sgi®

SGI® Management Center for SGI® ICE™ X

007–5787–002

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

c3:

conserver:

Record of Revision

Version	Description
001	March 2012 Original publication.
002	November 2012 Revised to support the SGI Management Center 1.7 release.

Contents

Figures

Examples

Procedures

About This Guide

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

Note: This manual is under development. The information in this revision that describes system installation is a working draft. Please check the SGI Technical Publications Library at the following website for recent updates to this manual:

http://docs.sgi.com

This manual contains the following chapters:

Related Publications

The following additional documentation might be useful to you:

- *SGI Management Center Installation and Configuration*

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

- *SGI Management Center System Administrator's Guide*

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

- *SGI ICE X System Hardware User Guide*

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

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

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

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

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

- *SGI Performance Suite X.X Start Here*

  This manual lists the current SGI software and hardware manuals.

- Documentation from other sources:
  - Novell documentation for SUSE Linux Enterprise Server 11 (SLES 11)
  - Red Hat documentation for Red Hat Linux Enterprise Server 6 (RHEL 6)
  - Intel compiler documentation
  - Intel documentation about Xeon architecture

Obtaining Publications

You can obtain SGI documentation in the following ways:

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

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

Conventions

The following conventions are used throughout this document:

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

Reader Comments

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

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

techpubs@sgi.com

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

  SGI
  Technical Publications
  46600 Landing Parkway
  Fremont, CA 94538

SGI values your comments and will respond to them promptly.

SGI ICE X System Overview

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

- "Hardware Overview" on page 1
- "Cluster Management Commands" on page 18
- "Software Management Overview" on page 19
- "VLAN Ethernet Network Configurations" on page 21
- "System Control Configuration" on page 33

Hardware Overview

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

- "Basic System Building Blocks" on page 1
- "System Nodes" on page 14

For detailed hardware descriptions, see the *SGI ICE X System Hardware User Guide.*

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

Basic System Building Blocks

The SGI ICE X system is a blade-based, scalable, high density compute system. The basic building block is the 21U high blade enclosure pair (see Figure 1-1 on page 2).

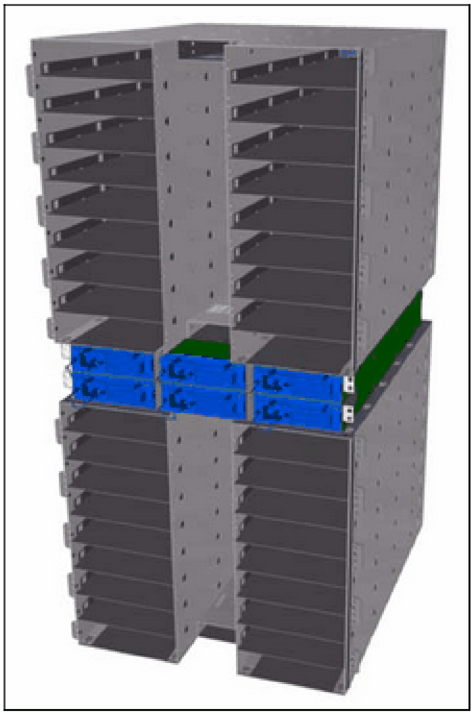

Figure 1-1 SGI ICE X Blade Enclosure Pair

The blade enclosure pair provides power, cooling, system control, and the network fabric for 36 compute blades. The backplane is configured as an enhanced 3D hypercube. Within the backplane, it is blocking. It is nonblocking only when leaving the blade enclosure. A single rack can accommodate up to 72 compute blade slots within two blade enclosure pairs.

Figure 1-2 on page 3 shows a functional block diagram of an SGI ICE X system D-rack blade enclosure pair. Figure 1-2 on page 3 shows the blade enclosure slot numbering, power supply numbering, and CMC slots. Two blade enclosures pairs can reside in a custom designed 42U high rack. Figure 1-4 on page 9 shows a 42U D-rack and a 42U M-rack used for systems with twin blades.

Note: A blade enclosure contains two CMCs when residing in a 42U M-rack used for systems with twin blades, otherwise, an enclosure contains one CMC.

