# SGI® Altix® ICE System Administrator's Guide

007–4993–001

The SGI Altix ICE software stack depends on several open source packages which require attribution. They are, as follows:

**c3:**

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

**conserver:**

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

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

Copyright 1992 Purdue Research Foundation, West Lafayette, Indiana 47907. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or the Regents of the University of California. Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following restrictions: 1. Neither the authors nor Purdue University are responsible for any consequences of the use of this software. 2. The

origin of this software must not be misrepresented, either by explicit claim or by omission. Credit to the authors and Purdue University must appear in documentation and sources. 3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. 4. This notice may not be removed or altered.

---

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

**pysqlite:**

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

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

---

LIMITED RIGHTS LEGEND

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

---

TRADEMARKS AND ATTRIBUTIONS

SGI, the SGI logo, and Altix are registered trademarks and SGI ProPack is a trademark of SGI in the United States and/or other countries worldwide.

Altair is a registered trademark and PBS Professional is a trademark of Altair Engineering, Inc. Intel, Xeon, and Itanium are trademarks or registered trademarks of Intel Corporation. InfiniBand is a trademark of the InfiniBand Trade Association. Linux is a registered trademark of Linus Torvalds. Novell is a registered trademark and SUSE is a trademark of Novell, Inc., in the United States and other countries.

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

# Record of Revision

| Version | Description |
| --- | --- |
| 001 | August 2007<br>Original publication. |

# Contents

# Figures

# Examples

# Procedures

# About This Guide

This guide is a reference document for people who manage the operation of SGI Altix ICE 8000 series systems running SUSE Linux Enterprise Server 10 Service Pack 1 with SGI ProPack 5 for Linux Service Pack 2. It describes how to use SGI Tempo systems management software (v1.0) to perform general system discovery, installation, configuration, and operations on SGI Altix ICE 8200 systems.

This manual contains the following chapters:



## Related Publications

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

- *SGI Altix ICE 8000 System User's Guide*

  This is the hardware user's guide for the SGI Alitx 8000 series systems. It describes the features of the SGI Altix ICE 8000 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 5 for Linux Service Pack 2 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 Service Pack 2 Start Here*, the SGI ProPack 5 SP2 release notes, which contain the latest information about software and documentation in this release, the list of RPMs distributed with SGI ProPack 5 SP2, and a useful migration guide, which contains helpful hints and advice for customers moving from earlier versions of SGI ProPack to SGI ProPack 5, can be found in the `/docs` directory on the SGI ProPack 5 Open/Free Source CD.

  The SGI ProPack 5 for Linux SP2 release notes get installed to the following location on a system running SGI ProPack 5: `/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 System Overview

An SGI Altix ICE 8200 system is an integrated blade environment in the SGI Altix ICE 8000 series 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 system and covers the following topics:



## Hardware Overview

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



For a detailed description, see *SGI Altix ICE 8200 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 4. 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.

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, MVAPICH MPI. The `ib1` fabric is reserved for storage related traffic. The default configuration for MVAPICH MPI is to use only the `ib0` fabric. For more information on the InfiniBand fabric, see Chapter 4, "System Fabric Management" on page 103. For more information on MVAPICH MPI, see Chapter 6, "MVAPICH MPI" on page 143.

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.

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 4. 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 RLC (leader node). The RLC 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 RLC has asked the CMC to enable master power, the RLC can then command each BMC to power up its associated blade. The RLC can also query each BMC to obtain some environmental and error log information about each blade.

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

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 your 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 avoiding the admin node 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 14 and "Network Interface Naming Conventions" on page 19.

---

The rack leader controller (leader node) and system admin controller (admin node) are described in the section that follows ("System Nodes" on page 6).

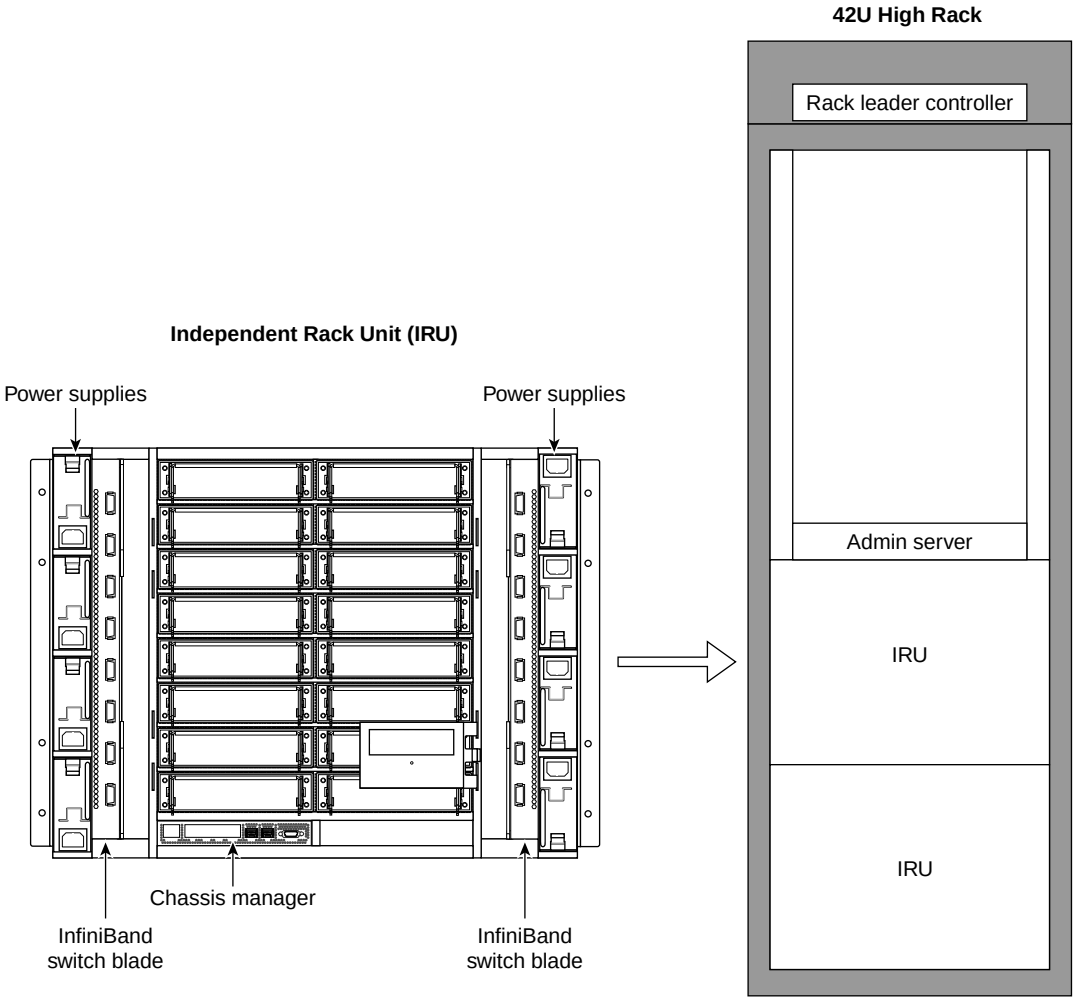


**Figure 1-1** Basic System Building Blocks

Figure 1-2 on page 5 shows chassis manager cabling.

**Note:** All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 5 and Figure 1-3 on page 10 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 17).

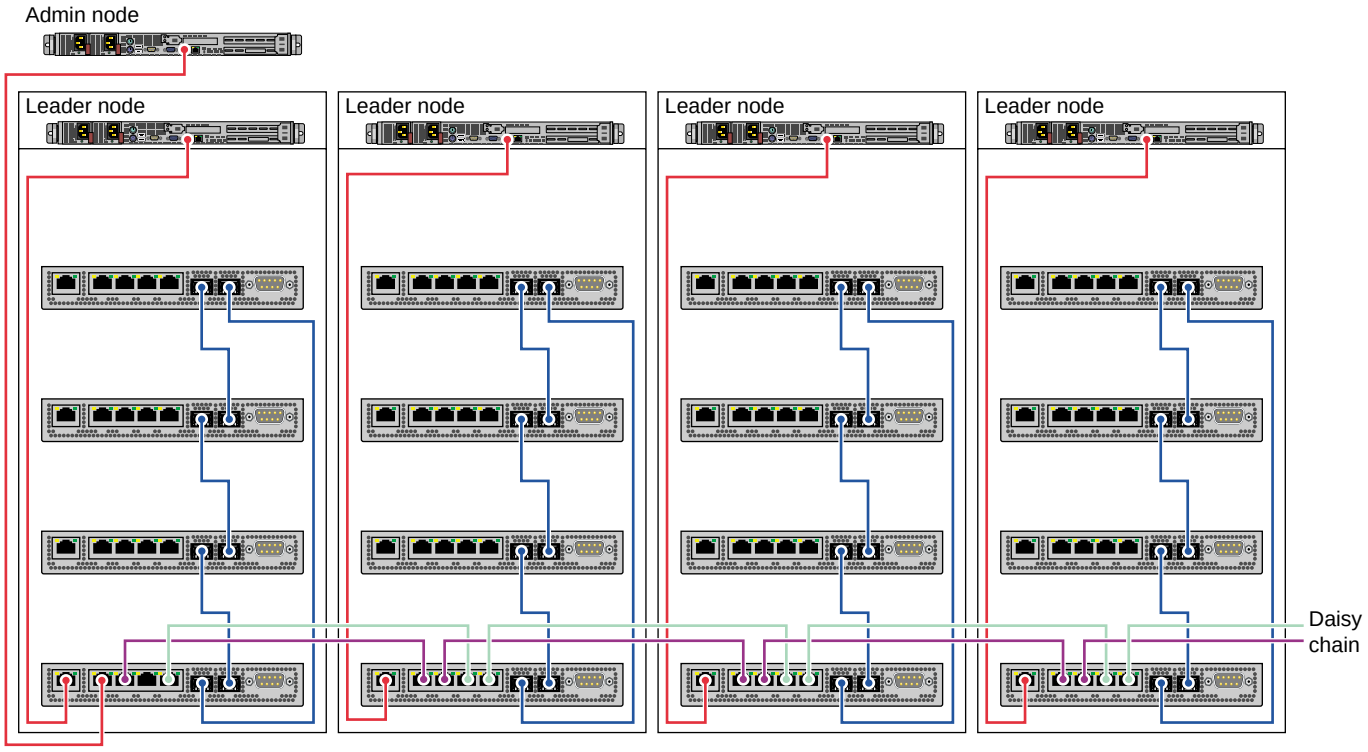


**Figure 1-2** Chassis Manager Cabling

Figure 1-3 on page 10 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 5 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 5. A GigE connection provides the network connection to other leader nodes and to first IRU within its rack as shown in Figure 1-4 on page 12. 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 so on) and a minimal swap space for the compute nodes within its rack

The leader node can contain multiple images for the compute nodes. "Customizing Compute Node Software" on page 80 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 4. 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 RLC (leader node). The RLC 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 RLC has asked the CMC to enable master power, the RLC can then command each BMC to power up its associated blade. The RLC can also query each BMC to obtain some environmental and error log information about each blade.

### Compute Node

Figure 1-1 on page 4 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 or Torque (OpenPBS), as shown in Figure 1-4 on page 12. The compute nodes are controlled and monitored by the leader node to their rack as shown in Figure 1-2 on page 5. 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 via the InfiniBand fabric and to compute nodes in other racks via the InfiniBand fabric connections to other leaders nodes. 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 rack leader controller, which communicates to the appropriate CMC and compute blades. The compute nodes do not communicate directly to the CMC or admin nodes, or RLCs 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 LCD control panel. For example, if you added a NAS cube to your system you need to reconfigure the CMC.

---

**Note:** The LCD control panel is not operational for the first release.

---

### 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 4. 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 12. 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 or Torque (OpenPBS). 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, such as, PBS Professional or Torque.

### Gateway Service Node

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. 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 10 shows a service node and storage service node (NAS cube).

---

**Note:** All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 5 and Figure 1-3 on page 10 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.

---

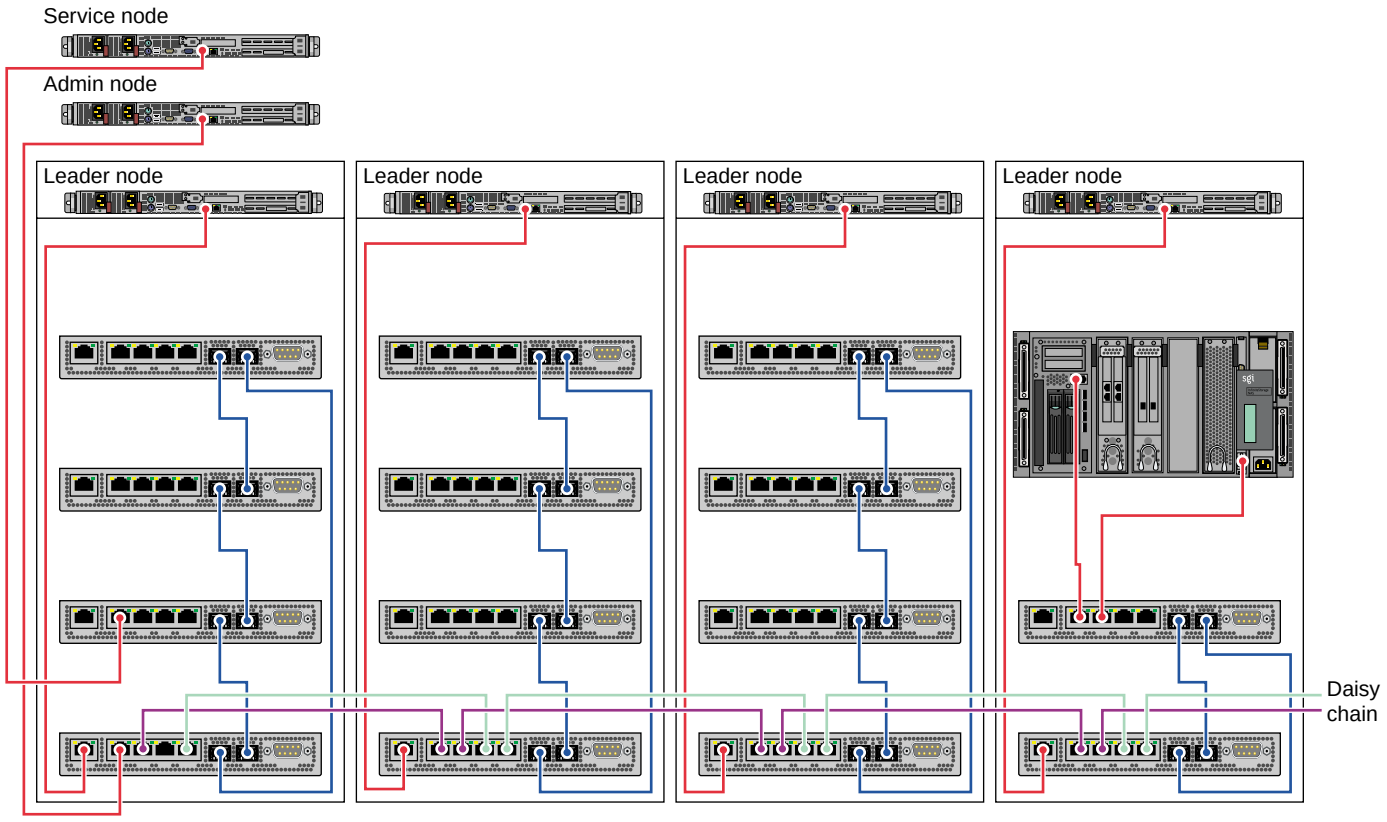


**Figure 1-3** Service Nodes

## 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:

- "InfiniBand Fabric" on page 19

## 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 12. 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 16, Figure 1-7 on page 17, and Figure 1-8 on page 18. 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 RLC is connected to blades in its rack via the GigE VLAN. It is connected to all blades and service nodes via InfiniBand fabric.

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 12).

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.

