Scali Manage™ On SGI® Altix® ICE System Quick Reference Guide

007–5450–001

COPYRIGHT

© 2008, SGI. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of SGI.

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. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries. Scali Manage is a trademark of Platform Computing, Inc.

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

Record of Revision

Version	Description
001	April 2008 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 or Red Hat Enterprise Linux 5.1 (RHEL5.1) with SGI ProPack 5 for Linux Service Pack 4 (or later). It describes how to use Scali Manage version 5.6.1–1 (or later) management software to perform general system discovery, installation, configuration, and operations on SGI Altix ICE 8000 series systems.

This manual contains the following chapters:

- Chapter 1, "SGI Altix ICE 8000 Series System Overview" on page 1
- Chapter 2, "Getting Started with Scali Manage" on page 27
- Chapter 3, "System Fabric Management" on page 43
- Appendix A, "InfiniBand Fabric Details" on page 45
- Appendix B, "InfiniBand Fabric Troubleshooting" on page 55

Related Publications

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

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

  This is the hardware users 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.

The following manuals for Scali Manage are available from Platform Computing, Inc:

- *Scali Manage User's Guide,* (2006)

  This document provides an overview of a Scali system in terms of instructions for building a Scali system. Configuration guidelines for hardware and software are covered along with instructions on use and general management of the cluster system.

- *Scali Quick Start Guide*, (2007)

This document is for system administrators and provides an overview of how to use Scali Manage to operate your system.

- *Scali Manage Installation Guide*, (2006)

  This document describes the Scali Manage software installer that helps the user with installation of the OS, Scali software and third-party applications that are installed as RPMs.

For a list of manuals supporting SGI ProPack for Linux releases covering the following topics, see the *SGI ProPack 5 for Linux Service Pack 4 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 4 Start Here*, the SGI ProPack 5 SP4 release notes, which contain the latest information about software and documentation in this release, the list of RPMs distributed with SGI ProPack 5 SP4, 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 SP4 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.

Chapter 1

SGI Altix ICE 8000 Series System Overview

An SGI Altix ICE 8000 series system is an integrated blade environment that can scale to thousands of nodes. The Scali Manage management software enables you to provision, install, configure, and manage your system. This chapter provides an overview of the SGI Altix ICE 8000 series system and covers the following topics:

- "Hardware Overview" on page 1
- "Networks" on page 12
- "Network Interface Naming Conventions" on page 22

Hardware Overview

This section provides a brief overview of the SGI Altix ICE 8000 series system hardware and covers the following topics:

- "Basic System Building Blocks" on page 1
- "System Nodes" on page 7

For a detailed hardware description, see the *SGI Altix ICE 8000 Series System Hardware User's Guide*.

Basic System Building Blocks

The SGI Altix ICE 8000 series system is a blade-based, scalable, high density compute system. The basic building block is the individual rack unit (IRU). The IRU provides power, cooling, system control, and the network fabric for 16 compute blades, as shown in Figure 1-1 on page 2. Each compute blade supports two either dual–core or quad-core Xeon processor sockets and eight fully-buffered, double-data-rate two (DDR2) memory dual in-line memory module (DIMMs). Four IRUs can reside in a custom designed 42U high rack.

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

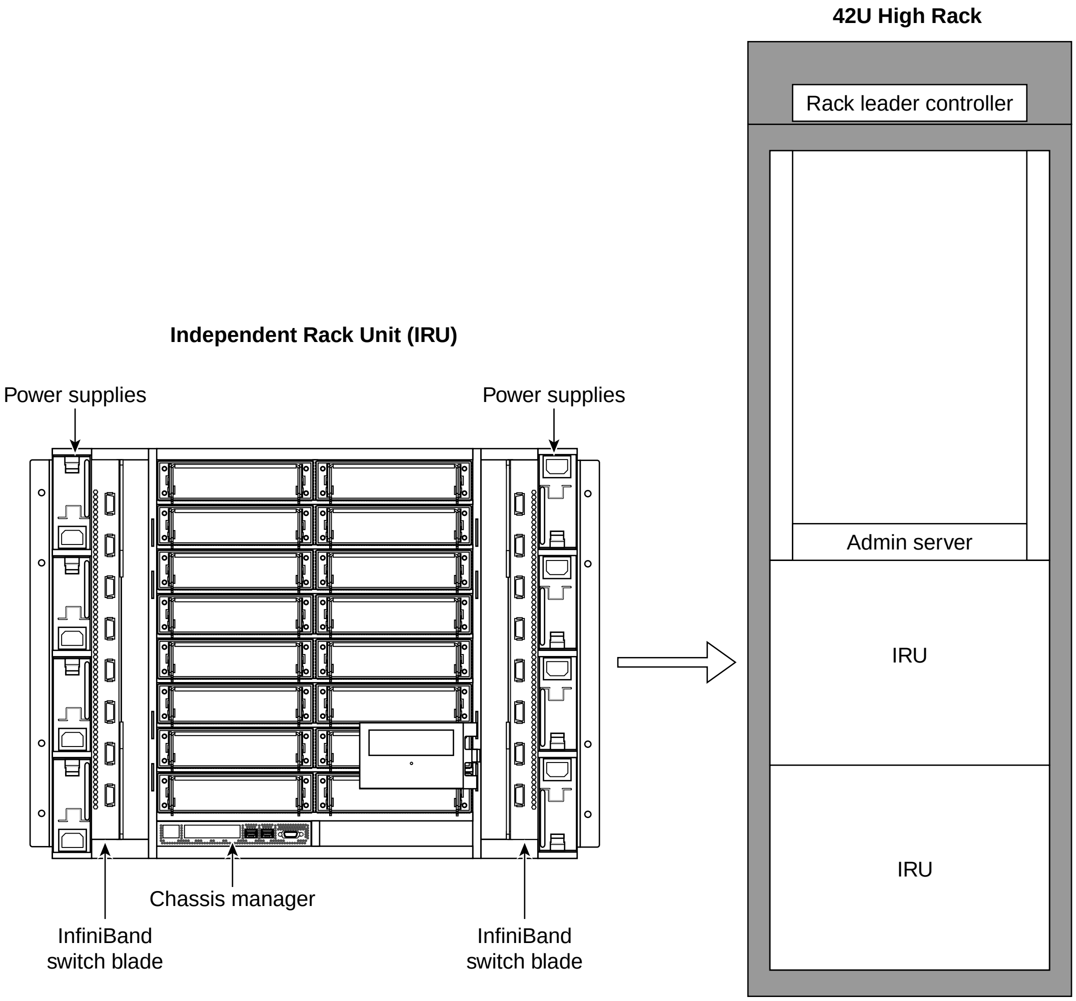


Figure 1-1 Basic System Building Blocks

This hardware overview section covers the following topics:

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

- "Power Supply" on page 4
- "Four-tier, Hierarchical Framework" on page 5
- "Chassis Manager" on page 6

InfiniBand Fabric

The SGI Altix ICE 8000 series 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 8000 series 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, such as Scali MPI or SGI Message Passing Toolkit (MPT). The `ib1` fabric is reserved for storage related traffic. The default configuration for MPI is to use only the `ib0` fabric. For more information on the InfiniBand fabric, see Chapter 3, "System Fabric Management" on page 43.

Note: The “`ib0` fabric" is a convenient shorthand for "the fabric which is connected to the `ib0` interface on most of the nodes”. In the case of the storage service node, there are four interfaces called `ib0` through `ib3`, all of which are connected to the `ib1` fabric (see "Storage Service Node " on page 11).

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

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

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

Gigabit Ethernet Network

An Gigabit Ethernet connection network built into the backplane of the IRUs provides a control network isolated from application data. Traverse cables provide connection between IRUs and between racks. For more information on how the Gigabit Ethernet connection fabric is used, see "VLANs" on page 16.

Individual Rack Unit

Each IRU has a one chassis management control (CMC) blade located directly below compute blade slot 0 as shown in Figure 1-1 on page 2. This is the chassis manager that performs environmental control and monitoring of the IRU. The CMC controls master power to the compute blades under direction of the rack leader controller (leader node). The leader node can also query the CMC for monitored environmental data (temperatures, fan speeds, and so on) for the IRU. Power control for each blade is handled by its Baseboard Management Controller (BMC), also under direction of the rack leader controller. Once the leader node has asked the CMC to enable master power, the leader node can then command each BMC to power up its associated blade. The leader node can also query each BMC to obtain some environmental and error log information about each blade.

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

Power Supply

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

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

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

Four-tier, Hierarchical Framework

The SGI Altix ICE 8000 series 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 8000 series system does **not** have a head node. The head node is replaced by a hierarchy of nodes that enables system resources to scale as you add processors. This hierarchy is, as follows:

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

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

Each system rack has its own leader node. The leader node holds the boot images for the compute blades and aggregates cluster management data for the rack.

Ethernet traffic for managing the nodes in a rack is constrained within the rack by the leader node. Communication and control is distributed across the entire cluster, thereby avoiding a communication bottleneck to the admin node. 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 16 and "Network Interface Naming Conventions" on page 22.

The rack leader controller (leader node) and admin node are described in the section that follows ("System Nodes" on page 7).

Chassis Manager

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

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 7 and Figure 1-3 on page 12 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the Altix ICE hierarchical design. They are not meant as cabling diagrams.

The chassis manager in each rack connects to the leader node in its own rack and also the chassis manager in the adjacent rack. The admin node connects to one leader node in the rack. The admin node accesses the BMC on each compute node in the rack via VLAN running over a Gigabit Ethernet (GigE) connection (see Figure 1-7 on page 19).

