 1.4, the highlighted choice should be : **SUSE Linux Enterprise Server 10 SP2**. Enter **e** to edit the boot parameters for this boot only.

2. Arrow down once so that the line starting **kernel** is highlighted.
3. Enter **e** to edit the kernel parameters.
4. Now you need to add `console=tty0` as the final parameter in the list. This ensures that console output goes to the VGA screen for this boot. Enter the space character followed by `console=tty0`. The line should look similar to the following after adding the console parameter (characters wrapped in the front):

```
<hkernel=128M@16M rootflags=prjquota,logbsize=256k console=tty0
```

5. Press the `Enter` key.
6. 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.
7. Log in as root.
8. 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 Ethernet cards installed, in addition to the ones that come with the system itself, the script could possibly guess the integrated Ethernet devices incorrectly. This may mean you have to manually configure networking including the `ifcfg-eth-id-*` files in `/etc/sysconfig/network` and the `/etc/udev/rules.d/30-net_persistent_names.rules` file (to number them how you want and ensure integrated Ethernet is `eth0` and `eth1`).

At this time, networking should be operational.

9. Reboot the node and let it boot normally.

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 31.

To make an inventory snapshot of an Altix ICE system, use the following command from the system admin controller (admin node).

```
system-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:

```
system-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 use the `ivt` command to show general information about your system (note that only a portion of the output of this command is shown below), as follows:

```
system-admin:~ # ivt -S

Your system has 6 compute blades.

All 6 blades have the following characteristics:
    bios_date: 05/29/2007
    cpu_core_count: 8
    cpu_model: Intel(R) Xeon(R) CPU E5345 @ 2.33GHz
    kernel: 2.6.16.46-0.12-smp
    memsize: 2059264
    os_product: SLES
    os_vendor: SUSE
    os_version: 10.1

The following characteristics have different values for some blades.

  ib0_phys_state (State of InfiniBand ib0 physical link):
        4 blades have ib0_phys_state == LinkUp (r1i0n0, r1i1n0, r1i0n8, ...)
        2 blades have ib0_phys_state == unknown (r1i0n1, r1i1n1)
      Query the  value for all blades with:
        ivt -Q -w blades -f 'blade $blade has ib0_phys_state $ib0_phys_state'

  ib0_rate (Rate of InfiniBand ib0 link - Gb/sec):
        2 blades have ib0_rate == unknown (r1i0n1, r1i1n1)
        4 blades have ib0_rate == 20 (r1i0n0, r1i1n0, r1i0n8, ...)
      Query the  value for all blades with:
        ivt -Q -w blades -f 'blade $blade has ib0_rate $ib0_rate'
...

  ib_bios_rev (Revision of InfiniBand BIOS on blade):
        2 blades have ib_bios_rev == unknown (r1i0n1, r1i1n1)
        4 blades have ib_bios_rev == 1.2.0 (r1i0n0, r1i1n0, r1i0n8, ...)
      Query the  value for all blades with:
        ivt -Q -w blades -f 'blade $blade has ib_bios_rev $ib_bios_rev'
```

```
  image (image provisioned on blade):
          5 blades have image == compute-sles10sp1 (r1i0n1, r1i1n1, r1i1n0, ...)
          1 blades have image == erikj-blade-mksiimage (r1i0n0)
      Query the  value for all blades with:
        ivt -Q -w blades -f 'blade $blade has image $image'

  rack_blade_count (number of booted blades in this blades rack):
          2 blades have rack_blade_count == 5 (r1i0n1, r1i1n1)
          4 blades have rack_blade_count == 4 (r1i0n0, r1i1n0, r1i0n8, ...)
      Query the  value for all blades with:
        ivt -Q -w blades -f 'blade $blade has rack_blade_count $rack_blade_count'

InfiniBand GUID check:
  Do fabric (ibnetdiscover) and blades (ib stat) have same GUIDs?
    ib0 plane: unmatched GUIDs
    GUIDs seen on blade ports, missing on fabric: unknown 0030487aa7940000
    GUIDs see on fabric, missing on blade ports: 0030487aa7840000 0030487aa7980000
    ib1 plane: unmatched GUIDs
    GUIDs seen on blade ports, missing on fabric: unknown 0030487aa7950000
    GUIDs see on fabric, missing on blade ports: 0030487aa7850000 0030487aa7990000

InfiniBand Link state check:
  Are any IB ports not ACTIVE, not 20 Gb/sec rate or not Up?
...
```