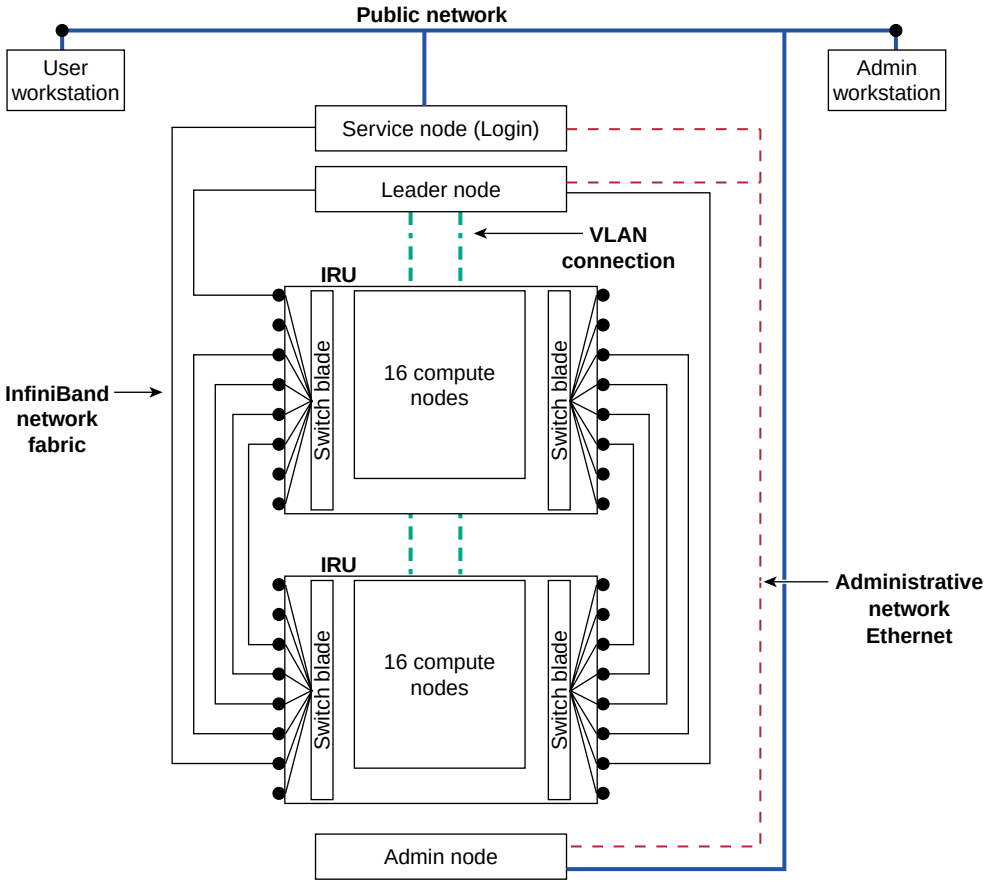


**Figure 1-4** Network Connections In a System With Two IRUs

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 131.

## 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 eth0 to the Linux kernel.

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 13 shows a more detailed view of the Chassis manager.

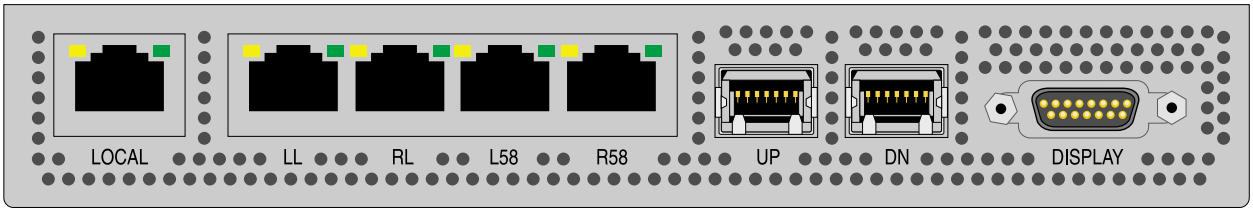


**Figure 1-5** Chassis Manager

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 connect to the GigE switch and the sixteen blade BMCs connect to the 10/100 switch.

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 13. 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 13. 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 5.

- **LL**

  Used to connect service nodes and service storage nodes.

- **RL**

  Used to connect service nodes and service storage nodes.

- **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 10.

## VLANs

The following virtual local area networks (VLANs) are established to optimize the control traffic flow across the entire Ethernet infrastructure:

- `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. This VLAN carries traffic from the system admin controller (admin node) to all of the leaders (including the leader BMCs).

- `VLAN_BMC`

  Includes all 10/100 sub-switches and the `leader_local` ports. This VLAN carries traffic from the leaders to the BMCs on each blade (but not to the leader BMC). See Figure 1-6 on page 16.

- `VLAN_GBE`

  Includes all GigE blade ports and the `leader_local` port. The VLAN carries traffic from the leaders to the primary Ethernet of the blades. See Figure 1-6 on page 16.

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 28. 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.

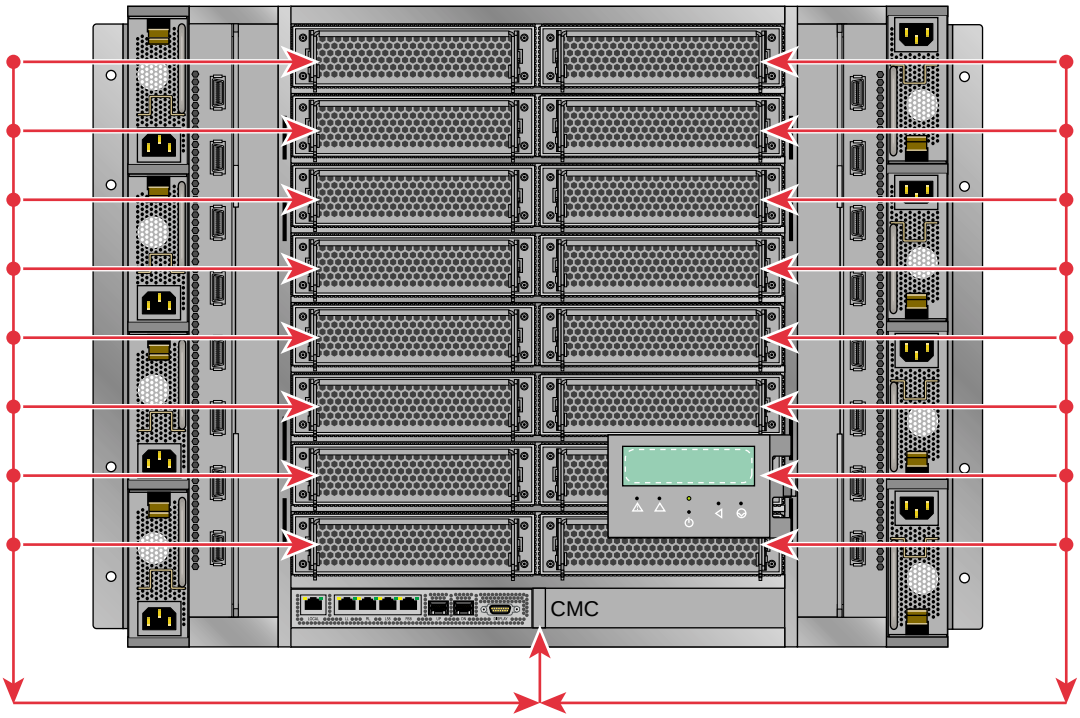


**Figure 1-6** VLAN_GBE and VLAN_BMC Network Connections - IRU View

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).

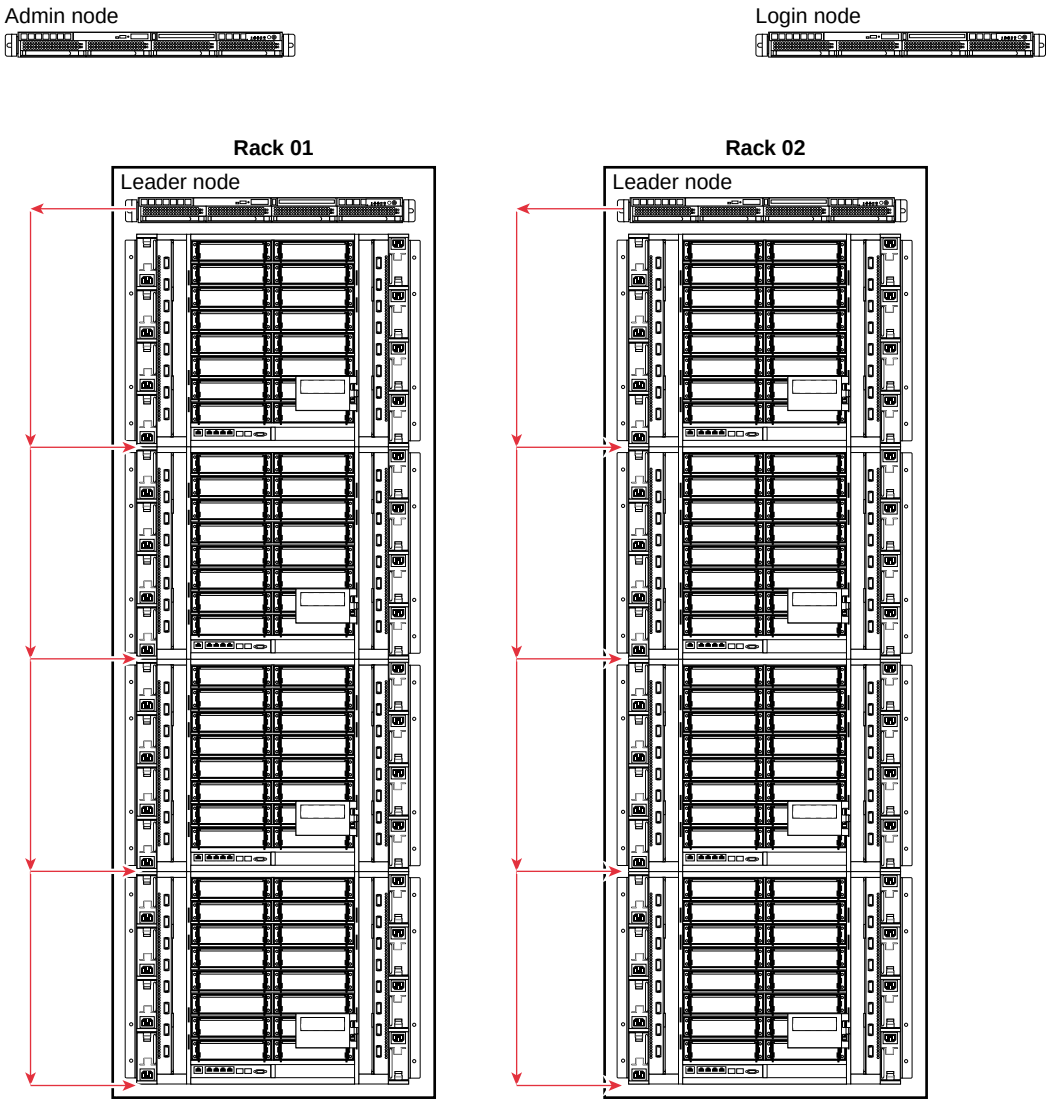


**Figure 1-7** VLAN_GBE and VLAN_BMC Network Connections – Rack View

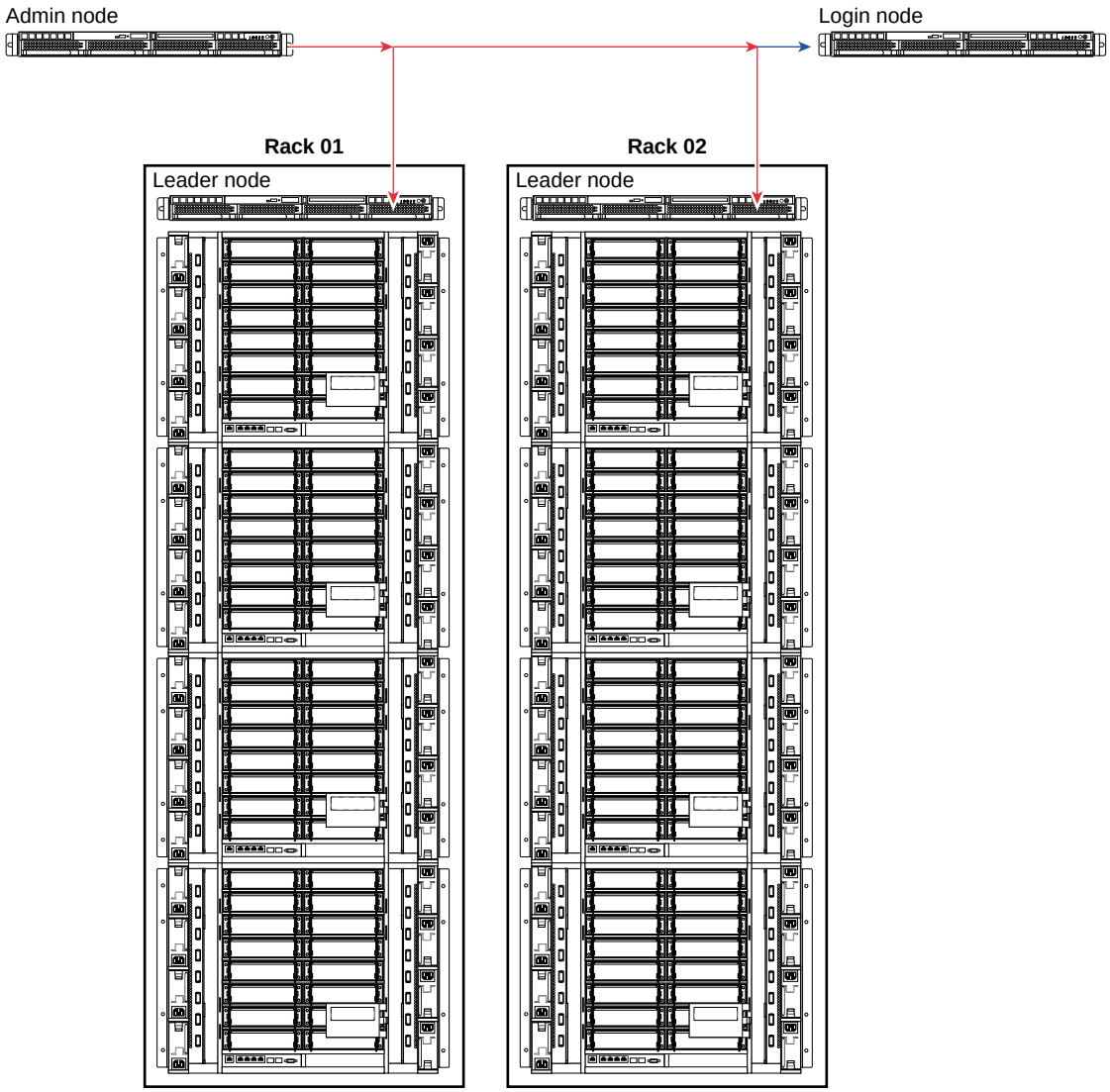


**Figure 1-8** VLAN_HEAD Network Connections

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.

### 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.

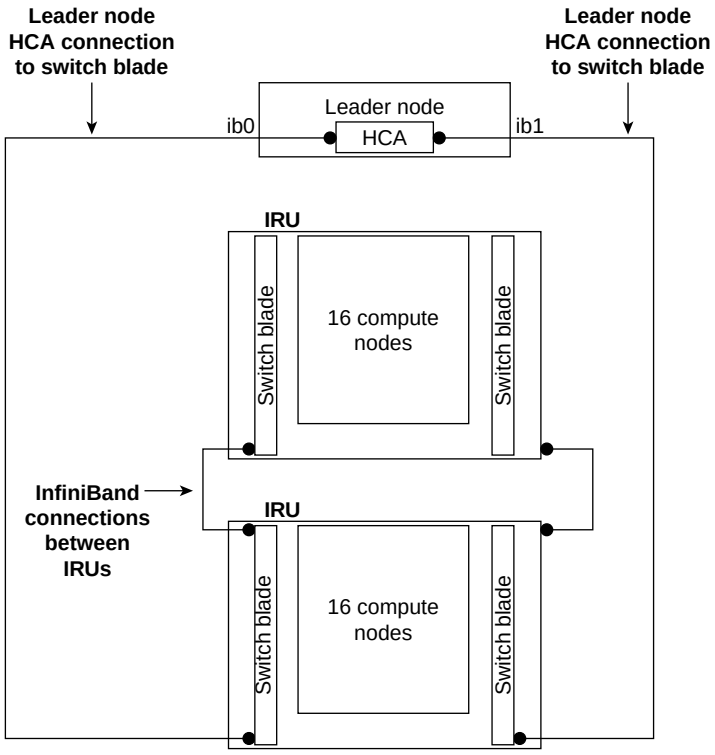


**Figure 1-9** Two InfiniBand Fabrics in a System with Two IRUs

## Network Interface Naming Conventions

As described in "Networks" on page 10, 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 14, 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

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 6) |
| 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-Routable Names" on page 23. |
| Hostname | Returns string of the hostname command. Is technically independent from the other names. |

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

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

## VLAN_Head Network Connections

Table 1-1 on page 21 shows the VLAN_Head network connection names. See Figure 1-8 on page 18.

**Table 1-1** VLAN_HEAD Connections

| Node | Connection Name |
|---|---|
| Admin | admin |
| Service | serviceX<br>serviceX-bmc |
| Leader | rXlead<br>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 the service nodes are normally accessible inside the network, however, you can make them accessible on your own external network instead.

## VLAN_GBE Network Connections

Table 1-2 on page 21 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 16). 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 22 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 16).