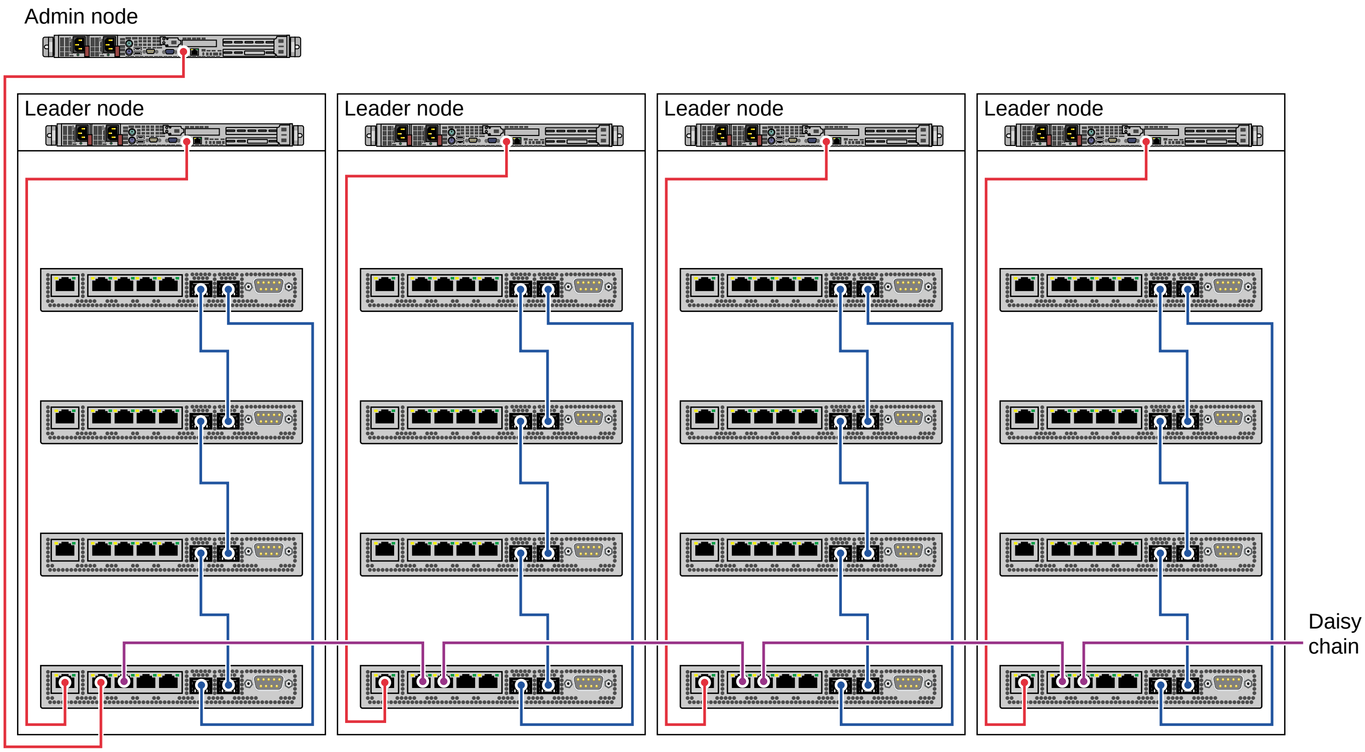


Figure 1-2 Chassis Manager Cabling

Figure 1-3 on page 12 shows cabling for a service node and storage service node (NAS cube).

System Nodes

This section describes the system nodes that are part of SGI Altix ICE 8000 series system and covers the following topics:

- "System Admin Controller" on page 8
- "Rack Leader Controller" on page 8
- "Chassis Management Control (CMC) Blade" on page 9

- "Compute Node" on page 9
- "Individual Rack Unit" on page 10
- "Login Service Node" on page 10
- "Batch Service Node" on page 10
- "Gateway Service Node " on page 11
- "Storage Service Node " on page 11

System Admin Controller

The system admin controller (admin node), is used by a system administrator to provision (install) and manage the SGI Altix ICE 8000 series system using Scali Manage systems management software. There is only one admin node per SGI Altix ICE 8000 series system, as shown in Figure 1-2 on page 7 and it cannot be combined with any other nodes. A GigE connection provides the network connection between the admin node, leader nodes, and service nodes. Communication to and from the CMC and compute blades from the admin node is controlled by VLANs to reduce network traffic bottlenecks in the system. The admin node is used to provision and manage the leader nodes, compute nodes and service nodes. It receives and holds aggregated management data from the leader nodes. The admin node is an appliance node. It always runs software specified by SGI.

The kernels, `initrds` and root filesystems (which together make up an "image") reside on the admin node.

When compute nodes are first set up with a new image, the leader nodes will cache this information to reduce the network load for the admin node.

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 admin node. There is one leader node per rack, as shown in Figure 1-2 on page 7. A GigE connection provides the network connection to other leader nodes and to first IRU within its rack as shown in Figure 1-3 on page 12 and Figure 1-4 on page 14. An InfiniBand fabric connects it to the compute nodes within its rack and compute nodes in other racks. The leader node is an appliance node. It always runs software specified by SGI. The rack leader controller (leader node) does the following:

- Runs the fabric management software to monitor and function the InfiniBand fabric on one or more leader nodes in your Altix ICE system
- Monitors, functions, and receives data from the IRUs within its rack
- Monitors, functions, and receives data from compute nodes within its rack
- Consolidates and forwards data from the IRUs and compute nodes within its rack to the admin node upon request

Chassis Management Control (CMC) Blade

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

Each IRU has a one chassis management control (CMC) blade located directly below compute blade slot 0 as shown in Figure 1-1 on page 2. This is the chassis manager that performs environmental control and monitoring of the IRU. The CMC controls master power to the compute blades under direction of the rack leader controller (leader node). The leader node can also query the CMC for monitored environmental data (temperatures, fan speeds, and so on) for the IRU. Power control for each blade is handled by the Baseboard Management Controller (BMC) also under direction of the rack leader controller. Once the leader node has asked the CMC to enable master power, the leader node can then command each BMC to power up its associated blade. The leader node can also query each BMC to obtain some environmental and error log information about each blade.

Compute Node

Figure 1-1 on page 2 shows an IRU with 16 compute nodes. Users submit MPI jobs to run in parallel on the Altix ICE system compute nodes using a public network connection via the service node. The service node provides login services and a batch scheduling service, such as the Scali MPI scheduling service. The compute nodes are controlled and monitored by the leader node in rack as shown in Figure 1-2 on page 7. A compute node is diskless and its filesystem is in memory. Scali Manage diskless means a "memory resident" operating system. This means that the operating system resides solely in the system memory. And this means that with each reboot, the compute nodes are re-imaged. For Scali Manage Altix ICE systems, there is only random random-access memory (RAM) available on compute nodes. Power cycle installs a fresh image and any changes to the compute node "filesystem" are volatile.

The image that gets loaded onto the Scali Manage Altix ICE compute nodes does get cached on the leader node. The image comes from the admin node.

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 leader nodes outside their rack.

Generally, the CMC controller is not meant to be accessed directly by system administrators, however, in some situations you may need to access it to change a configuration using the 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 8000 series system as shown in Figure 1-1 on page 2. It is described in detail in "Basic System Building Blocks" on page 1.

Login Service Node

The login service node allows users to login into the system to create, compile, and run applications. The login node is usually combined with batch and gateway service nodes for most configurations. The login service node is connected to the Altix ICE system via the InfiniBand fabric and GigE to the public customer network as shown in Figure 1-4 on page 14. Additional login service nodes can be added as the total number of user logins grow.

Batch Service Node

The batch service node provides a batch scheduling service, such as PBS Professional (not supported on Scali Manage on SGI Altix ICE systems for the first release. You need to install it separately. It is supported on the SGI Altix ICE software stack). It is commonly combined with login and gateway service nodes for most configurations. It is connected to the Altix ICE system via the InfiniBand fabric and GigE to the public customer network. This node may be separated from gateway and/or login nodes to scale for large configurations or to run multiple batch schedules.

Gateway Service Node

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

Storage Service Node

The storage service node is a network-attached storage (NAS) appliance bundle that provides InfiniBand attached storage for the Altix ICE system. There can be multiple storage service nodes for larger Altix ICE system configurations. Figure 1-3 on page 12 shows a service node and a storage service node (NAS cube).

Note: All nodes reside in the Altix ICE custom designed rack. Figure 1-2 on page 7 and Figure 1-3 on page 12 show how systems are cabled up prior to shipment. These figures are meant to give you a functional view of the Altix ICE hierarchical design. They are not meant as cabling diagrams.