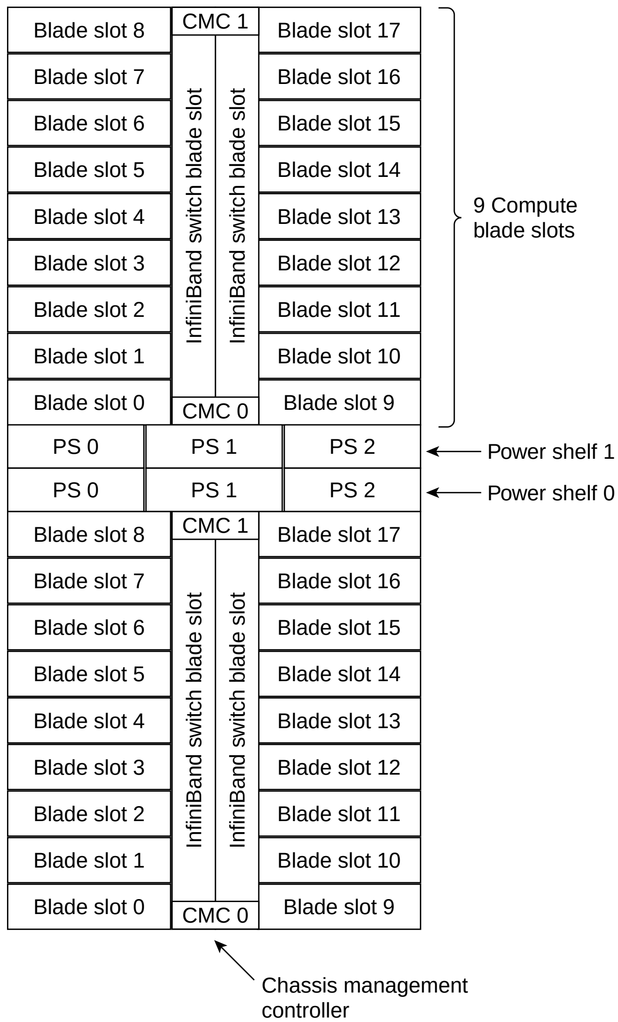

Figure 1-2 Basic System Building Blocks for SGI ICE X

Figure 1-3 on page 4 shows a blade enclosure and rack components.

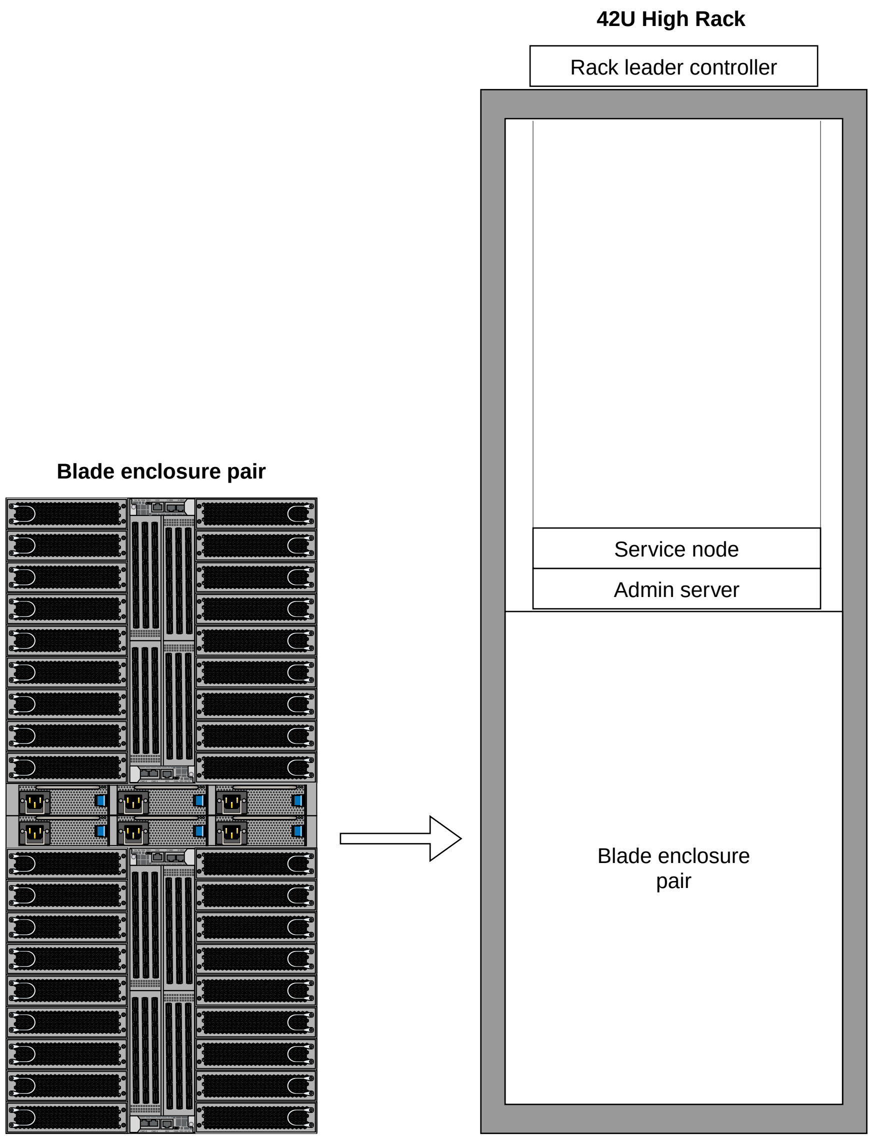

Figure 1-3 Blade Enclosure and Rack Components

This hardware overview section covers the following topics:

InfiniBand Fabric

The SGI ICE X system topology is based on a single-plane or dual-plane 4xFDR14 (Fourteen Data Rate, 14Gb/s data rate per lane) InfiniBand interconnect. Internal InfiniBand switch ASICs located in switch blades in the blade enclosure eliminate the requirement for external InfiniBand switches. The InfiniBand technology provides for fast communication between compute blades within the same chassis and also between compute blades in separate enclosures.