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 23 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-Routable 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 23 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-Routable Names

| Node | Node Name |
|---|---|
| Rack | rX |
| IRU | rXiY |
| Blade | rXiYnZ |
| CMC | rXiYc |

## Hostnames

Hostnames are distinct from the non-routable names and are shown in Table 1-6 on page 24. 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:** For the first release, hostnames should **not** be changed. Service hostnames need to match the node name, that is, serviceX.

## 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 25 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 neceassary to alias the rXiYnZ names (currently non-resolvable) to rXiYnZ-ib0 if this is needed by MPI.

**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



## `configure_cluster` Command

The `configure_cluster` command launches a cluster configuration tool. It allows you to perform the following:

- Change the subnet numbers for the various cluster networks
- Change and configure the domain of the cluster (which is likely different than the domain of `eth0` on the system admin controller itself)
- Prompts for the SLES10 SP1 media and directs creation of image repositories which you can use to customize your software image
- Runs a set of commands that allows you to setup the cluster

Information on using this tool is described in the procedure in the following section, see "Installing Software on the System Admin Controller" on page 28.

## 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 RPMs for the system admin controller and the software images for the rack leader controllers (leader nodes), service, and the compute nodes.

**Procedure 2-1** 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 28.

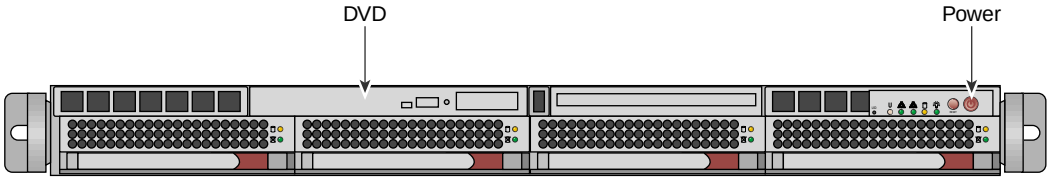


**Figure 2-1** System Admin Controller Power On Button and DVD Drive

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 28.
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.
Whichever you choose, the system will be set up to use the serial console.

Therefore, it is important that you connect to the serial console
the first time you boot the machine after
installation. The first boot after installation,
```

```
you will be prompted for system setup questions on the
serial console.

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 or just wait and the system installation process automatically starts. 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:** 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. The system comes up with console output going to the serial console.

   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:** You must connect to the serial console for this boot to answer the firstboot questions, starting with **Welcome** screen as shown in Figure 2-2 on page 30. If you connect to the serial console too late, you can enter `Ctrl -l` to re-draw the welcome screen.

---

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 30. 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.

---

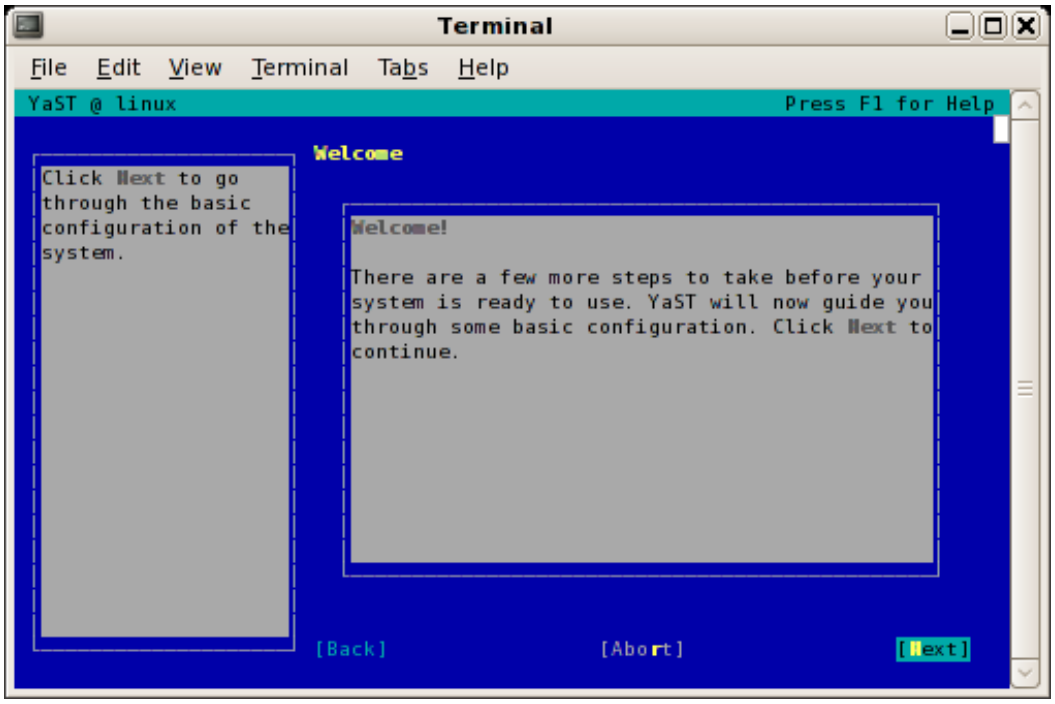


**Figure 2-2** YaST **Welcome** Screen

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 31, 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.

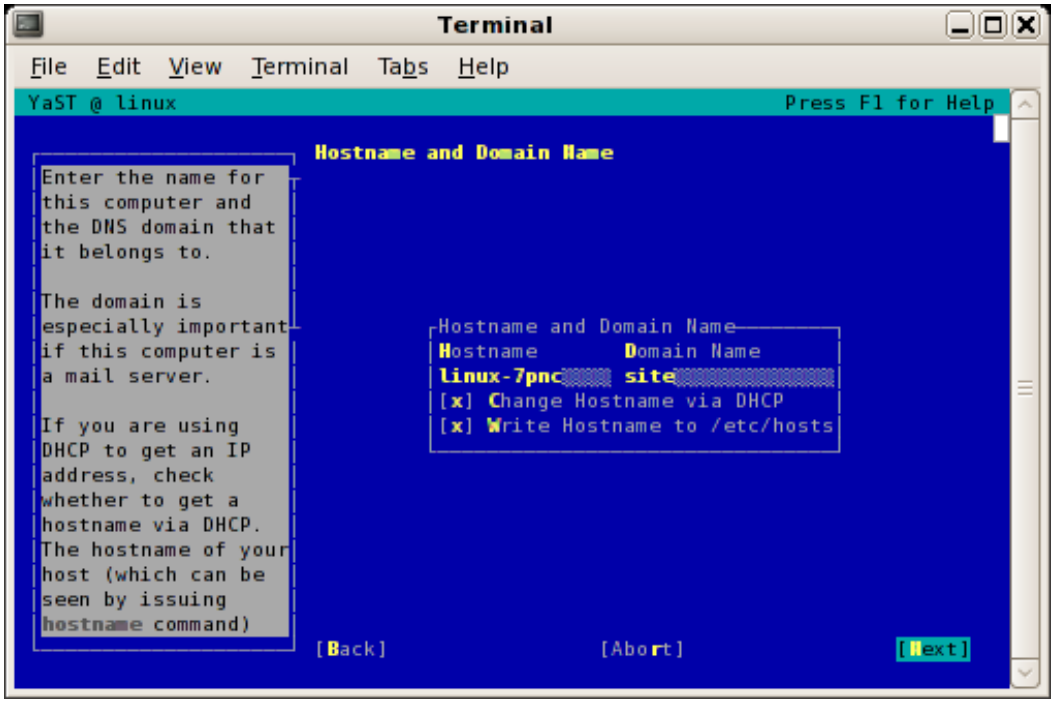


**Figure 2-3 Hostname and Name Server Configuration** Screen

**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 32. Click **Next** to continue.

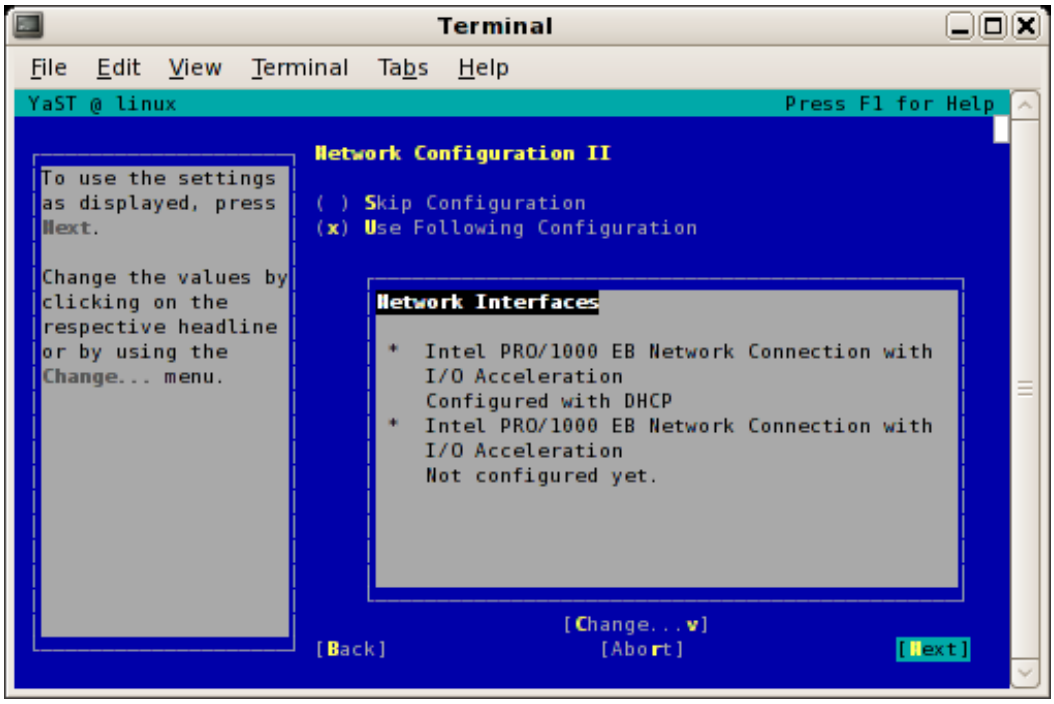


**Figure 2-4 Network Card Configuration Interfaces** Screen

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.

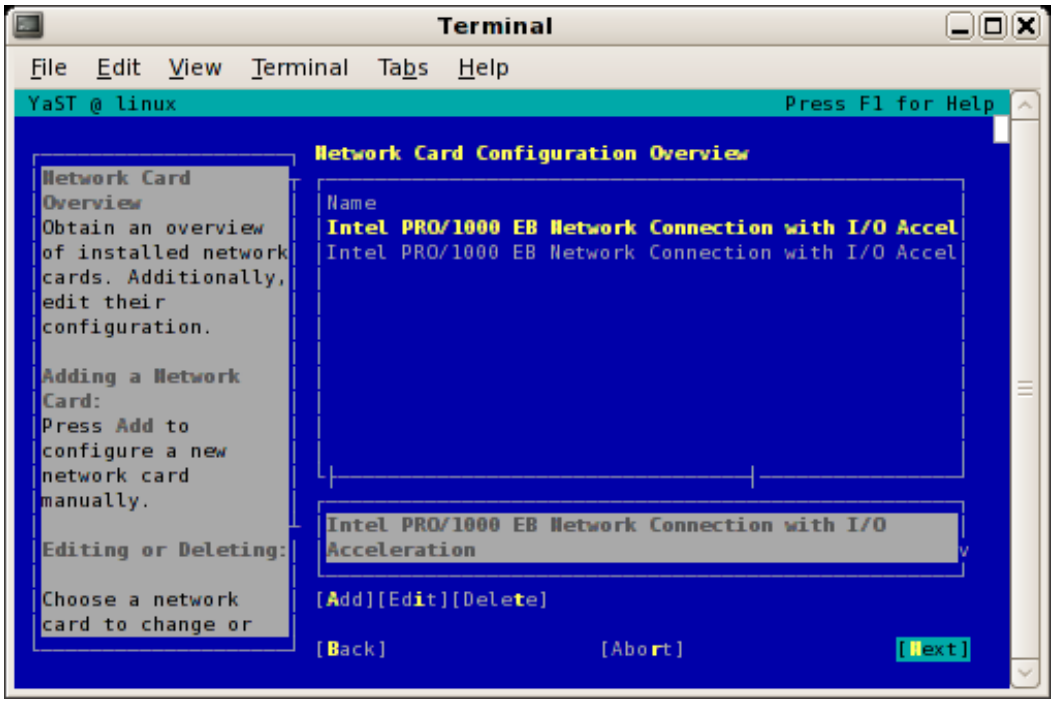


**Figure 2-5 Network Card Configuration Overview** Screen

**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.

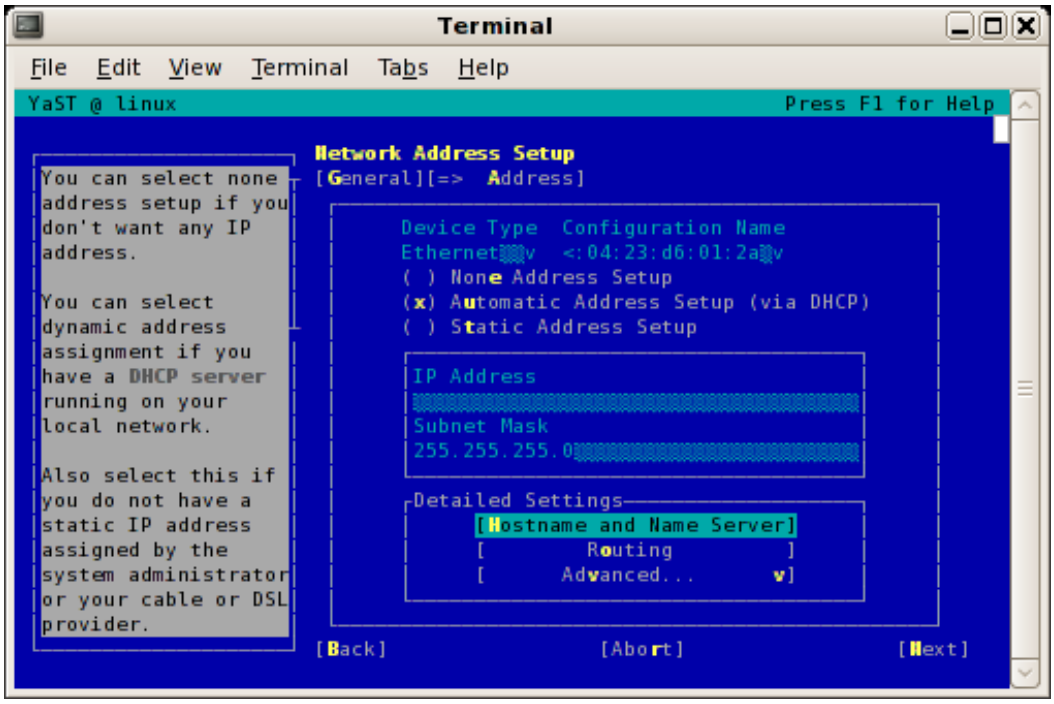


**Figure 2-6 Network Address Setup** Screen

11. From the **Hostname and Name Server Configuration** screen, enter the name and DNS domain name as shown in Figure 2-7 on page 35. Note that the hostname was entered in step 7.

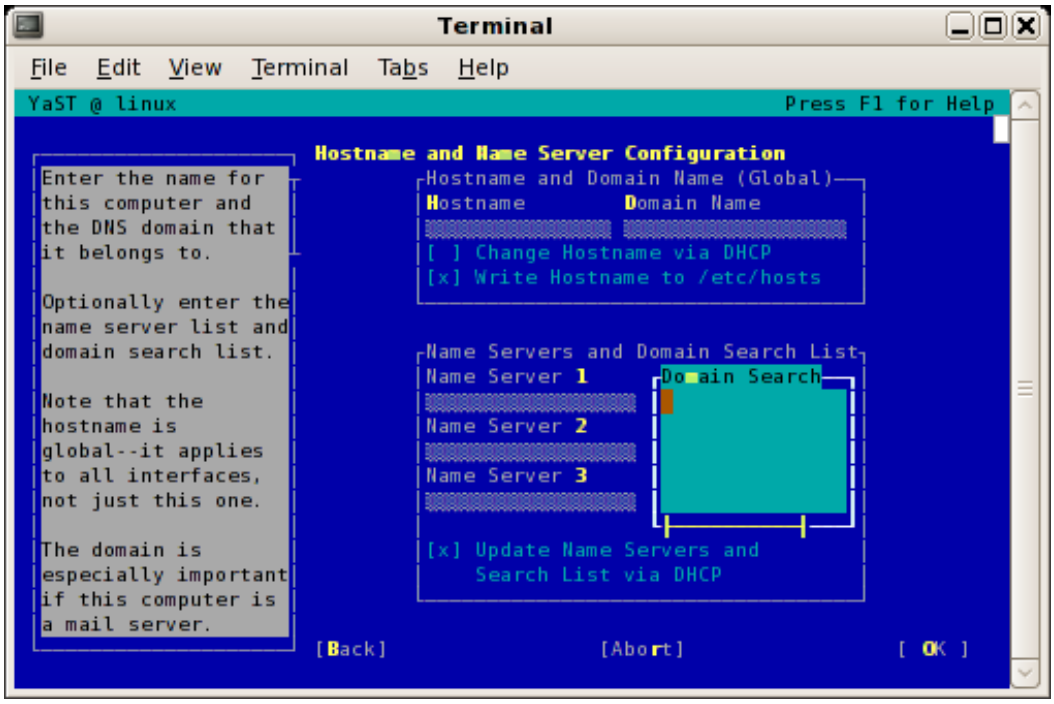


**Figure 2-7** Hostname and Name Server Configuration Screen

12. From the **Routing Configuration** screen, enter the appropriate gateway address and netmask. Click on the **Next** 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.
15. Select the authentication method to use for the users on your system. Click on the **Next** 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 36. Click on the **Finish** button.