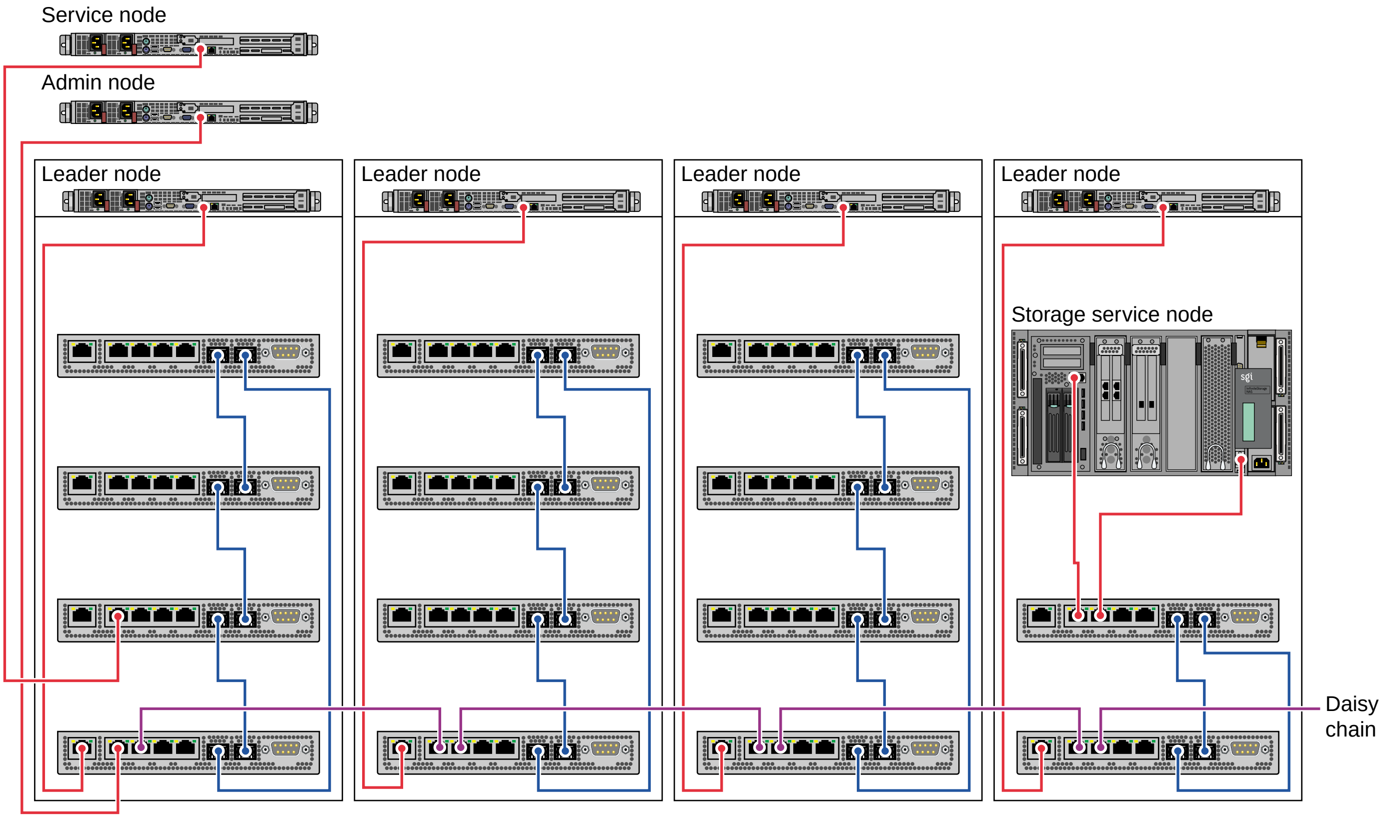


Figure 1-3 Service Nodes

Networks

This section describes the Gigabit Ethernet (GigE) and 10/100 Ethernet connections and the InfiniBand fabric in an SGI Altix ICE 8000 series system and covers the following topics:

- "Networks Overview" on page 13
- "Gigabit Ethernet (GigE) and 10/100 Ethernet Connections" on page 14
- "VLANs" on page 16

- "InfiniBand Fabric" on page 21

Networks Overview

This section describes the various network connections in the SGI Altix ICE 8000 series system. Users access the system via a public network through services nodes such as the login node and the batch service node, as shown in Figure 1-4 on page 14. A single service node can provide both login and batch services.

System administrators provision (install software) and manage the Altix ICE system via the logical VLAN network running over the GigE connection (see Figure 1-6 on page 18, Figure 1-7 on page 19, and Figure 1-8 on page 20). The admin node is on the house network (public network) and you access it directly.

The leader node is connected to blades in its rack via the GigE VLAN. It is connected to all blades and service nodes via InfiniBand fabric. Leader nodes have access to compute nodes in other racks via the leader node in that rack.

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

The admin node and service nodes communicate with the leader node over a GigE fabric that has logically separate, virtual local area networks (VLANs). This GigE fabric is embedded in the backplane of each IRU. This GigE fabric electrically connects much of the Altix ICE system (see Figure 1-4 on page 14).

Users access compute nodes strictly from the service nodes. Jobs are started on compute nodes using commands on the service node, such as, the OpenSSH client remote login program `ssh`(1), or the Scali Manage GUI invoked with the following command: `/opt/scali/bin/scalimanage-gui`.

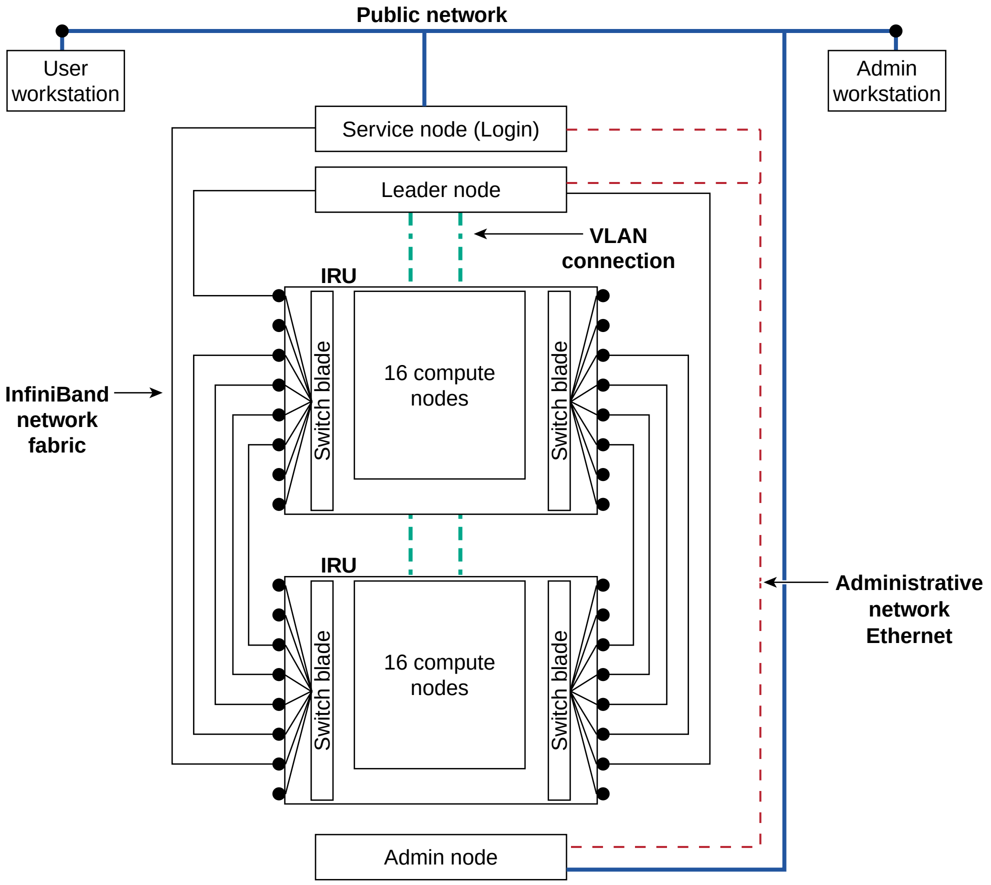


Figure 1-4 Network Connections In a System With Two IRUs

Gigabit Ethernet (GigE) and 10/100 Ethernet Connections

The SGI Altix ICE 8000 series 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 15 shows a more detailed view of the Chassis manager.

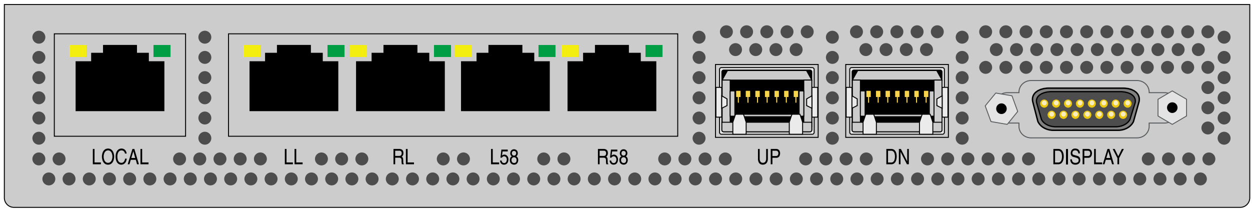


Figure 1-5 Chassis Manager

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

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

The GigE switches in each IRU are "stacked" using a special stacking connection between each IRU in a rack. This connection runs a special intra-switch protocol. All switches in a rack are ganged together to form one large 96 port switch. The connections from each CMC to another are labeled **UP** and **DN** as shown in Figure 1-5 on page 15. The switches are stacked in a ring so failure of one link still allows traffic to flow in the opposite direction on the ring.

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

The CMC has five additional RJ45 connections on its front panel as shown in Figure 1-5 on page 15. The function of these jacks is, as follows:

- **Local**

  This is a connection to the leader node at the top of the rack in which this CMC is located. Only one CMC (of the possible four) is connected to the leader node, as shown in Figure 1-2 on page 7.

- **LL**

  Used to connect service nodes and service storage nodes. The RL jack in the far left CMC connects to the LL jack of the right adjacent CMC to create or grow the Ethernet network. Figure 1-2 on page 7 shows this daisy chaining.

- **RL**

  Used to connect service nodes and service storage nodes. The RL jack in the far left CMC connects to the LL jack of the right adjacent CMC to create or grow the Ethernet network. Figure 1-2 on page 7 shows this daisy chaining.

- **L58**

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

- **R58**

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

A NAS cube storage service node uses both the **LL** and **RL** jacks to connect to the Altix ICE system as shown in Figure 1-3 on page 12.

For small, one IRU configurations, the **L58** and **R58** ports (see Figure 1-5 on page 15) can be used to connect service nodes.

VLANs

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

- `VLAN_1588`

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

- VLAN_HEAD

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

- VLAN_BMC

  Includes all 10/100 sub-switches and the leader_local ports. The VLAN_BMC VLAN connects the leader nodes to all of the BMCs on the compute blades and to the CMCs within each IRU. See Figure 1-6 on page 18.