The InfiniBand switch blade provides the interface between compute blades within the same chassis and also between compute blades in separate enclosures.

The following topologies are supported:

- Hypercube
- Enhanced Hypercube
- All-to-All
- Fat Tree

Fabric management software monitors and controls the InfiniBand fabric. SGI ICE X systems are usually configured with two separate InfiniBand fabrics. These fabrics (also sometimes called InfiniBand subnets) are referred to as "`ib0`" and "`ib1`" within this document.

For information on MPI and MPT, see the *Message Passing Toolkit (MPT) User's Guide* available on the SGI Technical Publications Library at http://docs.sgi.com. The `ib1` fabric is reserved for storage related traffic. The default configuration for MPI is to use only the `ib0` fabric. For more information on the InfiniBand fabric, see Chapter 5, "System Fabric Management" on page 227.

Note: The “`ib0` fabric" is a convenient shorthand for "the fabric which is connected to the `ib0` interface on most of the nodes”. Particularly in the case of storage service nodes, there may be several interfaces called `ib0`, `ib1`, and so on, all of which are connected to the same fabric (see "Storage Service Node " on page 18 and "NAS Configuration for Multiple IB Interfaces" on page 119).

For performance reasons, it is often beneficial to use one fabric for message passing interface (MPI) traffic and the other for storage- related traffic. The default configuration for MPI is to use only the `ib0` fabric and storage uses the `ib1` fabric. For more information on the InfiniBand fabric, see Chapter 5, "System Fabric Management" on page 227.

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

Blade Enclosure Assembly

The blade enclosure pair is a 21U high assembly that provides power, cooling, system control, and network fabric for up to 36 compute blade slots via a backplane. A maximum of two blade enclosure pairs can be placed in a single rack. To create larger system configurations, multiple blade enclosures are connected together via external InfiniBand cables between the InfiniBand switch blades.

The blade enclosure chassis supports the following components (see Figure 1-1 on page 2 and Figure 1-2 on page 3):

- A maximum of 18 single-width compute blades.
- Two InfiniBand switch blades that create and internal 4xFDR topology. There are two versions of switch blades (see "InfiniBand Switch Blades" on page 7).
- One chassis manager (CMC) that enables system-control connection to each blade slot. Two CMCs for systems with twin blades.
- Rear backplane and fan-control board with a real-time clock.

InfiniBand Switch Blades

There are two versions of the switch blade for the SGI ICE X system, as follows:

- Standard (single InfiniBand ASIC) switch blade

  Two or four standard switch blades (18 external IB ports per switch blade) are used with each enclosure pair to support a single- or dual-plane topology.

  The standard switch blade contains one 4xFDR (ConnectX3) 36-port InfiniBand ASIC. The switch blade provides the interface between compute blades within the same blade enclosure and also between compute blades in separate blade enclosures.

  The standard switch blade ASIC connects to all 18 compute nodes on one plane and provides 18 external ports to extend the topology.

- Premium (Dual InfiniBand ASIC) switch blade

  Two or four IB switch blades (48 external IB ports per switch blade) are used with each blade enclosure pair to support a single- or dual-plane topology.

  Each premium switch blade has two 4x FDR ASICs and each ASIC connects internally via the backplane to nine compute blades. The two FDR ASICs are also connected internally with three x4 FDR IB links.

Compute Blades

A single rack can accommodate up to 72 compute blade slots within two blade enclosure pairs. The custom-designed 42U-high racks can be air cooled for small configurations or water cooled for large, highly dense configurations. One rack supports the following:

- A single rack can accommodate up to 72 compute blade slots within two blade enclosure pairs
- Maximum of 72 twin-node compute blades (144 total compute nodes)
- Maximum 36.864 TB of memory (72 x 512GB)

Three versions of Intel® based compute blades support the Intel 4-core, 6-core or 8-core Xeon® processor sockets and DDR3 memory DIMMs.

Power control for each blade is handled by its Baseboard Management Controller (BMC), also under direction of the rack leader controller (RLC). Once the RLC node has sent a request to 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 environmental and error log information about each blade.

Note: For detailed information about powering your system on or off, see the "Powering the System On and Off" section in chapter 1 of the *SGI ICE X Hardware User's Guide*.

Power Supply

Figure 1-4 on page 9 shows the power shelf configurations for the SGI ICE X D-Rack and M-Rack, respectively. In the D-Rack, two 9.5 rack-unit (9.5U) rack-mount enclosures are coupled with two shared and independently-scalable power shelves that provide power to both enclosures through the shared 12V power bus. Four power shelves (and up to 12 power supplies) power a 12V bus bar on each side of the enclosures in the M-Rack.