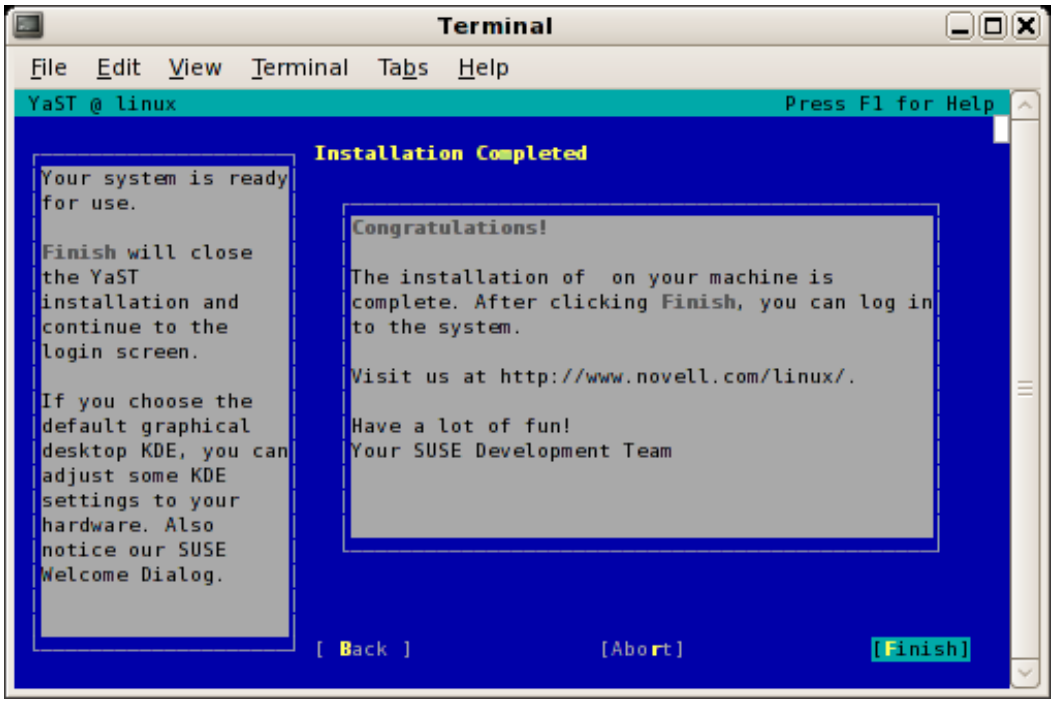


**Figure 2-8 Installation Competed** Screen

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

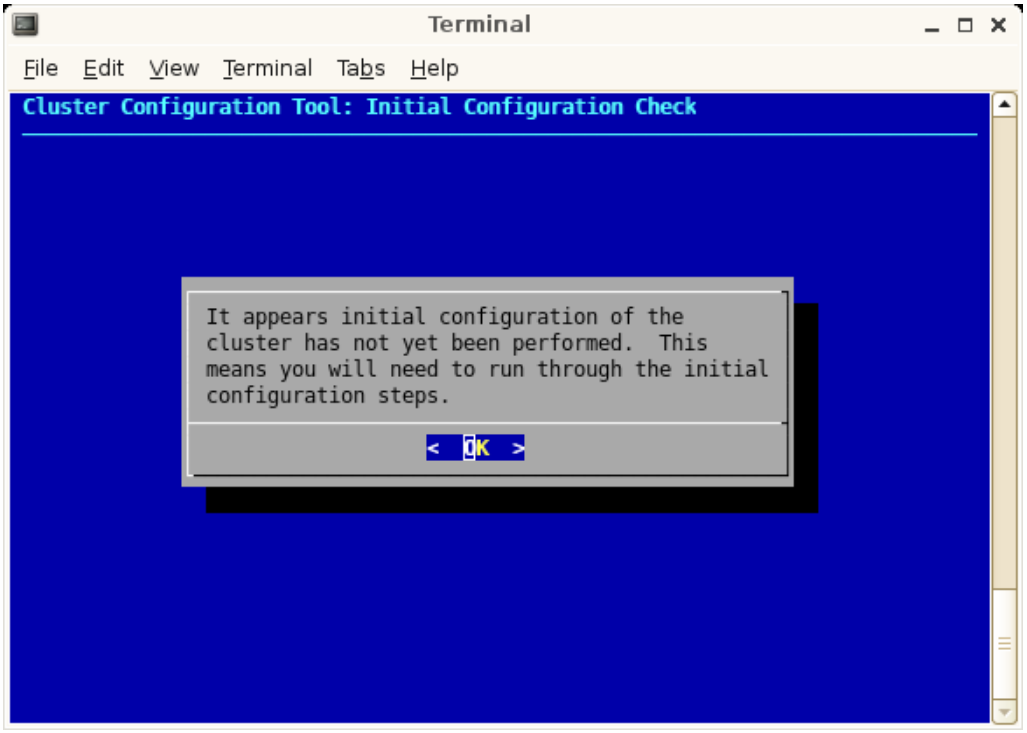


**Figure 2-9 Cluster Configuration Tool: Initial Configuration Check** Screen

22. The **Cluster Configuration Tool: Initial Cluster Setup** screen appears, as shown in Figure 2-10 on page 38. Read the notice and then click **OK** to continue.

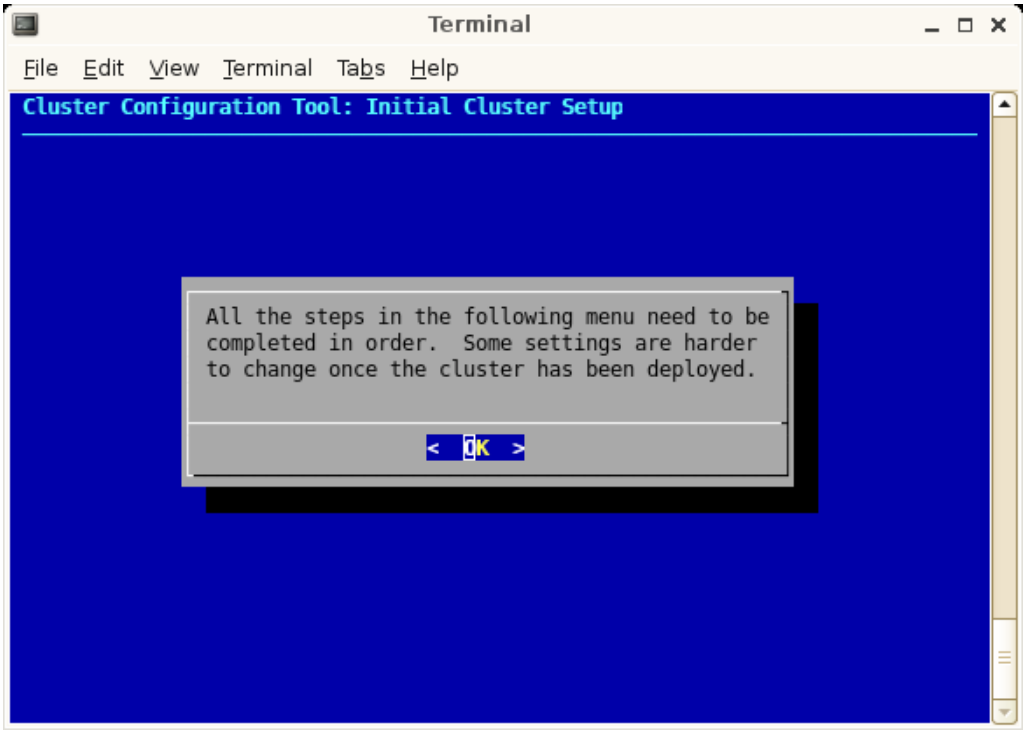


**Figure 2-10 Cluster Configuration Tool: Initial Cluster Setup** Screen

---

**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. Copy the RPMs from your local SLES media.

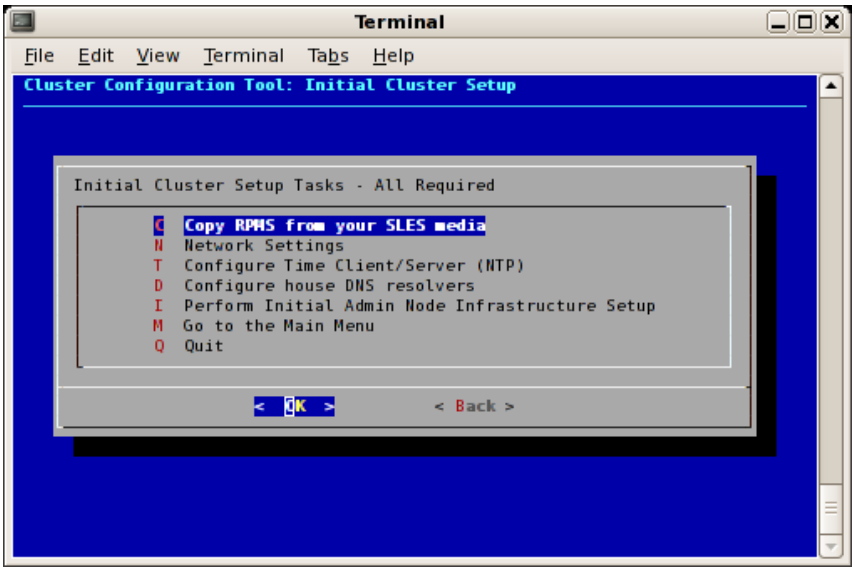


**Figure 2-11 Initial Cluster Setup Tasks** Screen

Insert the SLES10 SP1 DVD into the system admin controller DVD drive and mount it. Use the following command:

```
% mount /dev/dvd /mnt
```

Click **OK** to continue.

24. The first of three **Copy RPMS** screens appears, as shown in Figure 2-12 on page 40. Click **Yes** to continue.

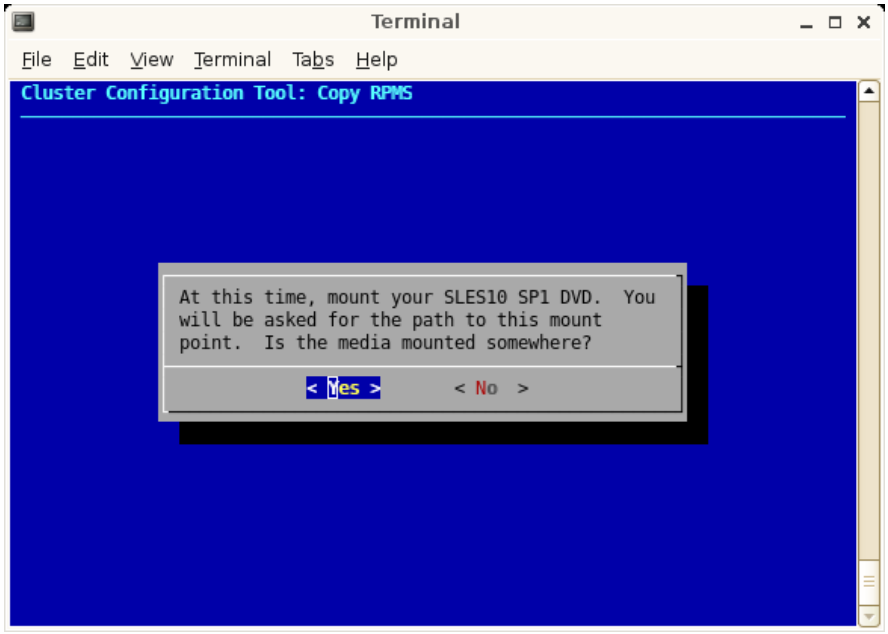


**Figure 2-12 Copy RPMS** Sreen One

25. Select **Network Settings** from the **Install Cluster Setup Tools menu...."**
26. Enter the **/mnt** directory to browse its contents, as shown in Figure 2-13 on page 41. Make sure the RPMs have been successfully copied. Click **OK** to continue.