- VLAN_GBE

  Includes all GigE blade ports and the leader_local port. See Figure 1-6 on page 18. The VLAN_GBE VLAN connects the leader nodes to the GigE interfaces of all the compute blades.

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

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

The rack leader controllers (leader nodes) must run 802.1Q VLAN protocol over their downstream GigE connection to the CMC and the CMC LL port must also run 802.1Q. This is done for you when the rack leader controllers are installed from the system admin node (see "Installing Service and Leader Nodes" on page 32). 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.

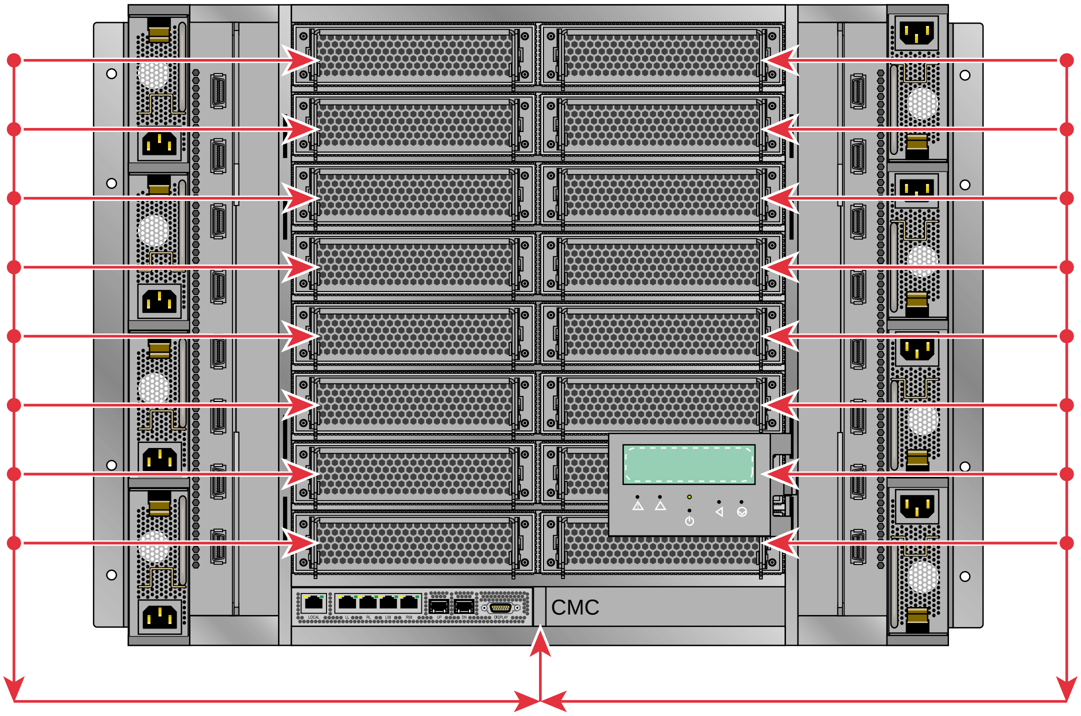


Figure 1-6 `VLAN_GBE` and `VLAN_BMC` Network Connections - IRU View

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).

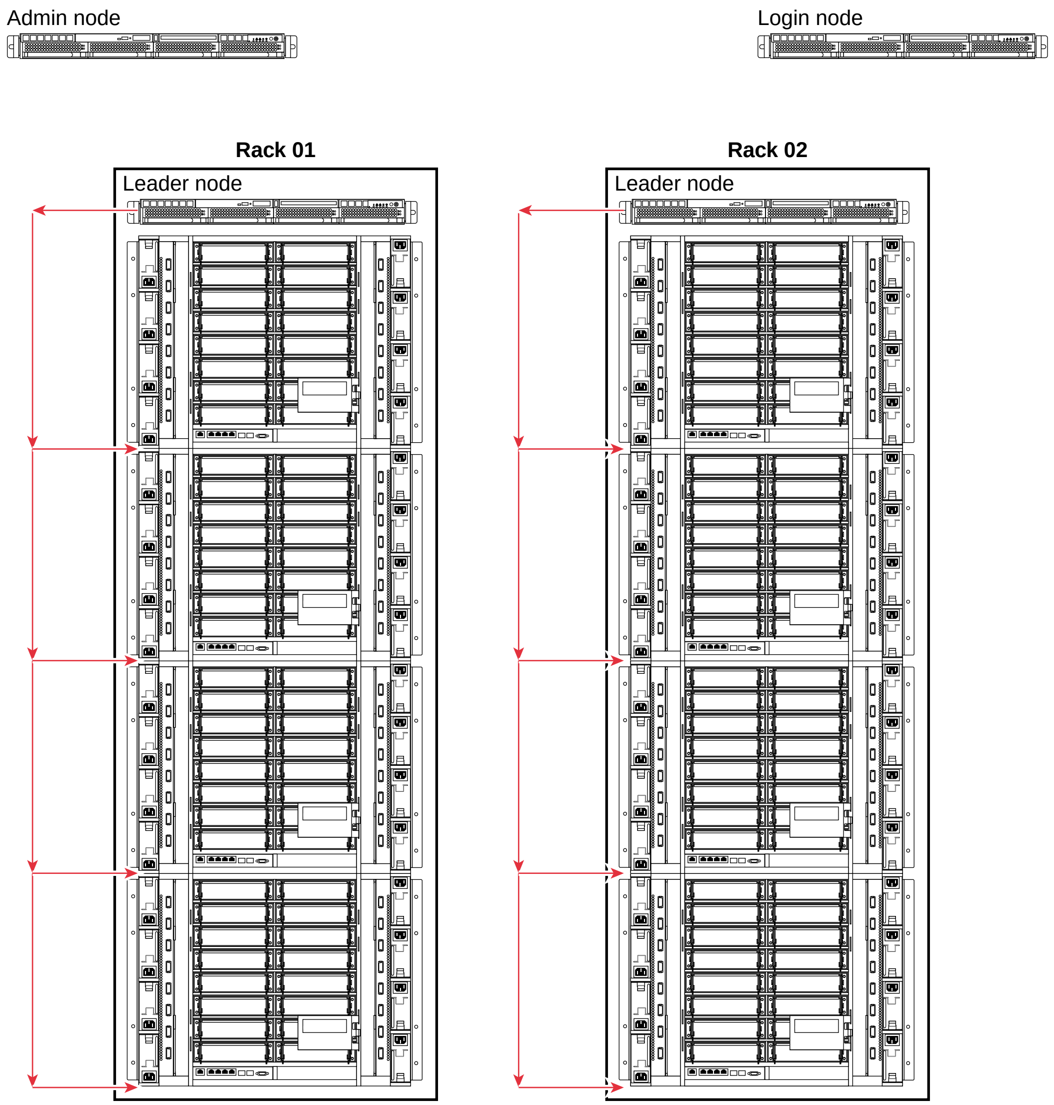


Figure 1-7 `VLAN_GBE` and `VLAN_BMC` Network Connections – Rack View

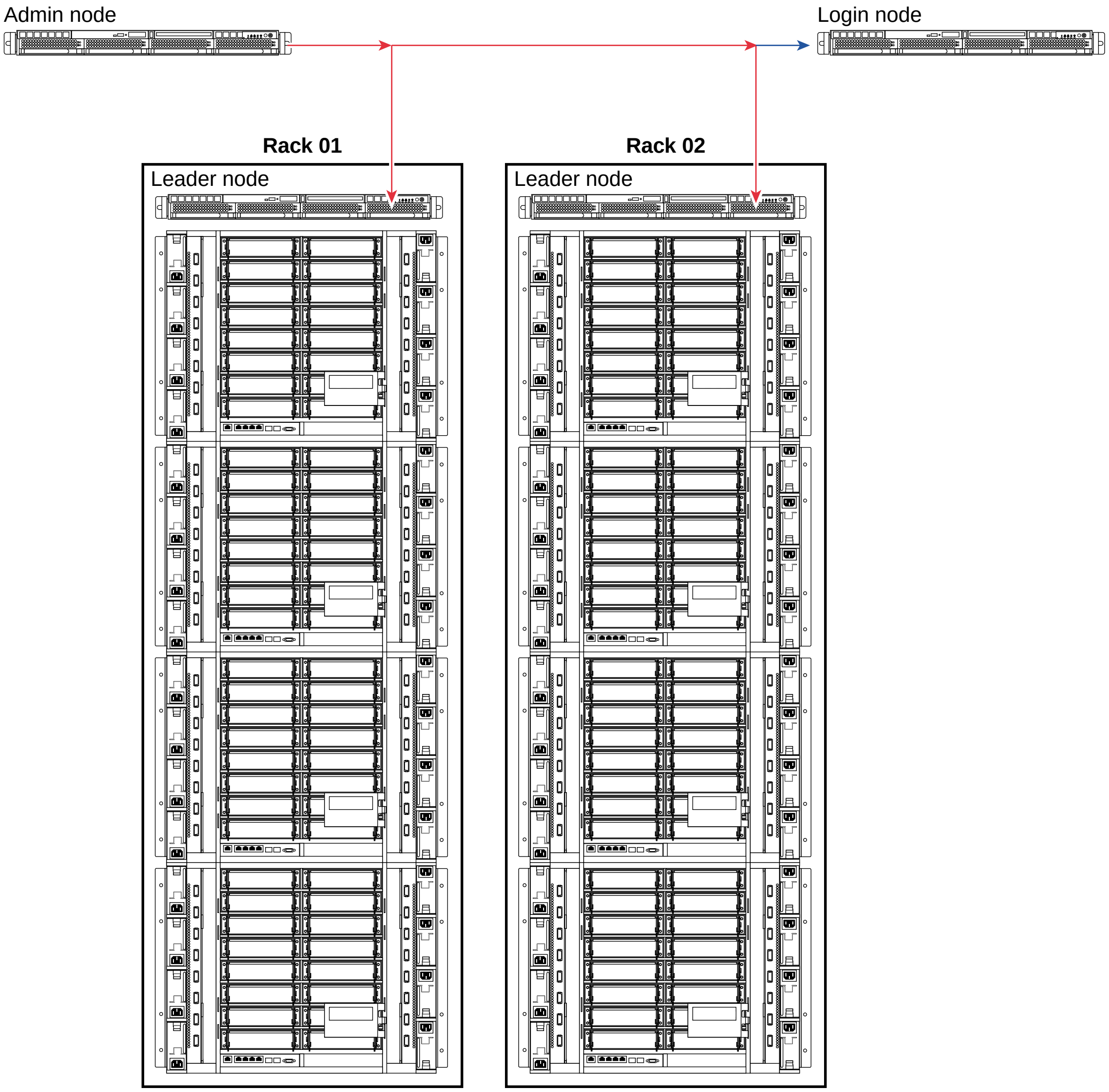


Figure 1-8 `VLAN_HEAD` Network Connections

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.