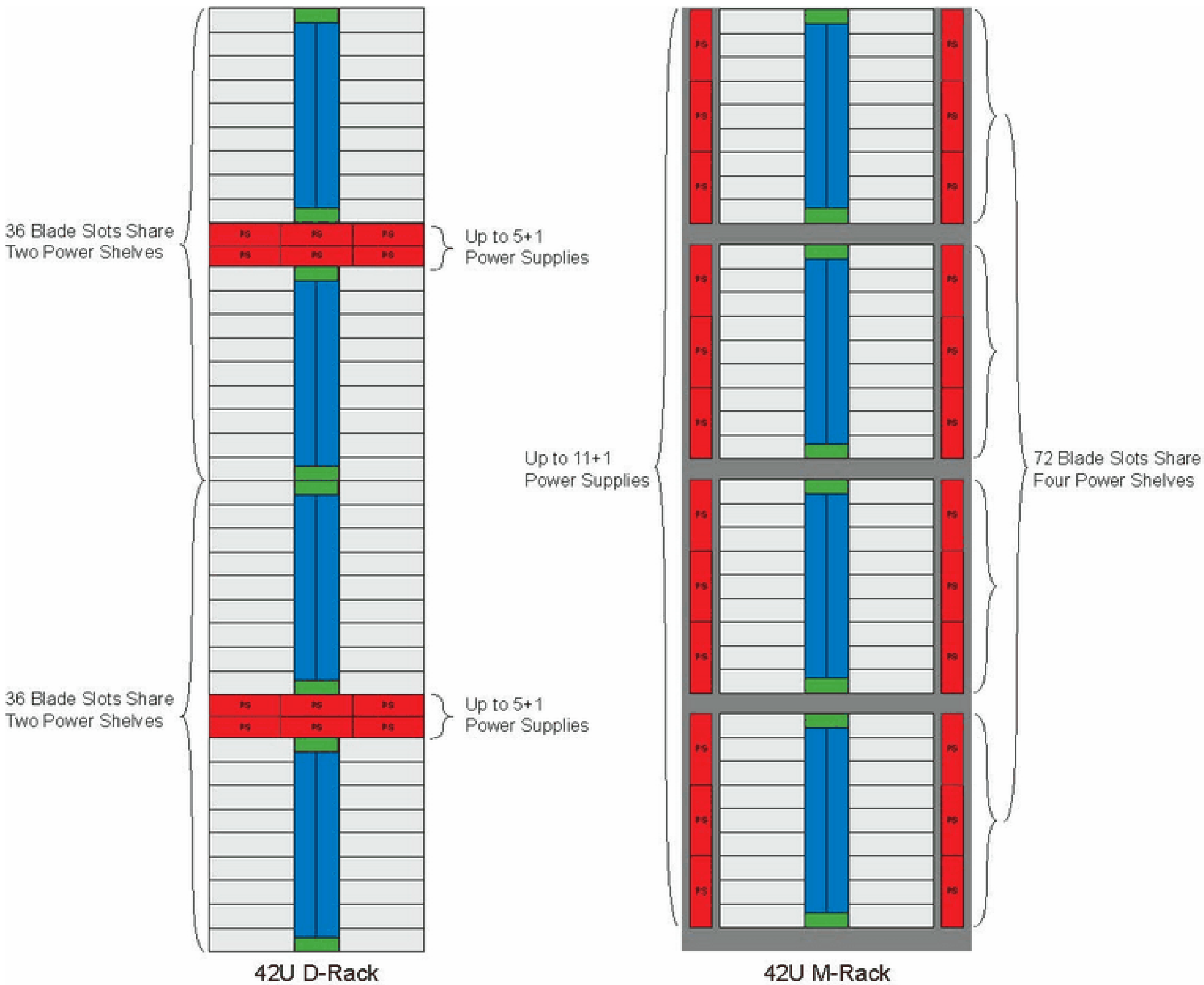

Figure 1-4 Power Shelf Assemblies

Four-tier, Hierarchical System Management Framework

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

- System admin controller (SAC) – One per system. Older revisions of the SGI documentation sometimes refer to this component as an *admin node*.

- Rack leader controller (RLC) – One per eight CMCs. Older revisions of the SGI documentation sometimes refer to this component as a *leader node*.
- Chassis management controller (CMC) – One per blade enclosure
- Baseboard Management Controller (BMC) – One per compute node, SAC, RLC, managed service node, InfiniBand blades and storage node

Unlike traditional, flat clusters, the SGI ICE X 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 (SAC)
- Rack leader controller (RLC)
- Service Nodes
  - Login
  - Batch
  - Gateway
  - Storage
  - Object Storage Server
  - Metadata Server

Figure 1-5 on page 11 shows the system management hierarchy.