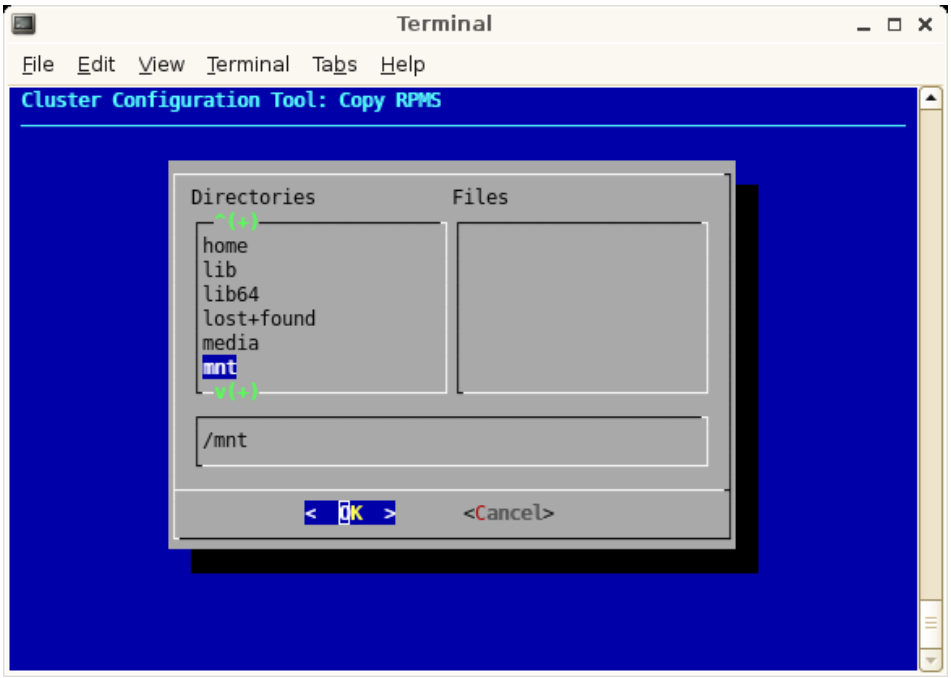


**Figure 2-13 Copy RPMS** Sreen Two

27. The Copy of RPMS from media complete message appears, as shown in Figure 2-14 on page 42. Click **OK** to continue.

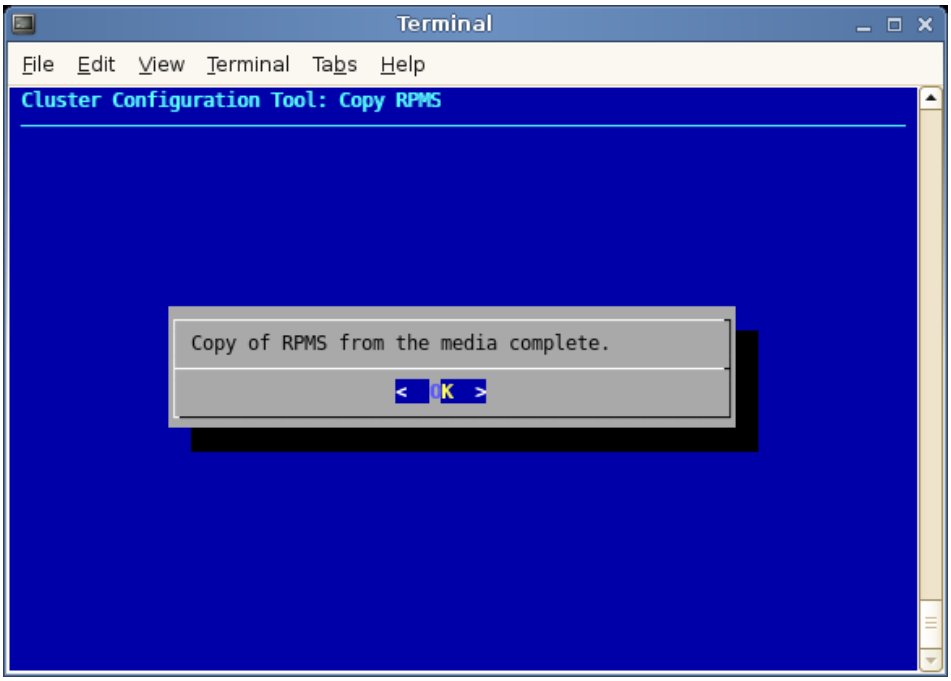


**Figure 2-14 Copy RPMS** Screen Three

28. The **Cluster Network Setup** screen appears, as shown in Figure 2-15 on page 43.

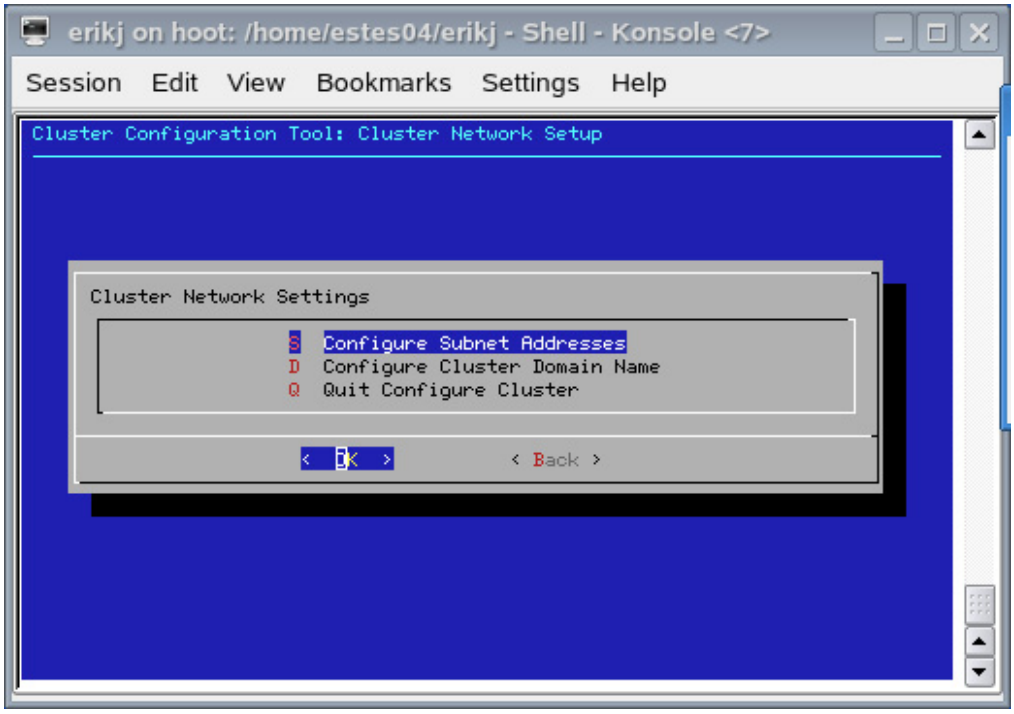


**Figure 2-15 Cluster Network Setup** Screen

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 30. A warning screen appears, as shown in Figure 2-16 on page 44.

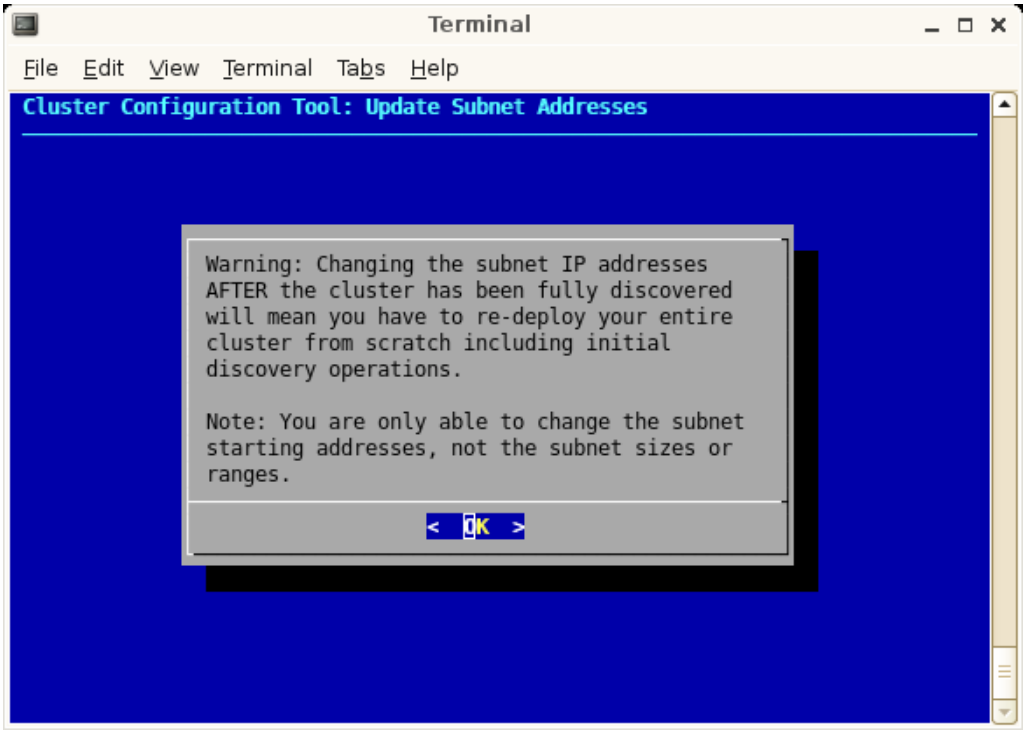


**Figure 2-16 Update Subnet Address** Warning Screen

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.

29. The **Update Subnet Addresses** screen appears, as shown in Figure 2-17 on page 45.

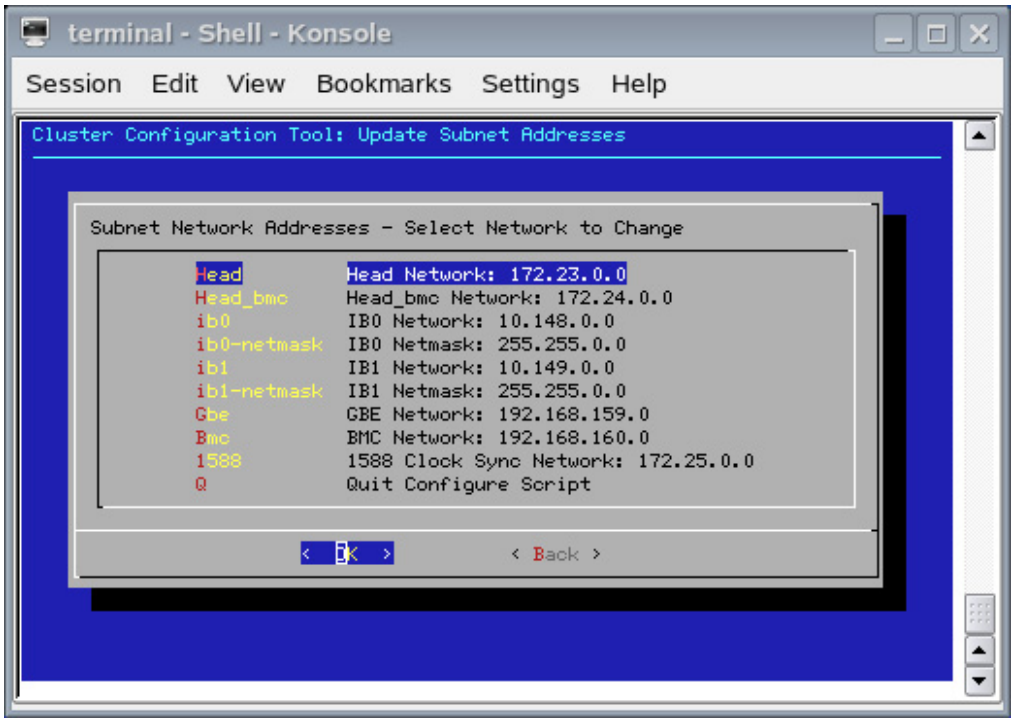


**Figure 2-17 Update Subnet Addresses** Screen

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.

30. Enter the domain name for your Altix ICE system, as shown in Figure 2-18 on page 46. Click **OK** to continue (this will be a subdomain to your house network, by default).

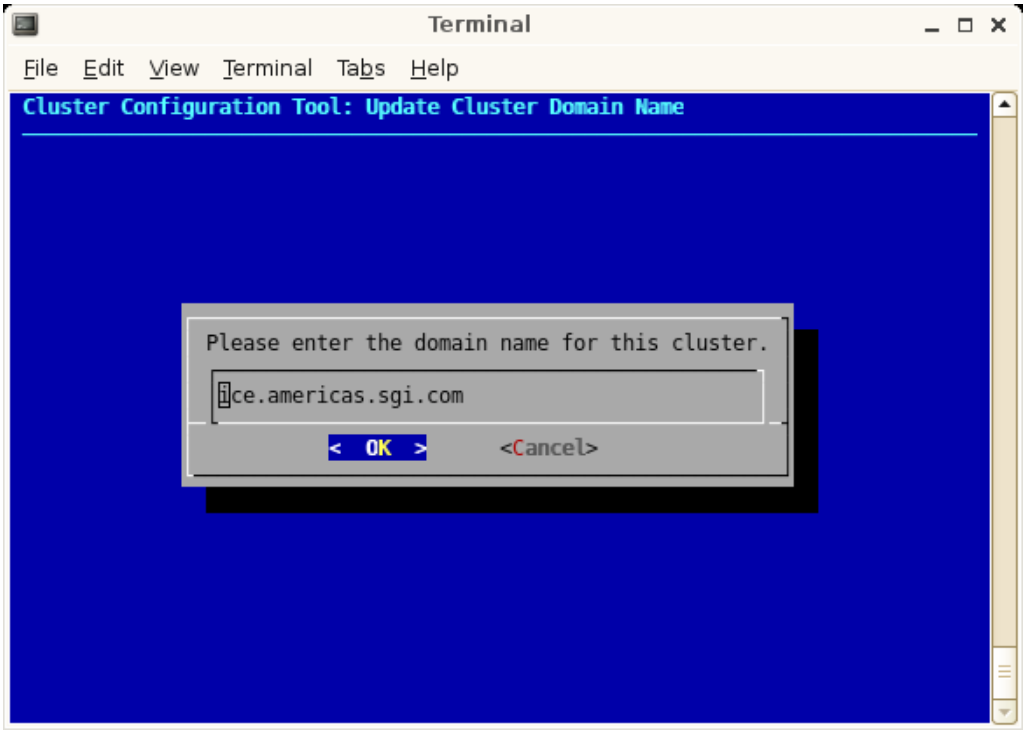


**Figure 2-18 Update Cluster Domain Name** Screen

31. The next steps in this procedure changes your NTP configuration file. Click on **Yes** 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.

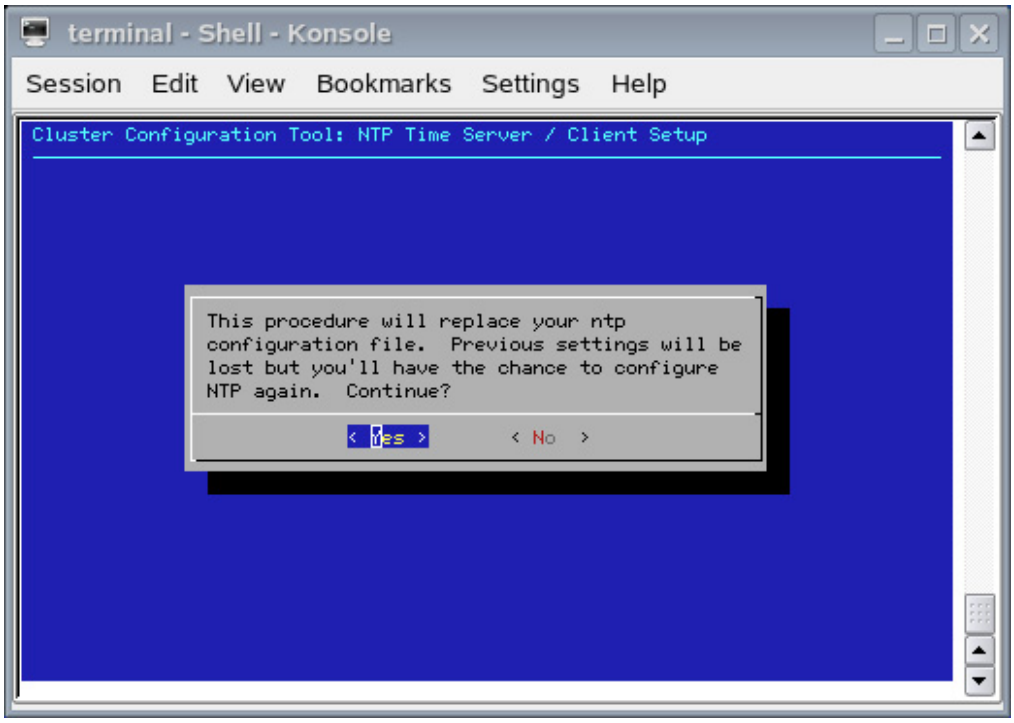


**Figure 2-19 NTP Time Server/Client Setup** Screen One

32. Configure NTP time service as shown in Figure 2-20 on page 48. Click **Next** to continue.

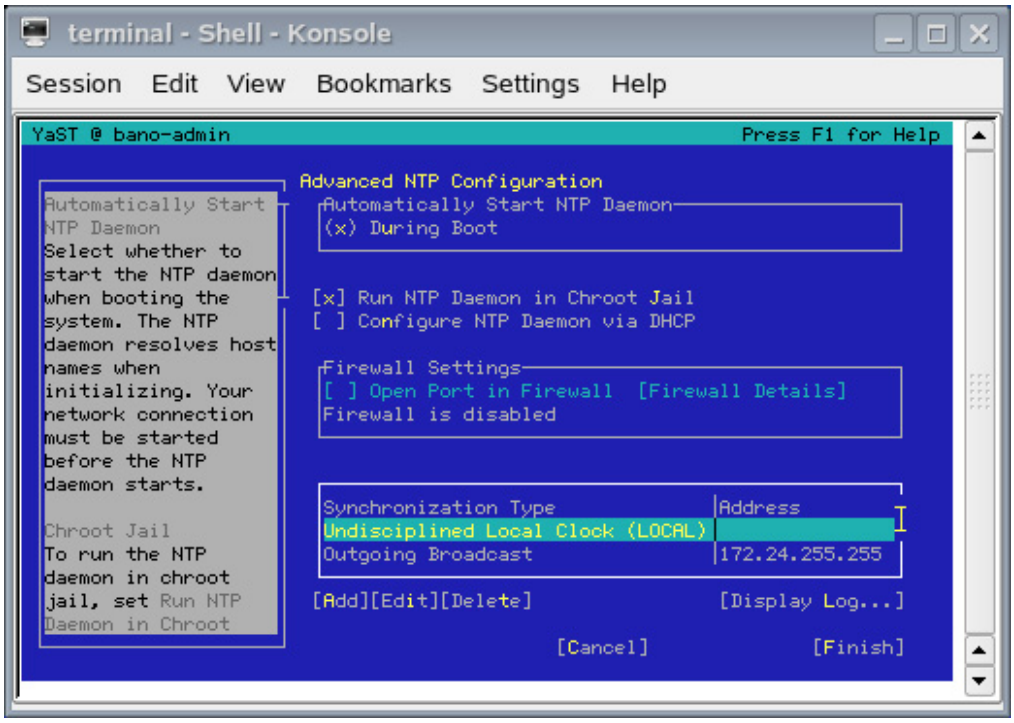


**Figure 2-20 Advance NTP Configuration** Screen

33. A new `ntp.config` configuration file is created. Click on **OK** to continue.

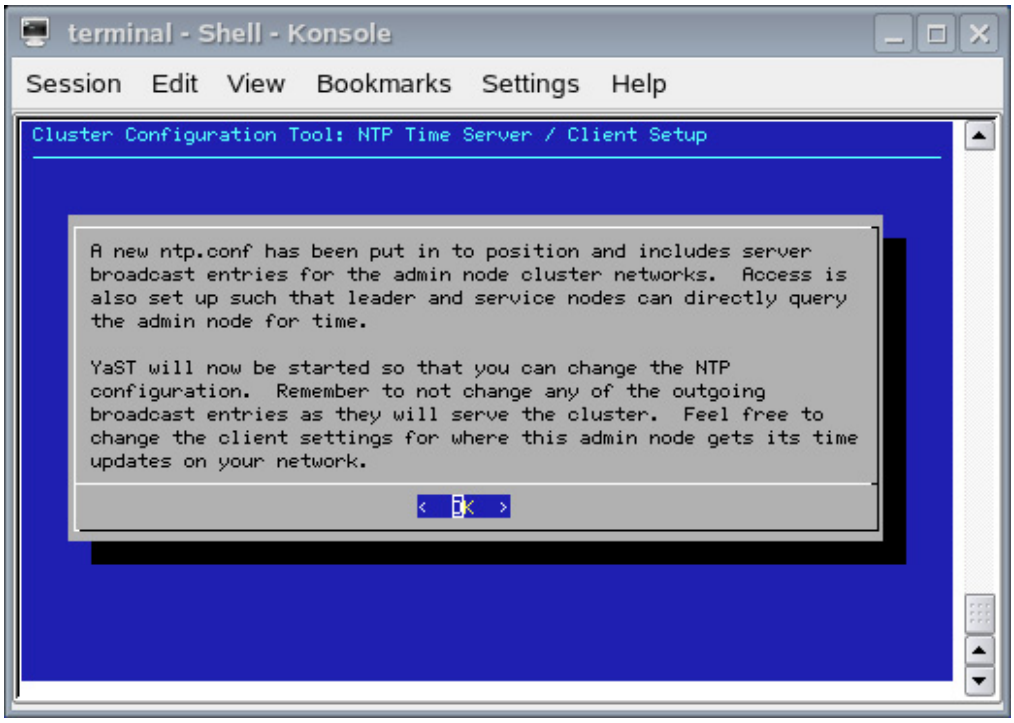


**Figure 2-21 NTP Time Server/ Client Setup** Screen Two

34. Optionally, configure the house domain name service (DNS) resolvers as shown in Figure 2-22 on page 50. 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).