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

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

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

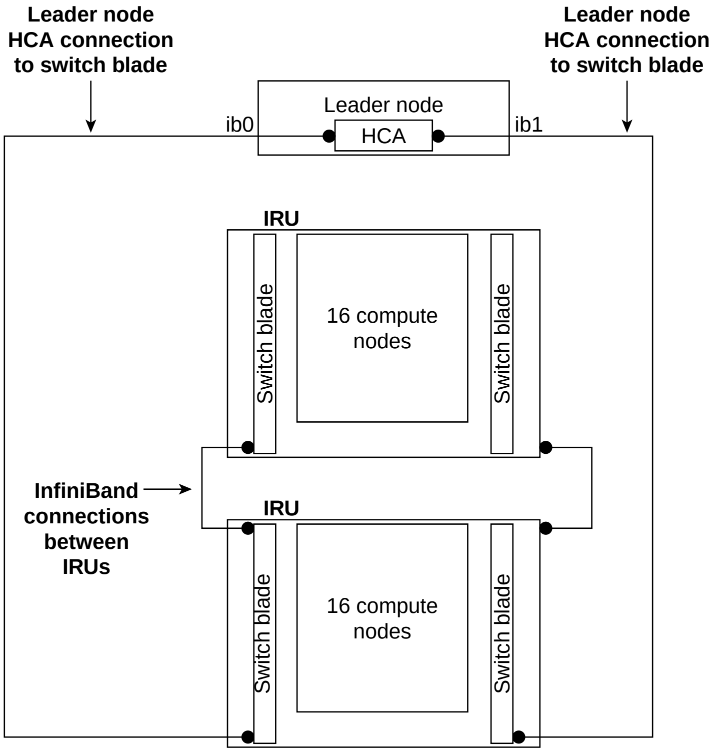


Figure 1-9 Two InfiniBand Fabrics in a System with Two IRUs

Network Interface Naming Conventions

To simplify the deployment and management of the Altix ICE system the `scaaltixice` package includes functionality to automatically configure the system according to a fixed policy tailored for the hierarchical topology used in the Altix ICE system (see "Hardware Overview" on page 1).

The network policy implemented by the `scaaltixice` package is described in this section, as follows:

- "Ethernet Networks" on page 23
- "InfiniBand Networks" on page 23
- "System Admin Controller" on page 24

- "Service Nodes" on page 24
- "Rack Leader Controllers" on page 25
- "Chassis Management Control (CMC) Blade" on page 25
- "Compute Nodes" on page 25

Ethernet Networks

The Ethernet networks implemented are, as follows:

- Corporate network, sometimes called the house network

  Your site's existing corporate network to which the Altix ICE system is connected.

- Head network

  Network for communication between admin node, service nodes, and rack leader controllers (leader nodes). The is the inter-rack communication network.

- Head BMC network

  Network for communication between admin node and the BMCs on service nodes and leader nodes. This network is on the same VLAN as the head network.

- Rack networks

  One per rack, and provides the intra-rack network for communication between the leader node and all the blades in a rack.

- Rack BMC network

  One per rack, and provides intra-rack network for communication between the leader node and the BMCs on all the blades in the rack.

InfiniBand Networks

The InfiniBand networks implemented are, as follows:

- IB subnet1

  Subnet for MPI communication. Default network.

- IB subnet2

Subnet for network filesystems.

System Admin Controller

The system admin controller (admin node) is the Scali Manage server. Networks implemented are, as follows:

- BMC

  Connected to corporate network. You set the IP address and subnet mask.

- eth0

  Connected to corporate network. You set the IP address and subnet mask.

- eth1

  Connected to the head network (IP 172.16.0.1, name `admin`) and head BMC network (IP 172.17.0.1, name `admin-mgm`).

Service Nodes

The service node networks implemented are, as follows:

- BMC

  Connected to the head BMC network (IP 172.17.0.[2-255])

- eth0

  Connected to the head network (IP 172.16.0.[2-255], name <`hostname`>

- eth1

  Optionally, connected to the corporate network. IP address and subnet mask set by the customer. Name <`hostname`>`-ext`.

- ib0

  Connected to IB subnet1. (IP 10.0.0.[2-255], name <`hostname`>`ib0`)

- ib1

  Connected to IB subnet2. (IP 10.1.0.[2-255], name <`hostname`>`ib1`)

Rack Leader Controllers

These are Scali Manage Gateways. Hostname is `rXXlead`. The rack leader controller (leader node) networks implemented are, as follows:

- BMC

  Connected to the head BMC network (IP 172.17.XX.[1-255])

- eth0:

  Connected to the head network (IP 172.16.XX.[1-255], name `rXXlead`)

  Tagged vlan 1: connected to rack BMC network (IP 192.168.1.1, name `rXXlead-mgm`)

  Tagged vlan 2: connected to rack network (IP 192.168.0.1, name `rXXlead-int`)

- ib0

  Connected to IB subnet1. (IP 10.0.XX.1, name `rXXlead-ib0`)

- ib1

  Connected to IB subnet2. (IP 10.1.XX.1, name `rXXlead-ib1`)

Chassis Management Control (CMC) Blade

The chassis management controller (ethernet switch) networks implemented are, as follows:

- Hostname is `rXXcmc[01-04]`.
- Connected to the rack BMC network (IP 192.168.1.[2-5], name `rXXcmc[01-04]`)

Compute Nodes

The compute nodes (blades) networks implemented are, as follows:

- Hostname is `r[01-xx]i[01-04]n[01-16]`.
- BMC

  Connected to the rack BMC network (IP 192.168.1.[11-74], name `r[01-xx]i[01-04]n[01-16]-bmc)`

- eth0

  Connected to the rack network (IP 192.168.0.[11-74], name `r[01-xx]i[01-04]n[01-16]-eth0`)

- ib0

  Connected to IB subnet1. (IP 10.0.XX.[11-74], name `r[01-xx]i[01-04]n[01-16]`)

- ib1

  Connected to IB subnet2. (IP 10.1.XX.[11-74], name `r[01-xx]i[01-04]n[01-16]-ib1`)

Chapter 2

Getting Started with Scali Manage

This section describes how to install, configure, discover, and operate your SGI Altix ICE system using the Scali Manage software management tool. It covers the following topics:

- "Installing or Updating Software" on page 27
- "Administrative Tips" on page 28
- "Scali Manage Command CLI Help" on page 30
- "Configuring the Scali Manage Server" on page 31
- "Defining New Racks or Service Nodes" on page 31
- "Discovering Service and Leader Nodes" on page 32
- "Installing Service and Leader Nodes" on page 32
- "Discovering CMCs and Compute Nodes" on page 33
- "Installing Compute Nodes" on page 33
- "Configuration Session Example" on page 37
- "Scali Manage Troubleshooting Tips" on page 39
- "Compute Node RPMs" on page 40

Note: SGI Altix ICE systems running Scali Manage software are shipped pre-installed. Instructions in this section for defining and discovering nodes can be used if you are expanding the initially delivered cluster or reinstalling your software. They are NOT for configuring the initially delivered cluster.

Installing or Updating Software

Scali Manage offers a mechanism to upload and install software across the SGI Altix ICE system. This upload and installation process requires that the software installation be in RPM format. Tarball software distributions can be installed across a cluster.

Instructions for installing software options or uploading additional software for your system using the Scali GUI are covered in Chapter 3 of the *Scali Manage User's Guide*.

Customers with support contracts needing BIOS or Firmware updates, should check the SGI Supportfolio Web Page at: https://support.sgi.com/login

Administrative Tips

This section describes some useful administrative tips and covers these topics:

- "System Password Information" on page 28
- "Power on or Power off System Components or Obtain Status" on page 28
- "Scali Manage Installer Directory " on page 29

System Password Information

Root password and administrative information includes:

- Root password = **sgisgi** (system admin controller (admin node) and compute nodes)
- ipmitool user/password information: User = **ADMIN** Password = **ADMIN**

Power on or Power off System Components or Obtain Status

To power on or power off system componets, use the Scali Manage power command. To get a system console, use the Scali Manage console command. See the "The Power Interface" and "The Console Interface" sections in Chapter 9, "Scali Manage Command Line Interfaces" of the *Scali Manage User's Guide*.