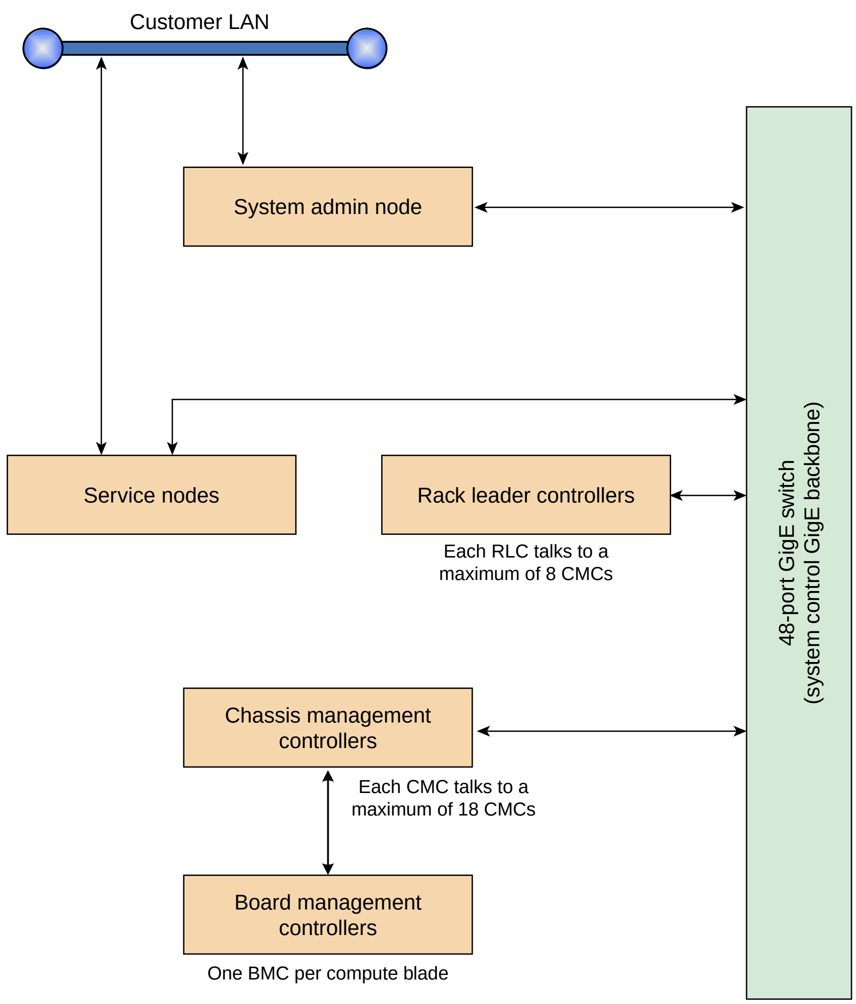

Figure 1-5 System Management Hierarchy

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

Systems configured with single node compute blades have one RLC per two racks. Systems configured with twin-blades (twin nodes) have one RLC per rack. The RLC holds the boot images for the compute blades and aggregates cluster management data for the rack. The RLCs are responsible for the compute nodes. They act as proxies for control and monitoring software (c3, `conserver`, `pdsh`, and so on). The RLCs monitor the compute nodes' BMCs helping to keep monitoring loads lighter.

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

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

For more information about the RLC and SAC, see "System Nodes" on page 14.

Chassis Management Controller (CMC) Blade

Each blade enclosure has two chassis management controller (CMC) slots, located directly above and below the InfiniBand network switch blades. One CMC per blade enclosure is required when using single-node compute blades. Two CMCs per blade enclosure are required when using twin-node compute blades.

The CMC supports powering up and down of the compute blades and environmental monitoring of all units within the blade enclosure. You can use the CMC to communicate with the compute nodes via IPMI protocol to monitor blade enclosure power and cooling.

The `/etc/sysconfig/module_id` file on the CMC stores the blade enclosure number.

Unlike previous SGI ICE systems, the CMCs do not manage VLANs and do not connect to each other. The CMCs all connect to the top level InfiniBand switch fabric.

Figure 1-6 on page 13 shows a front view of the CMC.

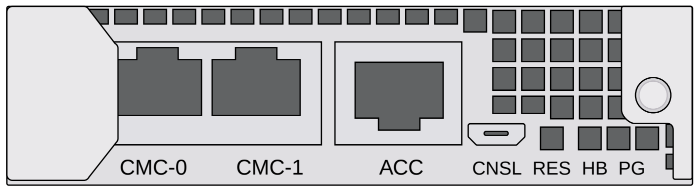

Figure 1-6 Chassis Management Controller Front View

The ports on the CMC are used, as follows:

- **CMC-0**

  Primary CMC connection. It connects to the rack leader controller (RLC) via the 48-port management switch.

- **CMC-1**

  Secondary CMC connection to the RLC via the 48-port management switch. It is used with redundant switch configurations.

- **ACC**

  Accessory port. It is used as a direct connection to the microprocessor for service.

- **CNSL**

  Console connection. It is used for troubleshooting.

- **RES**

  Reset switch. When you depress this switch, it resets the CMC microprocessor.

- **HB**

  Heartbeat LED. A green flashing LED indicates that the CMC is running.

- **PG**

  Power good LED. The LED is illuminated green when power is present.

System Nodes

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

- "System Admin Controller" on page 14
- "Rack Leader Controller (RLC)" on page 15
- "Chassis Management Controller (CMC) Blade" on page 16
- "Compute Node" on page 16
- "Login Service Node" on page 17
- "Batch Service Node" on page 17
- "Gateway Service Node " on page 17
- "Storage Service Node " on page 18
- "Object Storage Server" on page 18
- "Metadata Server" on page 18

System Admin Controller

The system admin controller (SAC), is used by a system administrator to provision (install), manage, and monitor the SGI ICE X system using SGI Management Center (SMC) software. There is only one SAC per SGI ICE X system and it cannot be combined with any other nodes. A GigE connection provides the network connection between the SAC, rack leader controllers (RLCs), and service nodes.