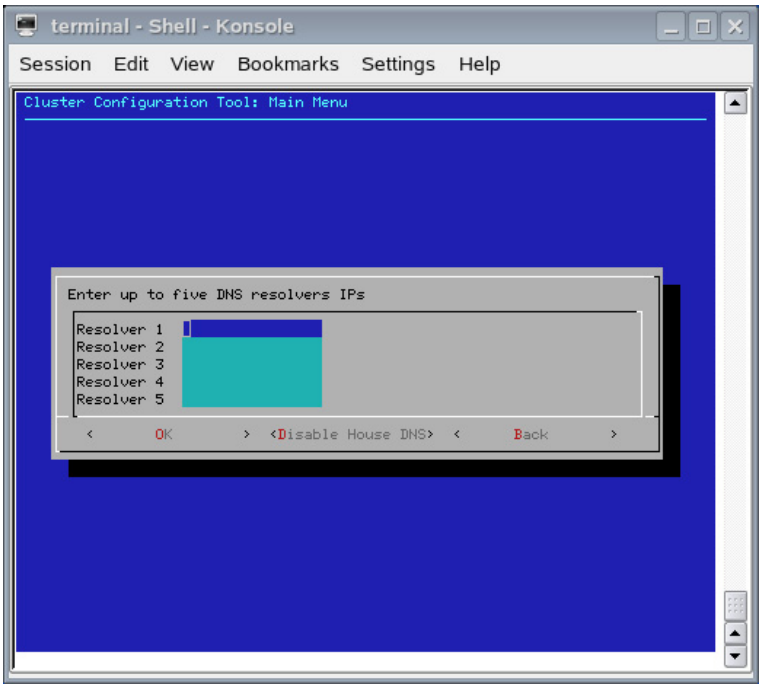


**Figure 2-22 Enter up to Five DNS Resolvers** Screen

35. When the **Admin Infrastructure One Time Setup** screen appears, as shown in Figure 2-18 on page 46, a series of scripts now will run to configure the system admin controller of the Altix ICE system. Click **OK** to continue.

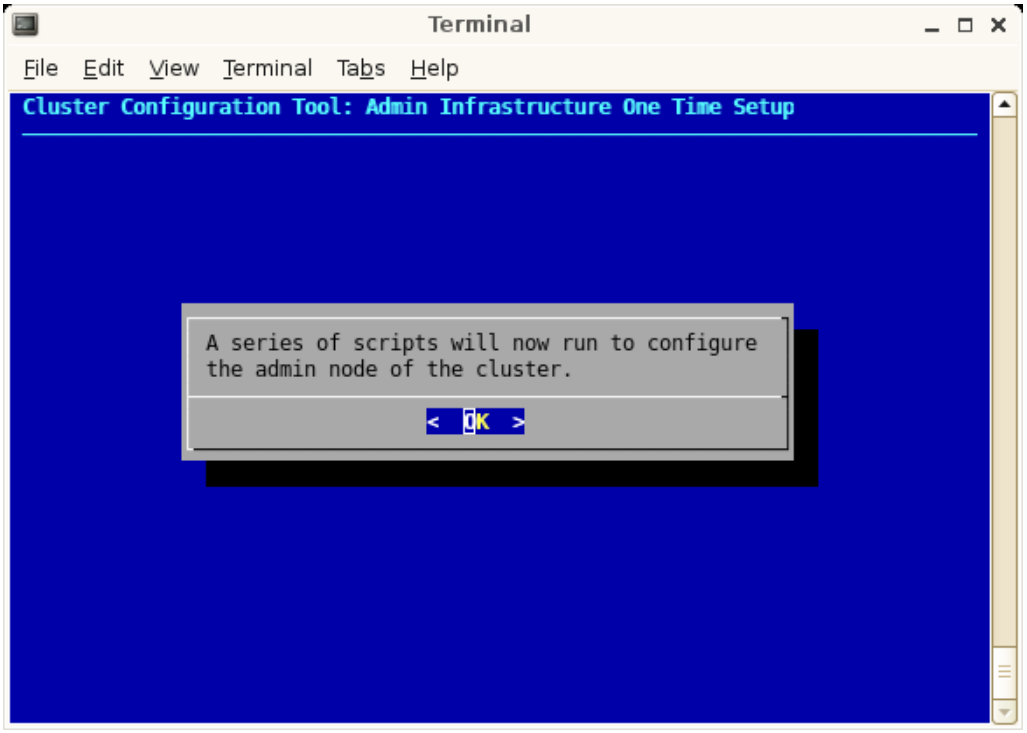


**Figure 2-23 Admin Infrastructure One Time Setup** Screen One

36. Once the scripts have completed configuring the system admin controller, a completion message appears, as shown inFigure 2-24 on page 52. Click **OK** to continue.

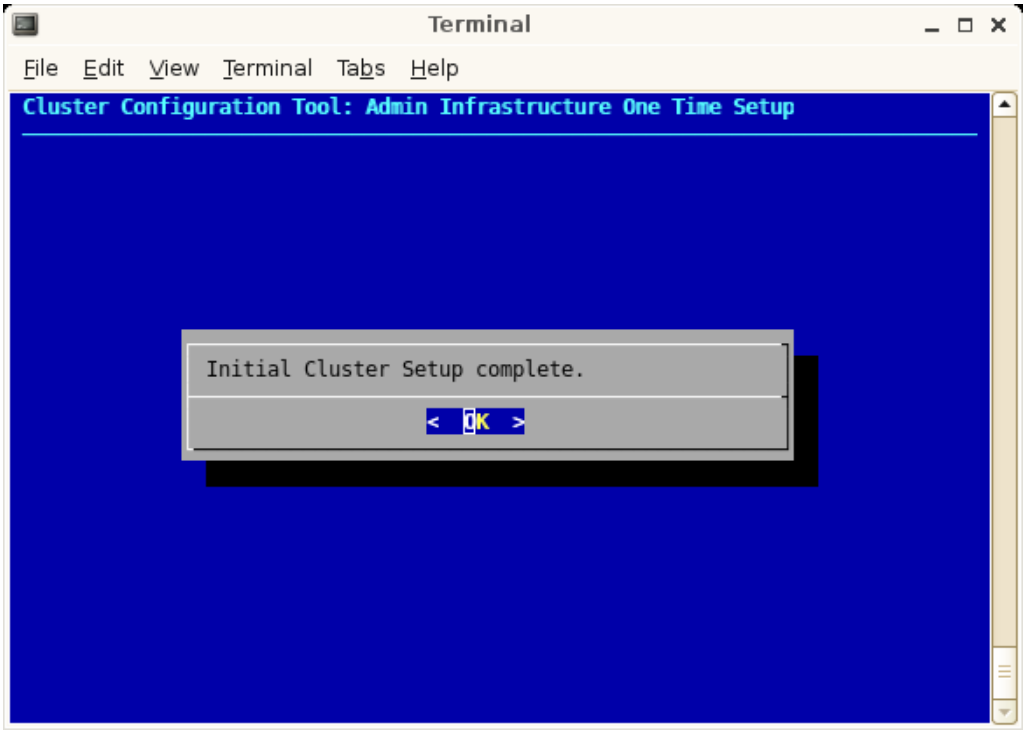


**Figure 2-24 Admin Infrastructure One Time Setup** Screen Two

---

**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.

---

37. Proceed to "Installing Software on the Rack Leader Controllers and Service Nodes" on page 54. It describes the discovery process for the rack leader controllers in your system and how to install software on the rack leader controllers.

## `discover` Command

The `discover` command is used to discover rack leader controllers (leader nodes), service nodes, including the their associated BMC controllers, and compute nodes in

an entire system or in a set of one or more racks that you select. Rack numbers generally start at one. Service nodes generally start at zero. 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 54).

The `discover` command is, as follows:

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

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. |
| `--delrack` | Deletes racks and associated leaders and blades |
| `--delservice` | Deletes a service node |
| `--help` | Usage and help text |

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`

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
```

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

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 1
```

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

```
# /opt/sgi/sbin/discover --rackset 1,5 --service 0 --service 1,altix450 --service 2,other
```

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

## Installing Software on the Rack Leader Controllers and Service Nodes

The `discover` command, described in "`discover` Command" on page 52, 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 2-2** 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 and service nodes using 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.

3. Using this method, 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.

4. You can use the `ssh` command to verify `r1lead` node is available, as follows:

   ```
   # ssh r1lead hostname
   r1lead
   ```

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 58.

## `discover-rack` Command

**Note:** Before you run the `discover-rack` command, make sure the rack leader controllers (leader nodes) have booted and are up.

The `discover-rack` command invokes a shell that calls various scripts to populate the cluster database with the rack hardware MAC addresses, generate the `hosts`, `dhcpd.conf` files, the Cluster Command & Control (C3) software configuration files and Ganglia real-time cluster monitoring software configuration files. For information on how to use this command, see "Discovering Compute Nodes" on page 58.

In addition, racks can be rediscovered by running the `discover-rack` command again on a previously discovered rack. Because the `discover-rack` command turns off the power for all blades, you need to power on the rack again. When powering up the rack it is not necessary to power up the rack leader controller because it is already on. SGI recommends that you avoid power cycling the rack leader controller.

**EXAMPLES**

**Example 2-2** `discover-rack` Command Examples

To discover rack 1, perform the following:

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

## Discovering Compute Nodes

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

**Procedure 2-3** Discovering Compute Nodes

To discover compute nodes (blades) in your Altix ICE system, complete the steps in "Installing Software on the Rack Leader Controllers and Service Nodes" on page 54. Then run perform this procedure for each rack in your system:

---

**Note:** Some of the output shown in this example has been modified to fit the format of this manual.

---

1. Run the `discover-rack` command for each rack in your system from the system admin controller, as follows:

```
system-admin:~ # /opt/sgi/sbin/discover-rack --rack 1
/opt/sgi/sbin/discover-rack: Running [ssh r1lead discover-blades|cat > /tmp/slot_file_1]
to discover the IRU in rack 1
/opt/sgi/sbin/discover-rack: Running [populate-db-rack --rack 1] to populate the DB with rack 1
/opt/sgi/sbin/discover-rack: Running [generate-leader-hostfile --rack 1] to generate the hosts file
for the leader for rack 1
/opt/sgi/sbin/discover-rack: Running [generate-leader-dhcpfile --rack 1] to generate the dhcpd.conf file
for the leader for rack 1
Shutting down DHCP server ..done
Starting DHCP server [chroot]..done
/opt/sgi/sbin/discover-rack: Running [generate-admin-c3-file] to generate
the c3 configuration file for the admin
/opt/sgi/sbin/discover-rack: Running [generate-leader-c3-file --rack 1]
to generate the c3 configuration file
for the leader for rack 1
/opt/sgi/sbin/discover-rack: Running [generate-service-c3-file] to generate
the c3 configuration file
for all service nodes
/opt/sgi/sbin/discover-rack: Running [generate-leader-ganglia-file --rack 1] to generate
```

```
the Ganglia configuration file
for the leader for rack 1
Shutting down gmond..done
Starting gmond..done
/opt/sgi/sbin/discover-rack: Running [generate-admin-ganglia-files] to generate
the Ganglia configuration files for the admin node
Use of $# is deprecated at /opt/sgi/sbin/generate-admin-ganglia-files line 344.
Use of uninitialized value in concatenation (.) or string
at /opt/sgi/sbin/generate-admin-ganglia-files line 344.
Shutting down gmond                                                  done
Starting gmond                                                       done
Shutting down gmetadsaving /dev/shm/rrds to /var/lib/ganglia/snaps/snap.tar.gz
                                                                     done
Starting gmetadrrd directory already exists in /dev/shm
snaps directory already exists in /var/lib/ganglia
restoring /dev/shm/rrds from /var/lib/ganglia/snaps/snap.tar.gz
                                                                     done
/opt/sgi/sbin/discover-rack: Running [generate-admin-dns-zonefile]
to generate DNS zone files for the admin node
Shutting down name server BIND  waiting for named to shut down (28s)  done
Starting name server BIND                                            done
/opt/sgi/sbin/discover-rack: Running [generate-conserver-files]
to generate conserver.cf files for admin and leader node
Reloading conserver: ..done
Reloading conserver:                                                 done
/opt/sgi/sbin/discover-rack: Running [push-and-set-default-compute-image --rack 1]
to push default compute node image to rack 1 and set new blades to boot it.
```

At this point, the compute nodes (blades) are ready to be powered up for the first time. They are configured to use the default compute node software image. For information on how to customize the compute node software images for your site, see "Customizing Compute Node Software" on page 80.

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 103.

---

**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 103.

---

# 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 2-4** 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 --add-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 83 and "Customizing Compute Node Software" on page 80.
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 `/etc/resolv.conf` or `/etc/named.conf` files. Additionally, the default gateway addresses for the service node may be a good choice. You can use the `netstat -rn` command for this information, as follows:

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

If the `ping` command executed from the service node to the selected IP address gets responses, network monitoring tools such as `tcpdump`(1) should be used. On the service node, monitor the `eth1` interface and simultaneously in a separate session

monitor the `ib[01]` interface. You should specify monitoring specific-enough to not have additional noise then attempt execute a `ping` command from the compute node.

**Example 2-3** `tcpdump` Command Examples

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

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

- Check the system admin controller's compute image configuration of the default route
- Verify that this image has been pushed to the compute nodes
- Verify that the compute nodes have booted with this image

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

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

## Service Node Configuration for Gateway Operation

You may chose to connect your compute nodes using routable addresses on the house network. This requires planning before the installation by reserving a large block of routable 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 2-5** Service Node Configuration for Gateway Operation

To connect your compute nodes using routable 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 routable range to the IB fabric(s) you desire.

   You can make either `ib0`, `ib1`, or both routable 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 "Installing Software on the System Admin Controller" on page 28.

## 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 2-6** 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:

```
system-1-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 61.

## 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 2-7** 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 83 and "Customizing Compute Node Software" on page 80).

---

**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 61.

**Procedure 2-8** 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 61.

**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 63.

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.

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

- The Tempo installation procedure overwrites data at `/dev/sda`. Keep `/dev/sda` exclusively for use by the system.
- Most administrators use `/dev/sdb` to house the home directories. Depending on your hardware, these devices may be single disks or RAIDs.
- This example uses `sdb` name here but sometimes disk ordering can be complicated. It is a good idea to ensure `sdb` matches the disk you really wish to use, then use filesystem LABELs to ensure the correct filesystem is mounted regardless of what letter it has.

## Partitioning, Creating, and Mounting Filesystems

**Procedure 2-9** Partitioning and Creating Filesystems for an NFS Home Server on a Service Node

**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/sdb`. The following example makes one filesystem out of the disk. You can use `parted` utility interactively or in a command-line driven manner.
2. Make a new `msdos` label, as follows:

   ```
   # parted /dev/sdb mklabel msdos
   ```

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

   ```
   # parted /dev/sdb print
   Disk geometry for /dev/sdb: 0kB - 500GB
   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/sdb mkpart primary ext2 0 500GB
   ```

5. 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 -L mylabel -N 20000000 /dev/sdb1
```

An xfs example is, as follows:

```
# mkfs.xfs -L mylabel /dev/sdb1
```

---

**Note:** SGI suggests using a label in this step with the -L parameter, as shown here in both examples.

---

6. Issue the following command to cause the /dev/disk/by-label device to be ready for use immediately and avoid rebooting after creating your home filesystem:

   ```
   # udevtrigger
   ```

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.

   ---

   **Note:** If you are using XFS, replace ext3 with xfs. Note here the LABEL= is used and not an actual device name.

   ---

   ```
   LABEL=mylabel       /home   ext3    defaults        1       2
   ```

   b. Add the /home filesystem to /etc/exports.