From the admin node, to power on compute nodes r01i01n01 and r01i01n02, perform the following:

```
system-admin: # scash -p -n r01lead /opt/scali/sbin/power r01i01n0[1,2] on
```

To check status for compute nodes r01i01n01 and r01i01n02, perform the following:

```
system-admin: # scash -p -n r01lead /opt/scali/sbin/power r01i01n0[1,2] status
r01lead  : r01i01n01: ON
```

```
r01lead  : r01i01n02: ON
```

To get a console for a service node, perform the following:

```
system-admin: # console service1
[Enter '^Ec?' for help]


Welcome to SUSE Linux Enterprise Server 10 SP1 (x86_64) - Kernel 2.6.16.46-0.12-smp (console).


service1 login:
```

Note: For this release, you need to log onto the appropriate leader node to run the power commands.

You can also use the Scali Manage GUI to execute power commands. The Scali Manage GUI supports a clean shutdown. Clean shutdown is required for the service and leader nodes because they have local disk. A power cycle with the Altix ICE compute nodes causes the complete re-imaging of the compute node.

For information on Scali Manage networking conventions used with the power commands, see "Network Interface Naming Conventions" on page 22.

Scali Manage Installer Directory

The Scali Manage installer directory (/usr/local/Scali###) is the location of the code used to install Scali Cluster management Software.

The Factory-Install directory is located on the admin node server at /usr/local/Factory-Install. The /Factory-Install directory contains software files that support the cluster integration and many files and scripts under /usr/local/ that may be helpful, including:

/Factory-Install/Apps	Scali, ibhost, Intel compilers, MPI runtime libraries, ipmitool, and so on
/Factory-Install/ISO	CD ISO images of the base OS for installing Scali Cluster Manage software
/Factory-Install/Docs	Cluster documentation manuals (Scali, PBS Professional, Voltaire, SMC, SGI)

/Factory-Install/Firmware	Voltaire HCA and Voltaire switch firmware files, etc
/Factory-Install/CFG	Cluster configuration files
/Factory-Install/Scripts	Miscellaneous utility scripts

Scali Manage Command CLI Help

You can get a help statement for the Scali Manage command line interface (CLI) as shown in the following example:

```
system-1:~ # scalimanage-cli help SGI
----  SGI Altix ICE commands ----
List of commands:
definealtixiceblade - Define Altix ICE Blades(s)
definealtixicecmc - Define Altix ICE CMCs(s)
definealtixiceleadnode - Define Altix ICE Lead node(s)
definealtixicerack - Define Altix ICE Rack(s)
definealtixiceservicenode - Define Altix ICE Service node(s)
discoveraltixicecmc - Discover CMC and Blade MAC addresses
discoveraltixiceservicenode - Find BMC MAC addresses for systems
initaltixicesms - Initate Scali Manage Server for Altix ICE
poweraltixiceiru - Control power to an Altix ICE IRU
restartaltixiceopensm - Restart the OpenSM subnetmanagers on the leadnodes
Type "help" followed by
command name for full documentation.
Command name abbreviations are allowed.
```

To get partial help information, perform the following:

```
[system-1 ~]# cli help all | grep -i remotefs addremotefs <systemnames> <fstype> <src>
<mntpoint> [options] listremotefs <systemnames> removeremotefs <systemnames> <mntpoint>
```

You can also get help on specific commands, as follows:

```
system-1:~ # scalimanage-cli help definealtixiceblade

definealtixiceblade  [irus=[1-4]] [slots=[1-16]]
    Define Altix ICE Blades(s)
```

```
racks       - Rack number for the blade(s) [..]
irus        - IRU number for the blade(s) [..]
slots       - Blade slots for the blade(s) [..]

Options:
    --irus=IRUS
    --slots=SLOTS
```

Configuring the Scali Manage Server

After installing the operating system and Scali Manage on the system admin node (Scali Manage server) to automatically configure the Scali Manage Server according to the Altix ICE network topology, perform the following:

`scalimanage-cli` **`initaltixicesms`** ***ProPack_path***
This will perform the following actions on your system:

- Define all the network subnets according to the policy (see "Network Interface Naming Conventions" on page 22)
- Add the `eth1` and `eth1:headbmc` interfaces with preset IP addresses
- Load the SGI ProPack software stack

Defining New Racks or Service Nodes

To add one or more racks of compute nodes, perform the following:

`scalimanage-cli` **`definealtixicerack <`*`racknumbers`*`>`**

To add multiple racks in one action, use a bracket expression for the racknumber, such as the following:

`scalimanage-cli` **`definealtixicerack [1-16]`**
Running the command above, will update the Scali Manage configuration database and define the following

- One rack leader controller (leader node), sixteen leader nodes for the example above

- A rack subnet and a rack BMC subnet per rack
- Four chassis management controllers (CMCs) per rack
- 16 compute blades per CMC per rack

Optionally, the number of CMCs per rack and the number of blades per CMC can be specified to define partial Altix ICE rack configurations. The command is a shortcut for `definealtixiceleadnode`, `definealtixicecmc` and `definealtixiceblade`. For more fine grained control, for example, adding only a partly full rack or add more compute nodes to an existing rack, use the `definealtixiceleadnode` command.

To define a service node, use the `definealtixiceservicenode` command.

After defining new hardware in the database, the service node(s) and leader nodes BMCs must be discovered and configured (see "Discovering Service and Leader Nodes" on page 32).

Discovering Service and Leader Nodes

Before new service or leader nodes can be installed, the associated BMCs MAC addresses must be discovered and IP addresses must be assigned. To do this, perform the following:

```
scalimanage-cli discoveraltixiceservicenode [systemnames]
```

This will perform the following actions on your system:

- Ask the operator to plug in one rack/service node at the time
- Discover the MAC addresses of the associated BMCs
- Assign IP addresses to the BMCs via dynamic host configuration protocol (DHCP)

Installing Service and Leader Nodes

To install service nodes or leader nodes, perform the following:

```
scalimanage-cli install <systemnames>
```

If the MAC address of the system is unknown, it will be automatically determined via DHCP discovery.

Discovering CMCs and Compute Nodes

Before compute nodes can be booted the CMCs must have IP addresses assigned and the MAC addresses of the BMCs and blades must be discovered and IP addresses must be assigned. To do this, perform the following:

```
scalimanage-cli discoveraltixicecmc [cmcnames]
```

This will perform the following actions on your system:

- Discover the MAC addresses of the CMCs
- Assign IP addresses to the CMCs
- Power on the IRUs
- Discover MAC addresses of BMCs through CMCs
- Assign IP addresses to the BMCs
- Power on nodes
- Discover MAC addresses of blades through CMCs

Installing Compute Nodes

As described in the chapter 1, “SGI Altix ICE 8200 System Overview”, on SGI Altix ICE systems, the InfiniBand network `ib1` is to be used for storage traffic, the InfiniBand network `ib0` is to be used for MPI traffic, and the Ethernet network is used only for system administration.

The node from which the compute node installation image is created must **not** have a `service-ib1:/home` NFS mount entry in the `/etc/fstab` file. It is very likely that you will need to manually delete this entry before creating an installation image of the node.

Use the `scalimanage-cli addremotefs` command to create the mounts. Scali Manage runs the NFS mount command later in the boot sequence than mounting of NFS filesystems listed in `/etc/fstab`.

The mounts created with `scalimanage-cli addremotefs` are persistent across reboots and reinstalls. If new nodes are added to the system, it is necessary to run `scalimanage-cli addremotefs` for the new nodes.

To install a compute node, perform the following steps:

1. The installation of the compute node can either be a direct installation using packages, or can be an installation using an installation image created from another node, such as a service node. If the former method is used, it is possible to use a compute node specific installation template.

2. Confirm that there is no `service-ib1:/home` NFS mount configured.

   Confirm that there is no entry for `service-ib1:/home` in the `/etc/fstab` file of the system from which the image is going to be created.

   If this is the initial installation, then no `/home` entry is expected in the `/etc/fstab` of the compute node that is about to be imaged. However, if you are not doing an initial installation, then this entry will exist in `/ec/fstab`. Delete the `service-ib1:/home` entry from the node `/etc/fstab` file. There may be other NFS entries included in `/etc/fstab` file that also need deleting, such as, a `service-ib1:/data` entry, or an entry or entries for off-cluster NFS servers.

3. Create an installation image from this node.

   From the Altix ICE admin node, run the `scalimanage-cli captureimage` command. You can get a usage statement for this command, as follows:

```
scalimanage-cli help captureimage
captureimage <systemname> <imagename> [description] [excludes..]
     Capture image from system
     Arguments
         systemname  - name of system
         imagename   - name of image
         description - Description of image; Default none
         excludes    - list of files or directories to be excluded (space separated)

     Options:
         --description=DESCRIPTION
```

   To capture an image from node `r01i01n02` and save that image as `r01i01n02-image1`, perform the following:

   ```
   scalimanage-cli captureimage r01i01n02 r01i01n02-image1
   ```

4. Configure the compute nodes to use this image.

```
scalimanage-cli help setdiskless
setdiskless <systemnames> <imagename>
     This method sets systems(s) diskless with software image
```

```
     Arguments:
         systemnames - system(s) {[..]}
         imagename   - os image to set
```

To set nodes `r01i01n02...r01i01n06` to use the image created from `r01i01n02`, perform the following:

```
scalimanage-cli setdiskless r01i01n0[2-6] r01i01n02-image1
```

Run `scalimanage-cli reconfigure all` to propagate the Scali Manage changes.