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:

```
system-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:

```
system-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:

```
system-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 8200 system. It displays web browser-based, real-time (on demand) histograms of system metrics, as shown in Figure 5-1 on page 196.

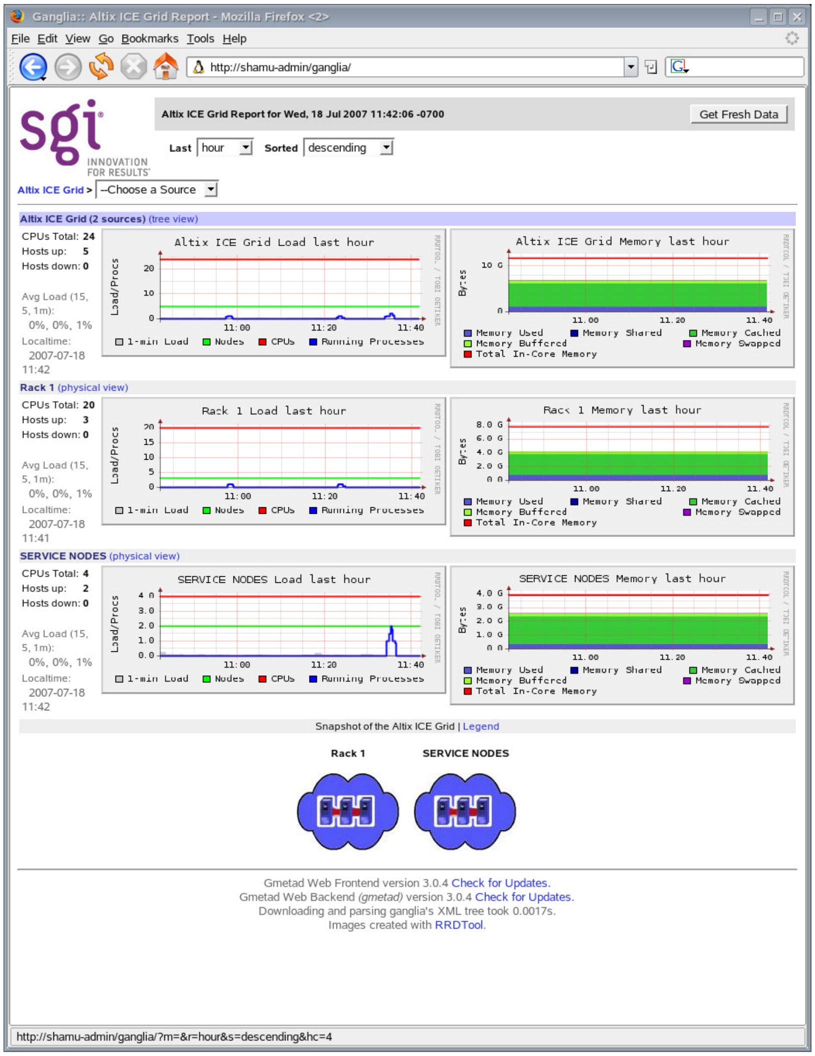

Figure 5-1 Ganglia System Monitor

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 5-2 on page 198 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.

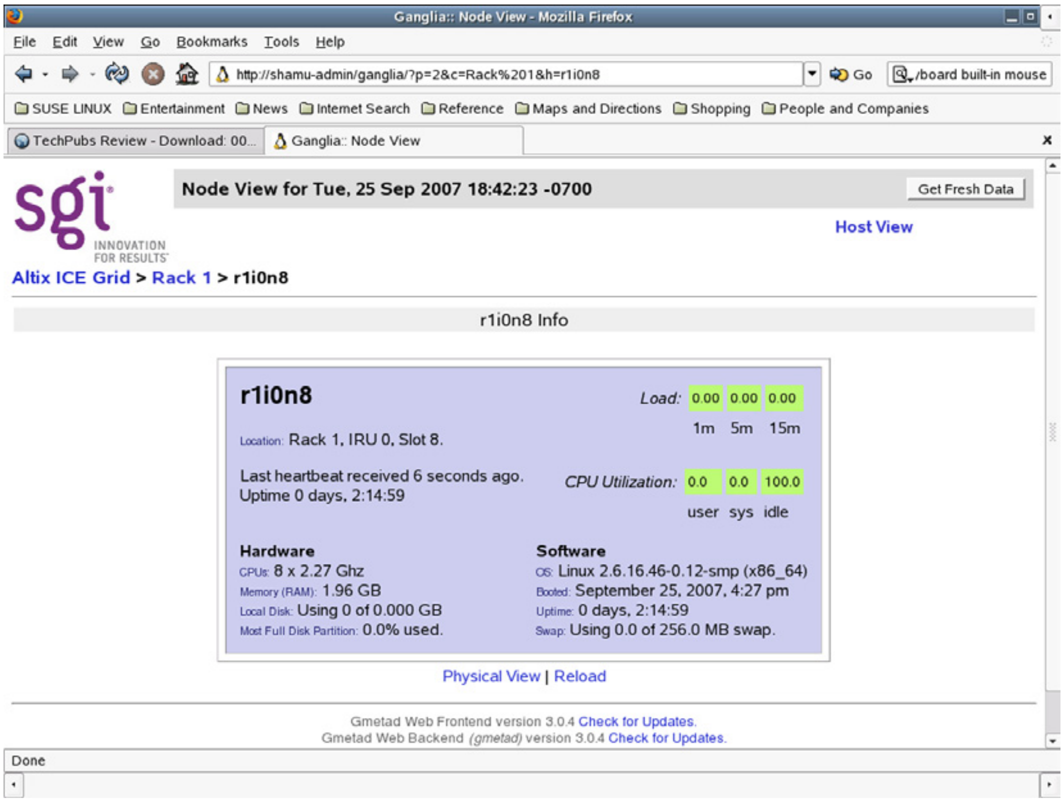

Figure 5-2 Ganglia System Monitoring Node View

System Monitoring Operation

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

- "Accessing the Ganglia System Monitor" on page 199
- "Monitoring System Metrics" on page 199
- "SEL / Hardware Event Monitoring" on page 199
- "Node Availability Monitoring" on page 200

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 the SGI Altix ICE system is monitored via Ganglia. A node is declared as down if it does not send a hearbeat for approximately 80 seconds. In this event, a `NODE_DOWN` Embedded Support Partner (ESP) event is generated. You can observe this event via `syslog` or using ESP. Furthermore, administrator-defined 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 `NODE_DOWN` event contains the following useful information:

`MSG ::=` <*syslog-prefix*> `TEMPO:`<*node*> `EVENT:`<*event*> `APP:`<*app*> `Date:`<*date*> `VERSION:`<*version*> `TEXT` <*text*>

The `NODE_DOWN` event is created only once for a failed node.

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*event*>	`NODE_DOWN`

<*app*>	MIA
<*date*>	date / time of the event
<*version*>	1.0
<*text*>	Ganglia Web link to failed node

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. As of the SGI ProPack 5 Service Pack 5 release, 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:

```
sys-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
```

Monitoring SDR Metrics

In SGI ProPack 5 SP5, 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:

```
sys-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>:<metricNam>

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

```
sys-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*.

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 5-8 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:

   ```
   sys-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:

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

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

```
system-admin:~ # espconfig -enable ipaddr 127.0.0.0
system-admin:~ # espconfig -enable ipaddr 127.0.0.1
system-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" on page 205
- "`tempo-info-gather` Command" on page 207
- "`cminfo` Command" on page 208

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:

```
system-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:

```
sys-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:

```
sys-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:

```
sys-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 ipmitool. For example, if you are logged onto the r1lead rack leader controller, the following command gets the BMC firmware revision:

```
# ipmitool -U ADMIN -P ADMIN -I lanplus -H r1i0n0-bmc 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:

```
system-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
```

Index