   ---

   **Note:** You may wish to use a more secure export than shown here. See the exports(5) man page for information.

   ---

   ```
   /home *(rw,sync,mountpoint=/home,no_subtree_check)
   ```

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

   ```
   # chkconfig nfsserver on
    # rcnfsserver restart
   ```

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 "Creating a Simple Compute Node Image Clone" on page 83.

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-sles10sp1-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:

```
service-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 --add-rack compute-sles10sp1-clone "r*"
```

Using `add-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 83.

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 70. 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 66. In this example, you need to replace `service-ib1` with the `ib1` InfiniBand host name for the NAS server and you need to know where on the NAS server the home filesystem is mounted to craft the `sgi-fstab` script properly.

## Setting Up a NIS Server for Your Altix ICE System

This section describes how to set up a network information service (NIS) server running 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 64. 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.
- NIS synchronization traffic between NIS master and slave servers (leader nodes) goes over Infiniband connections when NIS maps are adjusted and pushed out.
- Service node NIS (besides NIS master) traffic goes over InfiniBand because of how host name resolution works.
- 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 2-10** Setting Up a Service Node as a NIS master

To set up a service node as a NIS master, 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 78.

   ---

5. Select **Add** and enter **r1lead-ib1** in the **Edit Slave** window. Enter any other rack leader controllers you may have just like above. Click on **Next** to continue.

   ---

   **Note:** This example uses **r1lead-ib1** because **r1lead** would not resolve to anything on a service node.

   ---

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 **Next** 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 71.

---

**Procedure 2-11** 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. Set up the service node to broadcast bind by creating this simple `yp.conf` file, as follows:

   ```
   # echo "broadcast" > /etc/yp.conf
   ```

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. One set of instructions uses YaST, the other uses a set of commands that could be scripted if you so choose. 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 2-12** Setting up a Rack Leader Controller as a NIS Slave Server and Client

Use the following set of commands 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.

---

```
# cexec --head chkconfig ypserv on
   # cexec --head chkconfig ypbind on
   # cexec --head chkconfig portmap on
   # cexec --head chkconfig nscd on
   # cexec --head rcportmap start
   # cexec --head "echo ice > /etc/defaultdomain"
   # cexec --head "ypdomainname ice"
   # cexec --head "echo ypserver 127.0.0.1 > /etc/yp.conf"
   # cexec --head "echo +::: >> /etc/group"
   # cexec --head "echo +:::::: >> /etc/passwd"
   # cexec --head "echo + >> /etc/shadow"
   # cexec --head /usr/lib/yp/ypinit -s service0
   # cexec --head rcportmap start
   # cexec --head rcypserv start
   # cexec --head rcypbind start
   # cexec --head 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 2-13** 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 "Creating a Simple Compute Node Image Clone" on page 83.
2. Change the compute nodes to use the cloned image/kernel pair, as follows:

```
# cimage --set compute-sles10sp1-clone 2.6.16.46-0.12-smp "r*i*n*"
```

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

```
# echo "ice" > /var/lib/systemimager/images/compute-sles10sp1-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:

```
# echo "ypserver lead-eth" > /var/lib/systemimager/images/compute-sles10sp1-clone/etc/yp.conf
```

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

```
# chroot /var/lib/systemimager/images/compute-sles10sp1-clone chkconfig ypbind on
```

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

```
# echo "+:::" >> /var/lib/systemimager/images/compute-sles10sp1-clone/etc/group
# echo "+::::::" >> /var/lib/systemimager/images/compute-sles10sp1-clone/etc/passwd
# echo "+" >> /var/lib/systemimager/images/compute-sles10sp1-clone/etc/shadow
```

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

```
# cimage --add-rack compute-sles10sp1-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` network 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_customizations/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 < $iruslot/etc/sysconfig/network/ifcfg-vlan1
# ifcfg config file for vlan ib1
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='${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
```

## 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 73.

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
```

## 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 2-14** 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
   ```

# 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:



## Compute Node Software

This section describes SLES10 services turned off on compute nodes by default, how you can customize the software running on compute nodes, create a simple clone image of compute node software, and how to use the `cimage` command. 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 (SP1) 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

## Customizing Compute Node Software

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 blades, 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 leaders. 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
#
# 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.  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

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

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

**Procedure 3-1** Customizing a Compute Node Image

To customize the compute node operating system image, perform the following steps:

1.

---

**Note:** How to create a clone of the compute node image is also described in "Creating a Simple Compute Node Image Clone" on page 83.

---

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

```
# cimage ---clone-image compute-sles10sp2 new
```

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

```
# cimage --list-images
  image: compute-sles10sp1
          kernel: 2.6.16.46-0.12-carlsbad
          kernel: 2.6.16.46-0.12-smp

  image: compute-sles10sp1-clone
          kernel: 2.6.16.46-0.12-carlsbad
          kernel: 2.6.16.46-0.12-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 a file called `new`. To install them into your new compute node image, perform the following commands:

   ```
   # chroot new bash
   ```

   And then:

   ```
   # rpm -Uvh /tmp/newrpm.rpm
   ```

## Creating a Simple Compute Node Image Clone

This section describes how to make a copy of a compute node image.

**Procedure 3-2** Creating A Simple Compute Node Image Clone

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:

   ```
   # cimage --clone-image compute-sles10sp1 compute-sles10sp1-clone
   ```

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

   ```
   # cimage --list-images
     image: compute-sles10sp1
            kernel: 2.6.16.46-0.12-carlsbad
            kernel: 2.6.16.46-0.12-smp

     image: compute-sles10sp1-clone
            kernel: 2.6.16.46-0.12-carlsbad
            kernel: 2.6.16.46-0.12-smp
   ```

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

   ```
   # cimage --set compute-sles10sp1-clone 2.6.16.46-0.12-smp "r*i*n*"
   ```

## `cimage` Command

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

The `cimage` command accepts the following options:

| Option | Description |
|---|---|
| `--help` | Usage and help text |
| `--list-images` | Lists images present in the database |
| `--list-nodes RACK ...` | Lists what compute nodes are set to |
| `--set IMAGE KERNEL NODE ...` | Sets the compute nodes to a certain boot image and kernel combination |
| `--add-db IMAGE` | Adds an image to the database |
| `--del-db IMAGE` | Deletes an image from the database |
| `--add-rack IMAGE RACK ...` | Pushes an image to specified rack(s) |
| `--del-rack IMAGE RACK` | Deletes an image from specified rack(s) |
| `--clone-image OIMAGE NIMAGE` | Clones an existing image to a new image |
| `--del-image IMAGE` | Deletes an existing image entirely |

`RACK` arguments take the format `rX`.

`NODE` arguments take the format `rXiYnZ`.

`X`, `Y`, `Z` can be single digits, a [start-end] range, or * for all matches.

`...` indicates more than one `RACK` or `NODE` argument can be passed in.

**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-sles10sp1
        kernel: 2.6.16.46-0.12-carlsbad
        kernel: 2.6.16.46-0.12-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-sles10sp1 2.6.16.46-0.7-smp
r1i0n1: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n2: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n3: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n4: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n5: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n6: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n7: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n8: compute-sles10sp1 2.6.16.46-0.7-smp
[...snip...]
```

To set the `r1i0n0` compute node to boot the `2.6.16.46-0.12-carlsbad` kernel from the `compute-sles10sp1` image, perform the following: :

```
# cimage --set compute-sles10sp1 2.6.16.46-0.12-carlsbad 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-sles10sp1 2.6.16.46-0.7-carlsbad
r1i0n1: compute-sles10sp1 2.6.16.46-0.7-smp
r1i0n2: compute-sles10sp1 2.6.16.46-0.7-smp
[...snip...]
```

To set all nodes in all racks to boot the `2.6.16.46-0.7-carlsbad` kernel from the `compute-sles10sp1` image, perform the following:

```
# cimage --set compute-sles10sp1 2.6.16.46-0.7-carlsbad r*i*n*
```

To set two ranges of nodes to boot the 2.6.16.46-0.7-smp kernel, perform the following:

```
# cimage --set compute-sles10sp1 2.6.16.46-0.7-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the compute-sles10sp1 image to a new image (so that you can modify it) , perform the following:

```
# cimage --clone-image compute-sles10sp1 mynewimage
Cloning compute-sles10sp1 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/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// 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 --del-db mynewimage
# cimage --add-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 --add-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-sles10sp1
        kernel: 2.6.16.46-0.7-carlsbad
        kernel: 2.6.16.46-0.7-smp

image: mynewimage
        kernel: 2.6.16.46-0.7-carlsbad
        kernel: 2.6.16.46-0.7-smp
```

To set some compute nodes to boot an image, perform the following:

```
# cimage --set mynewimage 2.6.16.46-0.7-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.
```

## 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. |
| `--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

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: `--shutdown`, `--boot`, and `--reboot`. The `--shutdown` option is self-explanatory, but `--reboot` will ensure 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.

---

**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, but a node that has been shutdown with the `--shutdown` option does not have its power automatically turned off.

---

The `--shutdown` option 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. Usually you may only specify one action per command, however, with the `--shutdown` option, you may also `specify --off`. Using both these actions results in nodes being shutdown, 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 --shutdown --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 --shutdown service0
```

To shutdown and switch off everything in rack 3, perform the following:

```
# cpower --shutdown --off --rack r3
```

---

**Note:** Using the `--shutdown` and the `--off` options together is the only time you can use more than one command on the `cpower` command line. This combination will shutdown 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 --shutdown --system
```

To shutdown all the compute nodes, but not the service nodes, leader nodes, perform the following:

```
# cpower --shutdown --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 87).

**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*>**

## 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 3-3** 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/1                       at  Local
 r1lead                  on /dev/pts/0                       at  Local
```

2. To connect to a console, perform the following:

```
system-admin:~ # console service0
service0 login: root
```

## 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 99
- "Rack Leader controller NTP" on page 100
- "Service Node NTP" on page 100
- "Compute Node NTP" on page 100
- "NTP Work Arounds" on page 100


### System Admin Controller NTP

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 watching NTP broadcasts from the system admin controller. The NTP server 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.

### Service Node NTP

The NTP client on *managed* service nodes ( for a definition of managed, see "`discover` Command" on page 52) 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 instead of 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.

## 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 3-4** 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 systems uses Open Fabrics Enterprise Distribution (OFED) 1.2 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

Fabric management on SGI Altix ICE 8200 systems uses the OFED 1.2 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 19) with the following characteristics:

- Each network fabric has its own subnet manager (SM).
- There is one instance of SM running on the rack leader controller on each rack for each of the two network fabrics, `ib0` and `ib1` (see Figure 1-2 on page 5 and Figure 1-4 on page 12).
- Each instance of SM on the rack leader controller is controlled by the `/etc/opensm-ib0.conf` or `/etc/opensm-ib1.conf` configuration file. For more information, see "`smconfig` Automatic Fabric Configuration Tool" on page 105.

- Each instance of SM on the rack leader controller is either in `Master` or `Standby` state. For more information, see "InfiniBand Fabric Management Configuration and Operation Overview" on page 111 and Figure 4-1 on page 115.
- Rack leader controllers run the `opensm` daemon for each fabric over separate HCA ports (see Figure 1-9 on page 19).

---

**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 be routed incorrectly. After a system reboot, use the `smadmin` command to restart the fabric. For more information, see "`smadmin` InfiniBand Fabric Administration Tool" on page 106 and "Fabric Management and Rebooting" on page 110.

---

- 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.
- Coherency of the fabric database is handled by `sldd-ib[01].sh`. You must make sure `OSM_HOSTS` is configured correctly in the `/etc/opensm-ib0.conf` or `/etc/opensm-ib1.conf` configuration files.

---

**Note:** Currently, the InfiniBand fabric `ib0` is reserved for MVAPICH 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" on page 105
- "`smadmin` InfiniBand Fabric Administration Tool" on page 106
- "Fabric Management and Rebooting" on page 110

### 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 117 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 | Fabric ib0 or fabric ib1 (Required) |
| -o | OSM hosts list (override the default of automatic configuration) |
| -r | Routing engine (override the default of automatic configuration) |
| -l | Select (individual) rack lead (default is ALL rack leads) |

**Note:** The smconfig -o and -r options allows you to override the automatic configuration of the fabric performed by the smconfig command. SGI recommends that you use the smconfig to automatically configure each fabric.

**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:

```
# smconfig -f ib0
Configuring r1lead
Configuring r2lead
Configuring r3lead
Configuring r4lead
```

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

```
# 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 105).

---

**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 be routed incorrectly. After a system reboot, use the `smadmin` command to restart the fabric. For more information, see "Fabric Management and Rebooting" on page 110 and Figure 4-1 on page 115.

---

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 111) |
| `-l` | Select an (individual) rack leader controller (leader node) |
| `-m` | Find `opensmd` MASTER node (see Figure 4-1 on page 115) |

| | |
|---|---|
| `-c` | Attempt a fabric cleanup |

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

To use the `smadmin` command to start, stop, restart or get status about the `ib0` or `ib1` InfiniBand fabric on your system, perform the following:

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

```
# smadmin -f ib0 -u
      opensm is stopped
      opensm is stopped
      opensm start                                              [  OK  ]
      smagent-rack: opensm configuration r1lead: opensmd started on fabric ib0
      Running start on r2lead
      opensm is stopped
      opensm is stopped
      opensm start                                              [  OK  ]
      smagent-rack: opensm configuration r2lead: opensmd started on fabric ib0
      Running start on r3lead
      opensm is stopped
      opensm is stopped
      opensm start                                              [  OK  ]
      smagent-rack: opensm configuration r3lead: opensmd started on fabric ib0
      Running start on r4lead
      opensm is stopped
      opensm is stopped
      opensm start                                              [  OK  ]
      smagent-rack: opensm configuration r4lead: opensmd started on fabric ib0
```

2. To start fabric management on all the rack leader controllers (leader nodes) on the `ib1` fabric, perform the following:

```
# smadmin -f ib1 -u
      Running start on r1lead
      Another fabric has opensm (pid 11004) running...
      Another fabric has opensm (pid 11004) running...
      opensm start                                              [  OK  ]
      smagent-rack: opensm configuration r1lead: opensmd started on fabric ib1
      Running start on r2lead
      Another fabric has opensm (pid 8217) running...
```

```
Another fabric has opensm (pid 8217) running...
opensm start                                               [  OK  ]
smagent-rack: opensm configuration r2lead: opensmd started on fabric ib1
Running start on r3lead
Another fabric has opensm (pid 31293) running...
Another fabric has opensm (pid 31293) running...
opensm start                                               [  OK  ]
smagent-rack: opensm configuration r3lead: opensmd started on fabric ib1
Running start on r4lead
Another fabric has opensm (pid 17181) running...
Another fabric has opensm (pid 17181) running...
opensm start                                               [  OK  ]
smagent-rack: opensm configuration r4lead: opensmd started on fabric ib1
```

---

**Note:** The output for the command looks a 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:

```
# smconfig -f ib0 -l 1
Configuring r1lead
```

You should now be able to start fabric `ib0` (# **`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:

```
# smadmin -f ib0 -d
      Running stop on r1lead
      opensm is running with pid of 13037...
      ......
      opensm shutdown                                          [  OK  ]
      smagent-rack: opensm configuration r1lead: opensmd stopped on fabric ib0
      Running stop on r2lead
      opensm is running with pid of 10024...
      ......
      opensm shutdown                                          [  OK  ]
      smagent-rack: opensm configuration r2lead: opensmd stopped on fabric ib0
      Running stop on r3lead
      opensm is running with pid of 651...
      ......
      opensm shutdown                                          [  OK  ]
      smagent-rack: opensm configuration r3lead: opensmd stopped on fabric ib0
      Running stop on r4lead
      opensm is running with pid of 18988...
      ......
      opensm shutdown                                          [  OK  ]
      smagent-rack: opensm configuration r4lead: opensmd stopped on fabric ib0
```

5. The fabric manager runs on all of the rack leaders, there is one MASTER node and the remainder will run in standby mode to act as a failover should the MASTER node fail (see Figure 4-1 on page 115). To find the MASTER node, perform the following:

```
# smadmin -f ib0 -m
      smagent-rack: opensm configuration r1lead: opensmd master for ib0 is r2lead
      smagent-rack: opensm configuration r2lead: opensmd master for ib0 is r2lead
      smagent-rack: opensm configuration r3lead: opensmd master for ib0 is r2lead
      smagent-rack: opensm configuration r4lead: opensmd master for ib0 is r2lead
      Running status on r1lead
      opensm is running with pid of 13037...
      Running status on r2lead
```

```
      opensm is running with pid of 10024...
      Running status on r3lead
      opensm is running with pid of 651...
      Running status on r4lead
      opensm is running with pid of 18988...
```

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

```
# smadmin -f ib0 -s
      Running status on r1lead
      opensm is running with pid of 13037...
      Running status on r2lead
      opensm is running with pid of 10024...
      Running status on r3lead
      opensm is running with pid of 651...
      Running status on r4lead
      opensm is running with pid of 18988...
```