5. Configure Scali Manage to manage the `service-ib1:/home` NFS mount.

   With the image saved, use the `scalimanage-cli addremotefs` command to add the NFS mount to nodes. You can get a usage statement for this command, as follows:

```
scalimanage-cli help addremotefs
addremotefs <systemnames> <fstype> <src> <mntpoint> [options=_netdev]
     Add mounting for remote filesystem on system(s)
     Arguments:
         systemnames - name of system(s) {[..]}
         fstype      - type of filesystem, legal values: "nfs" "lustre"
         src         - source
         mntpoint    - mountpoint
         options     - options to mount command to be given as -o options to mount.
                       By default options="_netdev".
                       Options values should be comma seperated
                       for e.g "_netdev,tcp,hard,rsize=64K,wsize=64K,intr"

     Options:
         --options=OPTIONS
```

To configure Scali Manage with the NFS mount, perform the following:

```
scalimanage-cli addremotefs r01i01n0[2-6] nfs service1-ib1:/home /home
```

Propagate the changes with `scalimanage-cli reconfigure all` command. Errors may occur with the nodes that have not been installed.

You can confirm that Scali Manage knows about the mounts, as follows:

```
scalimanage-cli listremotefs r01i01n0[2-6]
```

6. Power off or confirm that the compute nodes are powered off.

   Currently, the Scali Manage GUI power node off or on does not work correctly from the admin node.

   You can log on to the rack leader controller (leader node) and power off the compute nodes from there using one of the methods described below.

   From the leader node, highlight all nodes in the GUI and select right click> **Node On/Off** > **Power Off**

7. From the rack leader node use power command, as follows:

   ```
   r01lead:~ # power r01i01n0[2-6] off
   r01i01n02: SUCCESS
   r01i01n03: SUCCESS
   r01i01n04: SUCCESS
   r01i01n05: SUCCESS
   r01i01n06: SUCCESS

   r01lead# power r01i01n0[2-6] status
   r01i01n02: OFF
   r01i01n03: OFF
   r01i01n04: OFF
   r01i01n05: OFF
   r01i01n06: OFF
   ```

8. From the rack leader node you can use the `ipmitool`, as follows:

```
r01lead# /usr/bin/ipmitool -I lanplus -o supermicro -U ADMIN -P ADMIN -H 192.168.1.14 power off
Chassis Power Control: Down/Off

r01lead# /usr/bin/ipmitool -I lanplus -o supermicro -U ADMIN -P ADMIN -H 192.168.1.14 power status
Chassis Power is off
```

9. Install the compute nodes.

Starting with the nodes powered off, install the compute nodes from the SGI Altix ICE admin node, as follows:

```
scalimanage-cli install r01i01n0[2-6]
```

DHCP requests can be followed on the rack leader nodes, as follows:

```
r01lead# tail -f /var/log/messages
```

You can follow the installation and boot of the nodes (or of representative nodes) using either the Scali Manage consoles (from the GUI) or using `ipmitool` SOL console interface.

10. From the admin node, verify that `/home` is mounted as expected, as follows:

```
# scashdc -p mount | grep home | sort
r01i01n02-eth0 : service1-ib1:/home on /home type nfs (rw,_netdev,addr=10.1.0.1)
r01i01n03-eth0 : service1-ib1:/home on /home type nfs (rw,_netdev,addr=10.1.0.1)
r01i01n04-eth0 : service1-ib1:/home on /home type nfs (rw,_netdev,addr=10.1.0.1)
r01i01n05-eth0 : service1-ib1:/home on /home type nfs (rw,_netdev,addr=10.1.0.1)
r01i01n06-eth0 : service1-ib1:/home on /home type nfs (rw,_netdev,addr=10.1.0.1)
```

Configuration Session Example

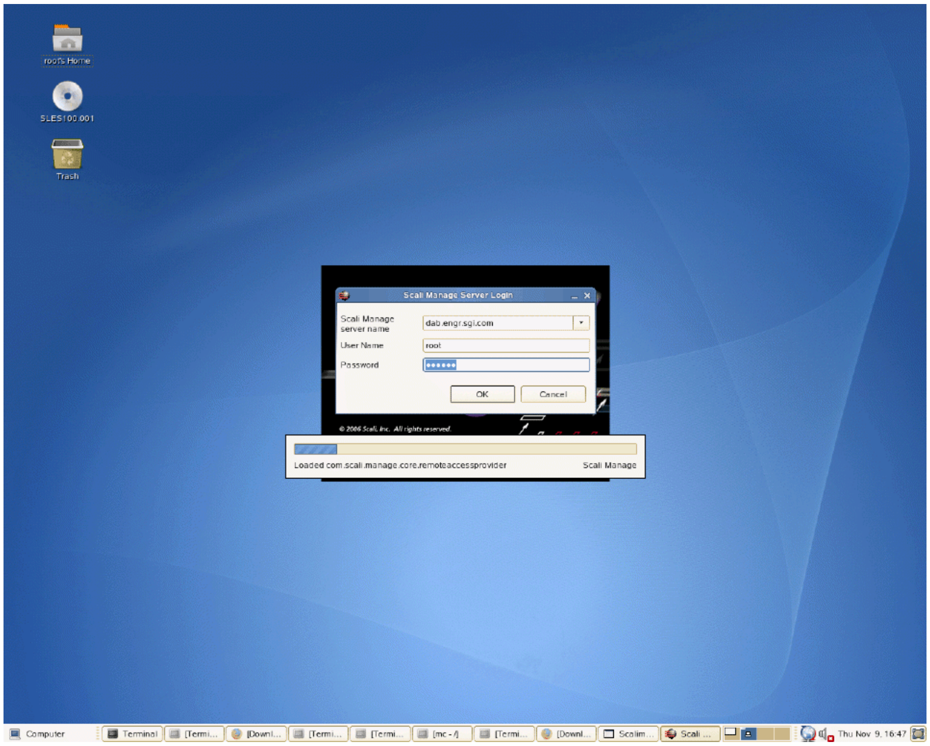
This is section shows a complete SGI Altix ICE configuration example, as follows:

```
scalimanage-cli initaltixicesms /tmp/ofed.stout5sp2.rpms.tgz
scalimanage-cli definealtixicerack 1 [1-2] [1-2]
scalimanage-cli definealtixiceservicenode service1
/etc/init.d/scance restart
scalimanage-cli discoveraltixiceservicenode
scalimanage-cli install "service1 r01lead"
scalimanage-cli discoveraltixicecmc
scalimanage-cli captureimage service1 image1
scalimanage-cli setdiskless r01i[01-02]n[01-02] image1
scalimanage-cli install r01i[01-02]n[01-02]
```

Using the Scali Manage GUI

This chapter provides general administrative information section and information on starting and using the Scali Manage GUI in a Scali managed cluster. For information on using the Scali Manage command line interface, refer to the *Scali Manage User's Guide*.

Login to the Scali Manage interface as root, the factory password is **`sgisgi`**. Use your system name and log in as root as shown in Figure 2-1 on page 38.



Scali Manage server name: dab.engr.sgi.com
User Name: root
Password: ••••••

Loaded com.scali.manage.core.remoteaccessprovider   Scali Manage

Figure 2-1 Example Starting Screen for the Scali Manage GUI

Displaying Cluster Components

Cluster components are shown in Figure 2-2 on page 39. **r01** is rack 01 and **r02** is rack 02, **i01** is IRU 1, and **n01** and **n02** are nodes 1 and 2. **r01lead** and **r02lead** are the rack leader controllers (leader nodes) for the cluster. **service1** is the service nodes for the cluster. System naming conventions when using Scali Manage are described in "Network Interface Naming Conventions" on page 22.

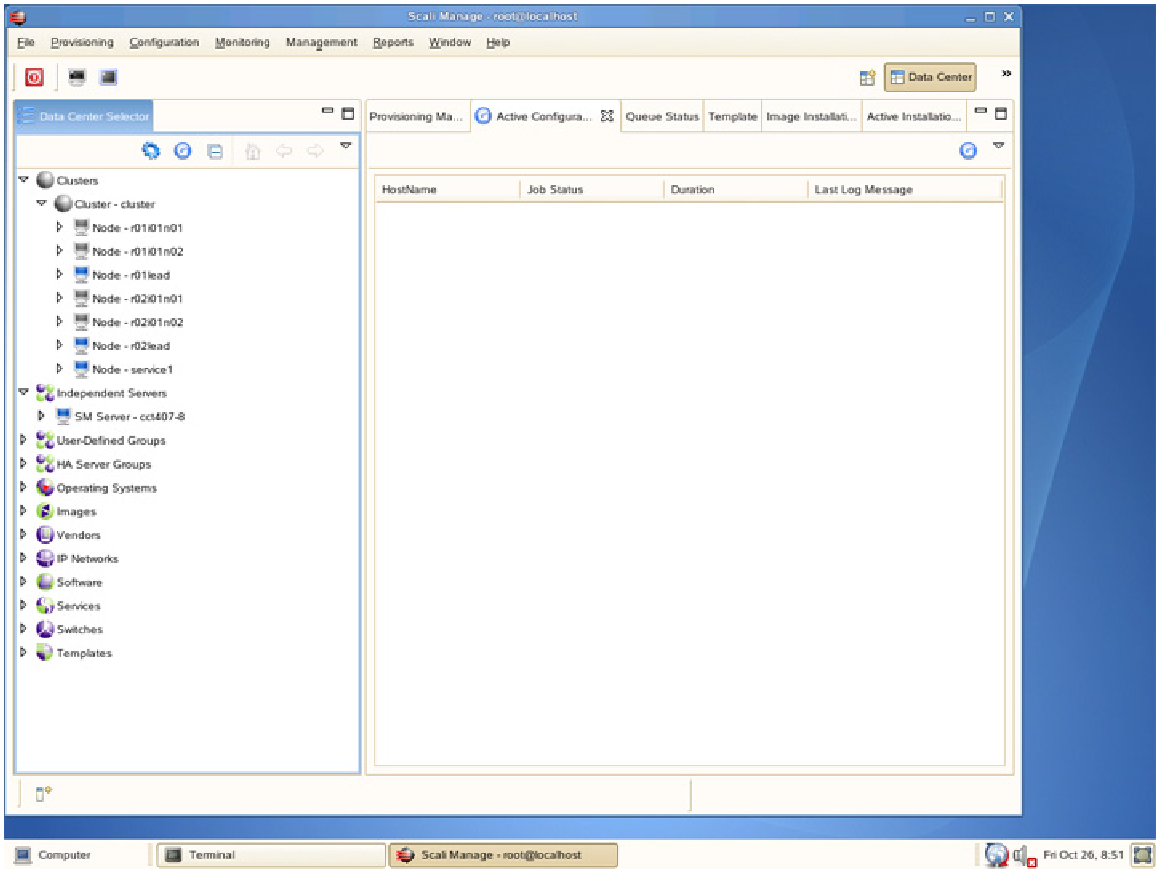


Figure 2-2 Cluster Components Selection Screen Example

Scali Manage Troubleshooting Tips

This section describes some general guidelines as well as emergency procedures.