The SAC does not have a direct Gigabit Ethernet network access to compute nodes and is not on the InfiniBand Fabric.

The SAC provisions and manages the RLCs, compute nodes and service nodes. It receives and holds aggregated management data from the RLC. The SAC is an appliance node. It always runs software specified by SGI.

You can use the `cimage` command from the SAC to push compute node images to RLCs and use `rsync` to synchronize the compute images to the RLCs (see "`cimage` Command" on page 21). The `cinstallman` command is used for all package updates including those within images, running nodes, including the SAC itself (see "Installing Updates on Running System Admin Controller (SAC), Rack Leader Controller (RLC), and Service Nodes " on page 130). If you are installing a system from scratch, the `crepo` command is one of the first commands you will run after the SAC has been installed (see "Installing SMC for ICE X System Admin Controller (SAC) Software" on page 91).

Rack Leader Controller (RLC)

The RLC manages the nodes in a blade enclosure. There is one RLC per eight CMCs. Only the first two RLCs have InfiniBand cards to facilitate InfiniBand services. The RLCs run the OpenSM InfiniBand Subnet Manager (OSM) software to act as the subnet master or backup subnet master for the InfiniBand fabric. Typically, the `r1lead` RLC is primary for the `ib0` fabric and secondary for the `ib1` fabric. Typically, the `r2lead` RLC is primary for the `ib1` fabric and secondary for `ib0`. In a single-plane SGI ICE X system configuration, the `r1lead` RLC is usually primary and the `r2lead` RLC is secondary.

The RLC is provisioned and functioned by the system admin controller (SAC). The RLC is an appliance node. It always runs software specified by SGI. The RLC does the following:

- Runs OSM software for InfiniBand fabric subnet master and backup.
- Provides the system boot image, local storage, `/`, `/etc`, `/var`, `/root`, and swap file system for compute nodes.
- Monitors and processes data from the blade enclosures and compute nodes to the SAC.
- Consolidates and forwards data from the blade enclosures and compute nodes to the SAC upon request.
- Contains the following network connections:
  - GigE connection to the SAC (`HEAD VLAN`)
  - GigE connection to compute nodes (`RACKx VLAN`)
  - Ethernet connection to the CMC in the rack, such as, `i0c`, `i1c`, `i2c`, `i3c`, and so on
  - The first two RLCs have InfiniBand PCI cards for access to the rest of the SGI ICE X system.

The RLC can contain multiple images for the compute nodes. "Customizing Software On Your SGI ICE X System" on page 161 describes how you can clone and customize compute node images.

Chassis Management Controller (CMC) Blade

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

Each blade enclosure has two chassis management controller (CMC) blades located directly below compute blade slot 0 as shown in Figure 1-2 on page 3. 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).

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 RLC. Once the leader node 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

Users submit MPI jobs to run in parallel on the SGI ICE system compute nodes using a public network connection via the service node. The service node provides login services and a batch scheduling service, such as, Altair® PBS™ Professional for job scheduling and workload management. The compute nodes are controlled and monitored by the rack leader controller (RLC) for their rack. Compute nodes are booted and mount the shared, read-only portion of the root file system from the RLC. The RLC provides the network connections to the compute nodes in the same rack and to RLCs in other racks that then provide the network connections to the compute nodes in their racks. These network connections are via the InfiniBand fabric. The system admin controller (SAC) does not communicate directly with the CMC or compute blades. Actions for the CMC and compute blades are sent to the appropriate RLC, which communicates to the appropriate CMC and compute blades. The compute nodes do not communicate directly to the CMC or the SAC, or RLCs outside their rack. Compute nodes can have two hard disk drive (HDD) or solid-state drive (SSD) attached. SGI ICE X systems can have striped (raid0) disks if there are two disks. Disks are only for scratch or swap space. For more information on swap space management, see "Changing the Size of Per-node Swap Space" on page 208.

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

- "Changing the Size of `/tmp` on Compute Nodes" on page 205

- "Enabling or Disabling the Compute Node iSCSI Swap Device" on page 208
- "Changing the Size of Per-node Swap Space" on page 208

Login Service Node

The login service node allows users to login into the system to create, compile, and run applications. The login node is usually combined with batch and gateway service nodes for most configurations. The login service node is connected to the SGI ICE X system via the InfiniBand fabric, GigE connection to the RLCs and other service nodes and a GigE connection to the public customer network as shown in Figure 1-5 on page 11.

End users log into the login service node to compile and monitor MPI programs. The login service node can run a batch scheduler, such as, PBS Professional, Simple Linux Utility for Resource Management (SLURM), or Torque.

Additional login service nodes can be added as the total number of user logins grow.

Batch Service Node

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

Gateway Service Node

The gateway service node is the gateway from the InfiniBand fabric to services on the public network such as storage, lightweight directory access protocol (LDAP) services, and file transfer protocol (FTP).