**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):

```
# /opt/sgi/sbin/smadmin -f ib0 -d
# /opt/sgi/sbin/smadmin -f ib0 -c
# /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 106.

## 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` directory.

---

**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-ib`*x*`.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, 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="none"

# 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 SM has a failover mechanism. If the master SM fails, the standby SM takes over operation of the fabric. This failover operation is performed automatically by the `opensm` software.

In a system with four racks, three of the instances of `opensm` software are in standby mode and one instance is the master, typically rack 02 as shown in Figure 4-1 on page 115.

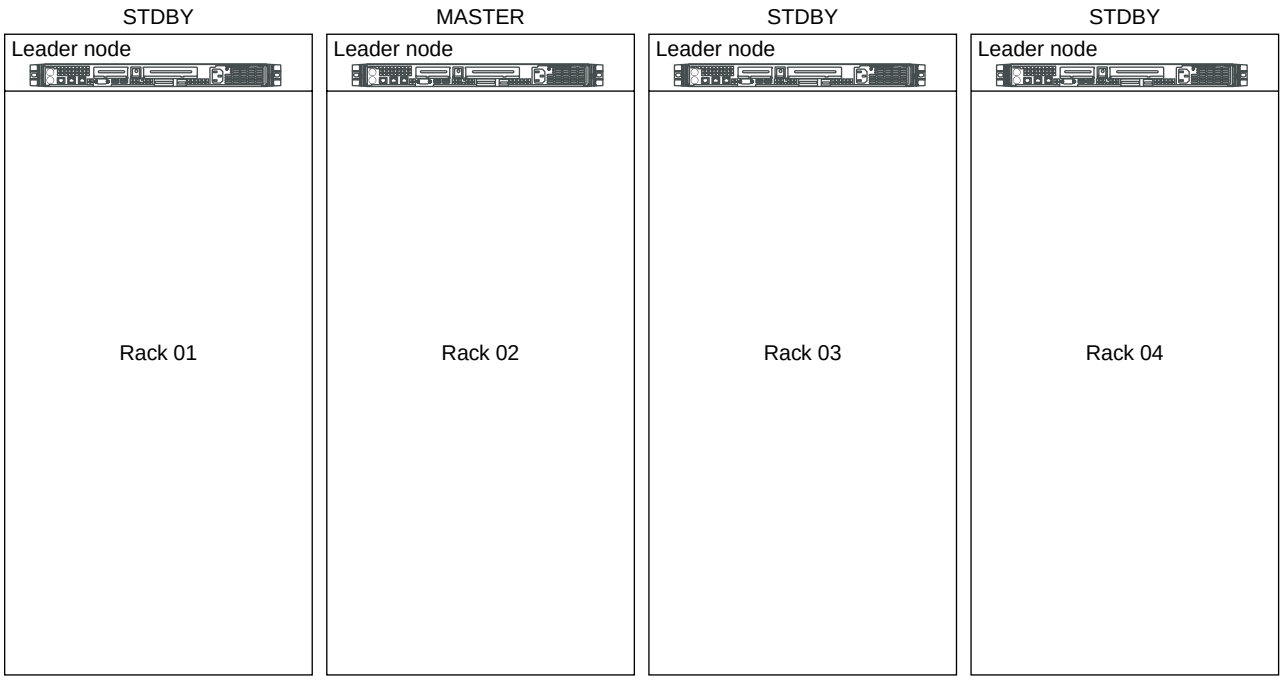


**Figure 4-1** `opensm` Software Failover

When configuring the InfiniBand fabric, you need to add the IP addresses of each rack leader controller (leader node) to the `OSM_HOSTS` variable in the `opensm-ib0.conf` and `opensm-ib.conf` configuration files. The `OSM_HOSTS` variable enables the

failover operation of the `opensm` software. For a system of two rack or less, you can leave this variable set to `''none''`. For system configurations of eight to sixteen racks, SGI recommends that every 4th rack be designated by the `OSM_HOSTS` variable.

Each fabric is addressed by a global unqiue identifier (GUID) and unique HCA port (see Figure 4-2 on page 116). Each fabric has a unique `GUID` set in its respective configuration file.

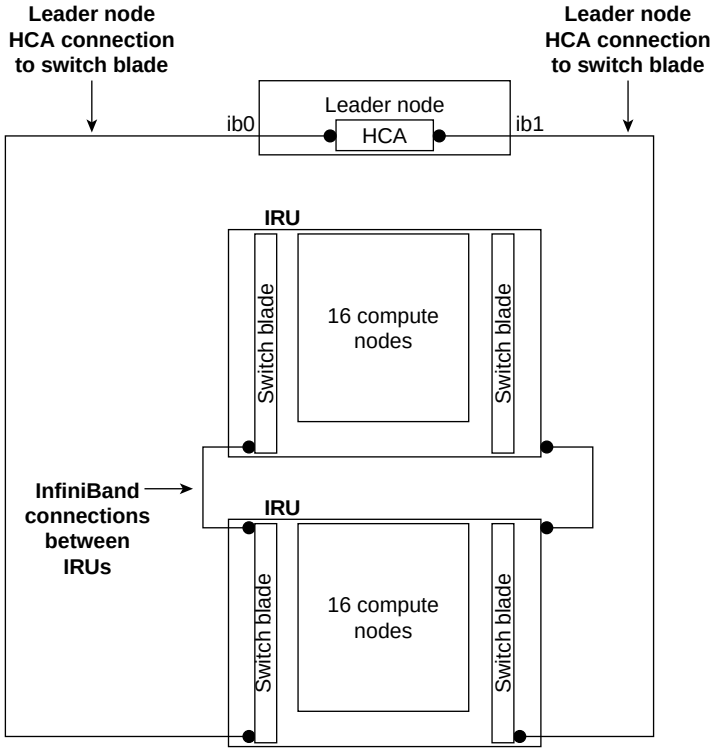


**Figure 4-2** Two InfiniBand Fabrics in a System with Two IRUs

For system with 128 compute nodes or less, you can use the `ROUTING_ENGINE="none"` default variable. For systems larger than 128 nodes, SGI recommends you set this variable to `''lash''` which enables LAyered SHortest Path Routing (LASH).

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 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 111.

**Procedure 4-4** 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. In both the `opensm-ib0.conf` file and `opensm-ib1.conf` file enable the failover (handover) mechanism on the leader nodes by adding the IP addresses recorded in step 2 to the `OSM_HOSTS` variable, as follows:

   ```
   OSM_HOSTS="172.16.0.2 172.16.0.3 172.16.0.4 172.16.0.5"
   ```

8. For systems with five or more racks, SGI recommends you change the `ROUTING_ENGINE` variable in both configuration files to `lash`, as follows:

   ```
   ROUTING_ENGINE="lash"
   ```

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

   ```
   # ./opensmd-ib0 start
   ```

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

    ```
    # ./opensmd-ib1 start
    ```

11. 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 Techn
Switch   : 0x08006900000000a4 ports 24 devid 0xb924 vendid 0x2c9 "MT47396 Infiniscale-III Mellanox Techn
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_
       --switch-map  specify a switch-map file
```

12. 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

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.
like this  :-

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 Monitoring and Debugging

This chapter describes system monitoring and covers the following topics:



## 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.

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 ISE 8000 system. It displays web browser-based, real-time (on demand) histograms of system metrics, as shown in Figure 5-1 on page 135.

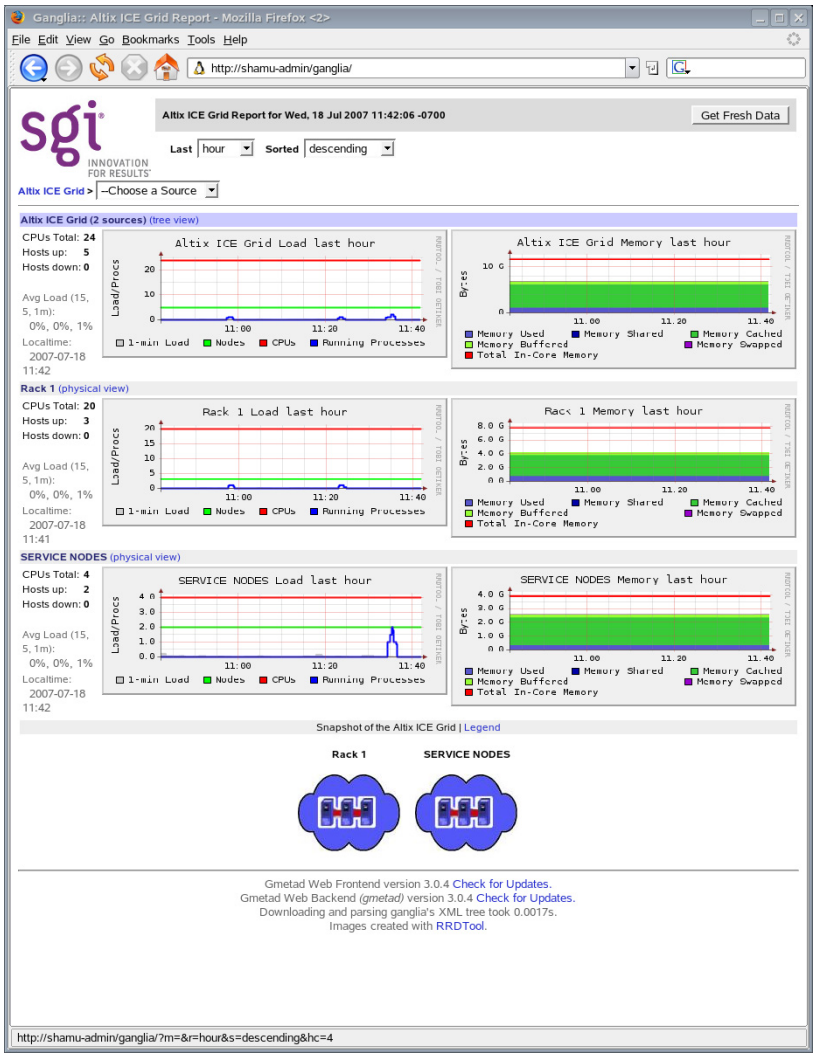


**Figure 5-1** Ganglia System Monitor

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 aggegrated 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 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.

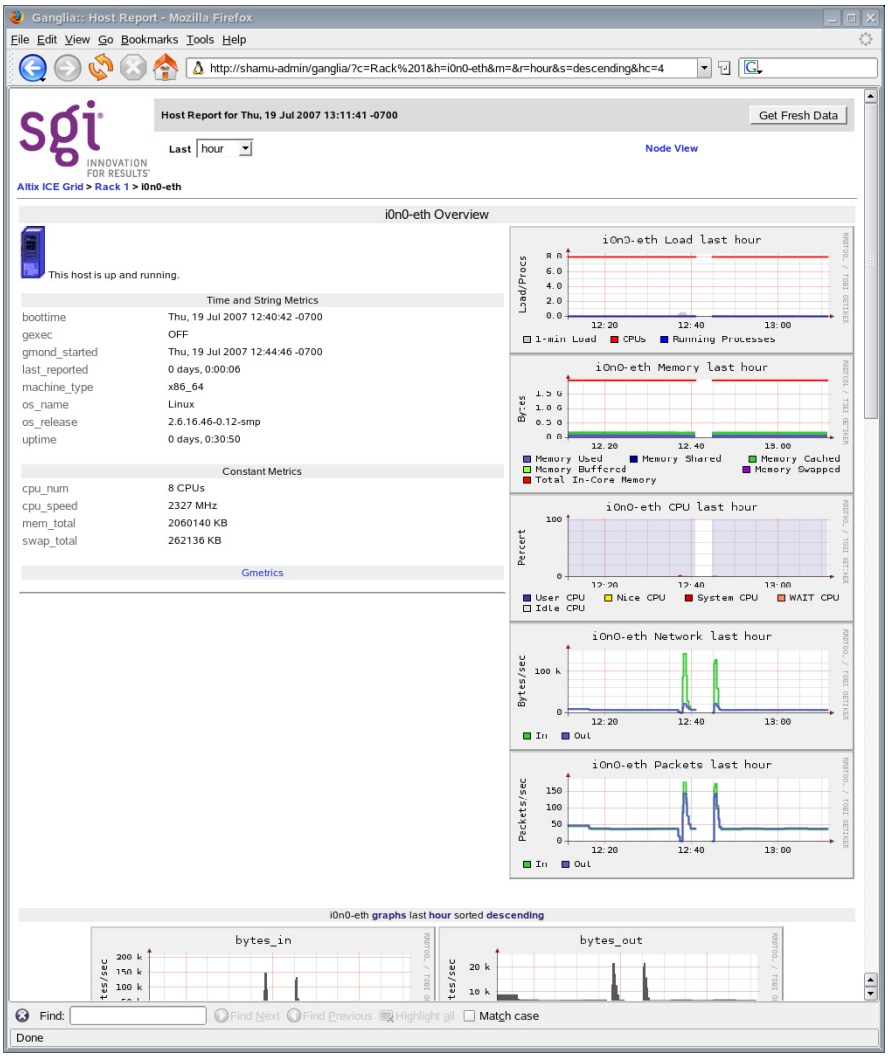


**Figure 5-2** Ganglia System Monitoring Node View

## System Monitoring Operation

To access the Ganglia system monitor, point your browser to the following location: http://*admin_pub_name*/ganglia

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.

## Troubleshooting

This section describes some troubleshooting tools and covers these topics:



### dbdump Command

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

The `dbdump` command is, as follows:

```
/opt/sgi/sbin/dbdump --admin
/opt/sgi/sbin/dbdump --leader
/opt/sgi/sbin/dbdump --rack  [--rack ]
/opt/sgi/sbin/dbdump
```

- Use the `--admin` argument to dump the system admin controller (admin node)
- Use the `--leader` argument to dump all rack leader controllers (leader nodes)
- Use the `--rack` argument to dump a specific rack
- Use the `dbdump` command without any argument to dump the entire Altix ICE system.

**EXAMPLES**

**Example 5-1** `dbdump` Command Examples

To dump the entire database, perform the following:

```
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:

```
system-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:

```
system-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:

```
system-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
```

# MVAPICH MPI

This section describes MVAPICH and covers the following topics:



## MVAPICH Overview

MVAPICH is installed as part of the OFED 1.2 software. MVAPICH is open source software developed largely by the Network-Based Computing Laboratory (NBCL) at Ohio State University. MVAPICH develops the Message Passing Interface (MPI) style of process-to-process communications for computing systems employing Infiniband and other Remote Direct Memory Access (RDMA) interconnects.

For more descriptions including the *MVAPICH User Guide* and other MVAPICH publications, see http://mvapich.cse.ohio-state.edu.

## MVAPICH Over InfiniBand

MVAPICH applications use the Infiniband network of SGI Altix ICE 8200 systems for interprocess RDMA communications. 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, including MVAPICH MPI. The `ib1` fabric is reserved for storage related traffic. The default configuration for MVAPICH MPI is to use only the `ib0` fabric.

A simple test to insure installation of MVAPICH is to enter the command:

```
% rpm -qa | grep mvapich
mvapich_gcc
```

MVAPICH executables, libraries, and man pages can be found in the following directory:

```
/usr/mpi/gcc/mvapich-0.9.9
```

Build options for MVAPICH are those assembled in the OFED1.2 distribution and include single-rail `ch_gen2` (`ibverbs`) with ROMIO.

Use of MVAPICH is a multiple step effort in that you must first build your MPI-based application using MVAPICH libraries and then execute your application with the MVAPICH-supplied launch tool, `mpirun_rshl`. A large number of tuning options and launch tool options are available. See the MVAPICH man pages, the *MVAPICH User Guide*, and the MVAPICH web site for information on these options.

## Compiling MVAPICH Applications

It is strongly recommended that MVAPICH applications be built with an MVAPICH-supplied compiler script: `mpicc`, `mpiCC`, `mpicxx`, `mpif77`, or `mpif90`. For example, to build an MVAPICH application perform the following:

```
% /usr/mpi/gcc/mvapich-0.9.9/bin/mpicc -o myapp mmpi.c
```

The compiler scripts provide links to all necessary libraries and include files. Applications can be launched using the `mpirun_rsh` command, which is available in the same MVAPICH executables directory. There are a number of options available for launching distributed applications across a cluster. For more information on the use of this tool, enter the following command:

```
% /usr/mpi/gcc/mvapich-0.9.9/bin/mpirun_rsh --help
```

A very simplified example of its use is, as follows:

```
% /usr/mpi/gcc/mvapich-0.9.9/bin/mpirun_rsh -rsh -np 4 sv0 sv1 sv2 sv3 myapp
```

where you have requested the creation of an MPI process named `myapp` on each of the four hosts `sv0`, `sv1`, `sv2`, and `sv3` using `rsh`.

# Index