Whenever a Scali cluster parameter is changed, it is necessary to apply the configuration. This can be done either through the graphical user interface (GUI) by selecting **Provisioning > Apply All Configuration Changes** or via the command line interface (CLI), as follows: `scalimanage-cli reconfigure all`. Changes can be made in batches and then applied all at once.

There are situations when the GUI does not reflect the cluster configuration properly. Restarting the GUI may solve this problem.

In rare cases the Scali product enters an inconsistent state. In this state it shows abnormal behavior and refuses to take any input. In this case try to reinitialize the admin node via `/etc/init.d/scance restart`.

This command must be run on the admin node. If this does not change Scali's state, then you should reboot the admin node. This should ensure that Scali will be in a consistent state.

Array services has configuration files `/etc/array/arrayd.{auth,conf}` with links to **`/usr/lib/arrayd.{conf,auth}`**. When you update your system configuration and later reboot the compute node(s), your configuration will be lost because the compute nodes are stateless. You need to capture another image after changing configuration files.

Compute Node RPMs

The following section describes what packages are installed on the compute node and covers these topics:

- "Compute Node RPMs on SLES" on page 40
- "Compute Node RPMs on RHEL" on page 41

Compute Node RPMs on SLES

The following RPMs reside on the compute node when you run Scali Manage on top of SUSE Linux Enterprise Server 10 (SLES10):

```
cpuset-utils
dapl
dapl-devel
dapl-utils
ibutils
intel-cluster-runtime
ipoibtools
kernel-ib-ice
libbitmask
libcpuset
```

```
libibcm
libibcommon
libibmad
libibumad
libibverbs
libibverbs-devel
libibverbs-utils
libmthca
libopensm
libosmcomp
libosmvendor
librdmacm
librdmacm-utils
lkSGI
mpitests_mpt
msr-tool
mstflint
numatools
ofed-docs
ofed-scripts
openib-diags
```

Compute Node RPMs on RHEL

The following RPMs reside on the compute node when you run Scali Manage on top of Red Hat Enterprise Linux 5 (RHEL5):

```
cpuset-utils
dapl
dapl-devel
dapl-utils
environment-modules
ibutils
intel-cluster-runtime
ipoibtools
kernel-ib-ice
kmod-numatools
kmod-ofa_kernel
kmod-xpmem
libbitmask
libcpuset
```

```
libibcm
libibcommon
libibmad
libibumad
libibverbs
libibverbs-devel
libibverbs-utils
libmthca
libopensm
libosmcomp
libosmvendor
librdmacm
librdmacm-utils
lkSGI
mpitests_mpt
msr-tool
mstflint
numatools
ofed-docs
ofed-scripts
openib-diags
pcp-open
perftest
rds-tools
sgi-arraysvcs
sgi-mpt
sgi-procset
sgi-release
sgi-support-tools
tvflash
xpmem
```

Chapter 3

System Fabric Management

The InfiniBand network on SGI Altix ICE 8000 series 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.

Fabric management on SGI Altix ICE 8000 series 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 21) with the following characteristics:

- Each network fabric has its own subnet manager (SM).
- For a system with two racks or more, one rack leader controller (leader node) runs an instance of SM to manage the `ib0` fabric and a second leader node runs an instance of SM to manage the `ib1` fabric.
- On a system with a single rack, both instances of `opensm` run on the same rack leader node.
- Each instance of SM on the rack leader controller is controlled by the `/etc/opensm-ib0.conf` or `/etc/opensm-ib1.conf` configuration file.
- Rack leader controllers run the `opensm` daemon for each fabric over separate HCA ports (see Figure 1-9 on page 22).

  Note: 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. To start the InfiniBand fabric, you can use the following command:

  ```
  scalimanage-cli restartaltixiceopensm
  ```

- 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 MPI or interprocess communication traffic and the InfiniBand fabric `ib1` is reserved for storage.

For more information on the InfiniBand fabric, see Appendix A, "InfiniBand Fabric Details" on page 45 and Appendix B, "InfiniBand Fabric Troubleshooting" on page 55.

Appendix A

InfiniBand Fabric Details

This appendix provides more a more detailed description of the InfiniBand fabric management.

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 A-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 fabric is addressed by a global unqiue identifier (GUID) and unique HCA port (see Figure A-1 on page 50). Each fabric has a unique `GUID` set in its respective configuration file.

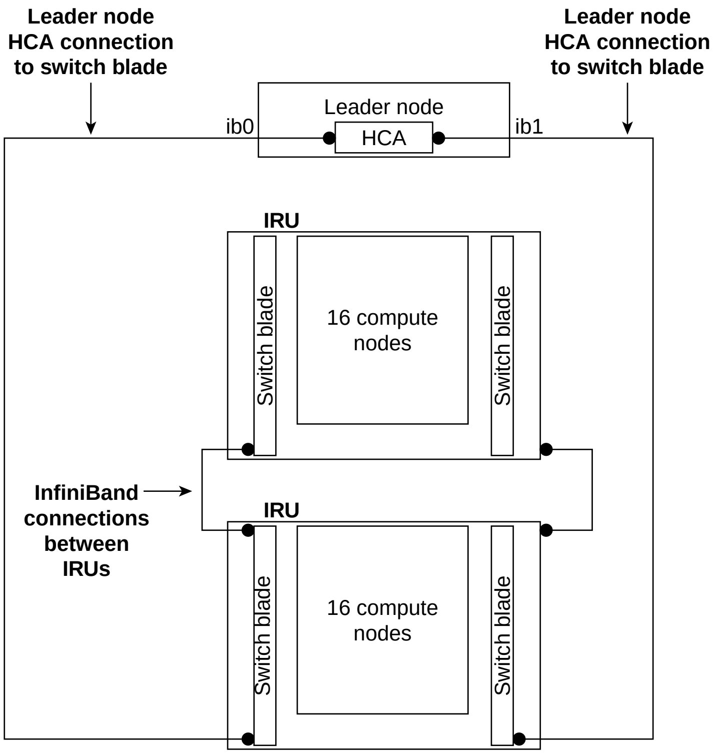


Figure A-1 Two InfiniBand Fabrics in a System with Two IRUs

With Scali Manage, the routing engine is chosen automatically based on the number of racks in the system. For up to two racks, the "`Min Hop`" algorithm is used. For more than two racks, the `''lash''` algorithm is used which enables LAyered SHortest Path Routing (LASH).

When the `lash` routing algorithm is used, the subnet managers need to be restarted after the entire Altix ICE system is up. To restart the subnet managers, perform the following command:

```
scalimanage-cli restartaltixiceopensm
```

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.

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 45.

Procedure A-1 Configuring and Initializing the InfiniBand Fabric Manually

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

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

   ```
   # ssh r01lead
   ```

   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 r01lead
PING r01lead.ice.americas.sgi.com (172.16.0.2) 56(84) bytes of data.
64 bytes from r01lead.ice.americas.sgi.com (172.16.0.2): icmp_seq=1 ttl=64 time=0.127 ms

--- r01lead.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:

```
r01lead:/ # 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:

```
r01lead:/ # 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:

   ```
   r01lead:/ # 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:

```
r01lead:/ # ibnetdiscover -l
Switch   : 0x08006900000000dc ports 24 devid 0xb924 vendid 0x2c9 "MT47396 Infiniscale-III Mellanox Technologies"
Switch   : 0x08006900000000a4 ports 24 devid 0xb924 vendid 0x2c9 "MT47396 Infiniscale-III Mellanox Technologies"
Ca       : 0x0030487aa7940000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa78c0000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0008f10403988198 ports 2 devid 0x6278 vendid 0x8f1 "service0-ib0 HCA-1"
Ca       : 0x0030487aa7840000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa79c0000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa7900000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0030487aa7980000 ports 1 devid 0x6274 vendid 0x2c9 " HCA-1"
Ca       : 0x0008f104039881a8 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"
```

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

```
r01lead:/ # 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]
 --switch-map  specify a switch-map file
```

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

Appendix B

InfiniBand Fabric Troubleshooting

This appendix describes some useful utilities and diagnostics for trouble shooting the InfiniBand fabric.

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:

```
r01lead:~ # cd /usr/bin
r01lead:/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` Command" on page 56
- "`perfquery` Command" on page 58
- "`ibnetdiscover` Command" on page 59
- "`ibdiagnet` Command" on page 60

ibstat Command

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:

```
r01lead:/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:

```
r01lead:/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
```

`ibstatus` Command

You can use the `ibstatus` (less verbose that `ibstat`) command to show the link rate, as follows:

```
r01lead:/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:

```
r01lead:/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:

```
r01lead:/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:

```
r01lead:/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:

```
r01lead:/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"

r01lead:/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:

```
r01lead:/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:

```
r01lead:/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  :-

r01lead:/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.
```

Index