In smaller systems, the gateway node can be combined with the login or batch node. Large configurations will normally have the gateway node as a separate server. The gateway node also acts as a gateway from IPoverIB to IPoverGigE or IPoverIB to IPover10GigE. This includes the following protocols:

- Network File System (NFS)
- Network Address Translation (NAT)

- Network Information Service (NIS)

Storage Service Node

The storage service node is a network-attached storage (NAS) appliance bundle that provides InfiniBand attached storage for the SGI ICE X system. There can be multiple storage service nodes for larger SGI ICE system configurations.

Object Storage Server

The Object Storage Server (OSS) is one of the elements of a Lustre File Storage system. The OSS is managed by the SGI ICE X management network.

Metadata Server

The Metadata Server (MDS) is one of the elements of a Lustre File Storage system. The MDS is managed by the SGI ICE X management network.

Cluster Management Commands

The SGI ICE X cluster management commands are located in the `/opt/sgi/sbin` directory of the system admin controller (SAC), login, and rack leader controllers (RLCs), as follows:

- `configure-cluster`

  Performs initial software, network, and NTP configuration of a new cluster. For more information, see "Running the Cluster Configuration Tool" on page 61.

- `discover`

  Discovers and provisions RLCs and service nodes. For more information, see "`discover` Command" on page 84.

- `cpower`

  Powers on or off the system, racks, service or compute nodes, and individual system comments and check their status. For more information, see "`cpower` Command" on page 184.

- `console`

Establishes a SOL (serial console) connection to the specified node. For more information, see "Console Management" on page 201.

- `crepo`

  Maintains a software image repository in the `/tftpboot` directory. For more information, see "`crepo` Command" on page 20 and "`crepo` Command" on page 150.

- `cinstallman`

  Builds custom images and maintains and updates software images on nodes. For more information, see "`cinstallman` Command" on page 20 and "`cinstallman` Command" on page 154.

- `cimage`

  Creates and pushes software images to RLCs from which the compute nodes use for booting. For more information, see "`cimage` Command" on page 21 and "`cimage` Command" on page 169.

- `cadmin`

  Modifies cluster attributes, such as, network specifics. For more information, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195.

Software Management Overview

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

- "`crepo` Command" on page 20
- "`cinstallman` Command" on page 20
- "`cimage` Command" on page 21

Compute node images are staged on the RLCs and pushed out to compute nodes using the `cimage` command. RLCs and service nodes do not require staging. RLCs and service nodes are installed from the system admin controller (SAC).

crepo Command

If you are installing a system from scratch, the crepo command is one of the first commands you will run after the system admin controller (SAC) has been installed (see "Installing SMC for ICE X System Admin Controller (SAC) Software" on page 91). You can use the crepo command to manage software repositories, such as, SGI Foundation, SMC for SGI ICE, SGI Performance Suite, and the Linux distribution(s) you are using on your system. You also use the crepo command to manage any custom repositories you create yourself. The crepo command stores RPMs from SGI media, distribution media, and is used to create images to update SAC, rack leader controller (RLC), and service nodes.

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

You can use the crepo --show command to view the location of the repositories on the SAC.

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

For more information, see "crepo Command" on page 150.

cinstallman Command

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

When SGI releases updates, you may run sync-repo-updates to download the updated packages that are part of a patch. The sync-repo-updates command automatically positions the files properly under /tftpboot (see "Overview of Installing SMC for SGI ICE Patches" on page 123).

Once the local repositories contain the updated packages (see "Update the Local Package Repositories on the System Admin Controler (SAC)" on page 124), it is possible to update the various SGI ICE X system admin controller (SAC), rack leader controller (RLC), and managed service node images using the cinstallman command. The cinstallman command is used for all package updates including those within images, running nodes, including the SAC itself (see "Installing Updates on Running System Admin Controller (SAC), Rack Leader Controller (RLC), and Service Nodes " on page 130).

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

For more information on the `cinstallman` command, see "`cinstallman` Command" on page 154.

`cimage` Command

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

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

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

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

For more information on the `cimage` command, see "`cimage` Command" on page 169.

VLAN Ethernet Network Configurations

This section describes the VLAN system management network for the SGI ICE X system.

VLAN Overview

The VLAN Ethernet network configurations for SGI ICE X have changed from previous SGI ICE systems. The following VLANs are configured across the Ethernet connections:

- HEAD VLAN

  The system admin controller (SAC), all rack leader controllers (RLCs), and all service nodes are in this VLAN

  This VLAN is always configured untagged. Any untagged packets coming in to an RLC, service, or SAC port will end up associated with VLAN HEAD.

  Head network, default for the top level switch is HEAD VLAN, as shown in Figure 1-7 on page 22.

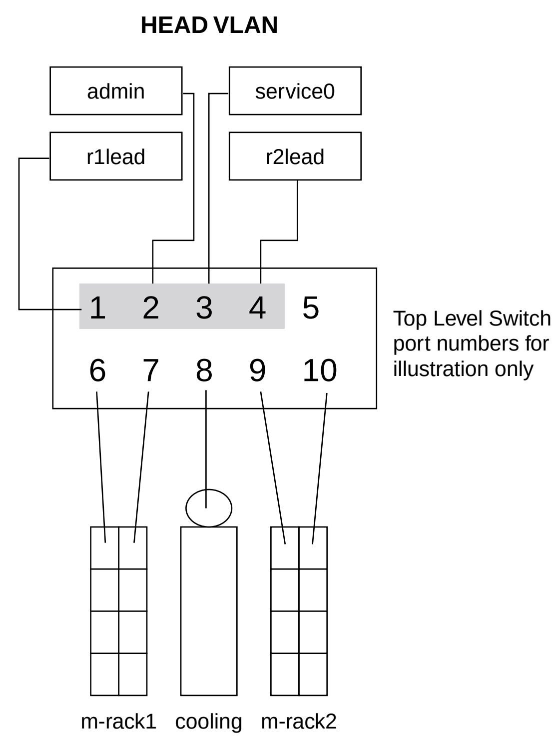

Figure 1-7 HEAD VLAN Ethernet Connections

- RACKx VLAN

  There is one RACKx VLAN for each rack in the system.

  The CMC internal switch cascades to the top level switch and the switch ports for all CMCs in rack x are configured with VLANx.

  Rack leader controllers (RLCs) reside both on VLAN HEAD and on the RACKx VLAN to communicate with the compute nodes in their rack.

- Compute node hosts and baseboard management controllers (BMCs) reside on the same VLAN.
- Compute node BMCs and compute node interfaces share the same physical CMC switch port. They are on the same VLAN but still have distinct IP ranges to make them managed similar to previous ICE 8400 and 8200 systems.

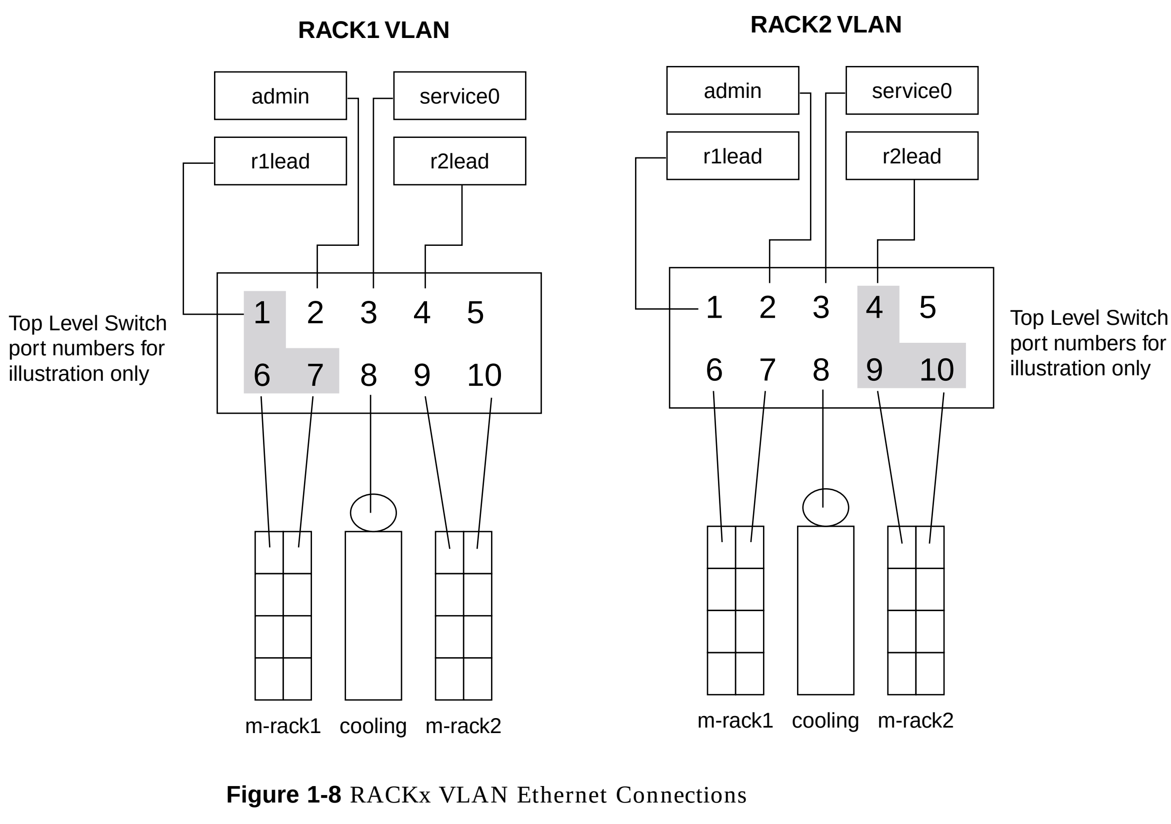

Figure 1-8 RACKx VLAN Ethernet Connections

System Admin Controller (SAC) VLAN Details

The SACs are on the HEAD VLAN.

On the SAC, the top level switch port is configured so that all untagged packets are associated with VLAN HEAD.

Service Node VLAN Details

Service nodes are on the HEAD VLAN.

The service node top level switch port is configured so that all untagged packets are associated with VLAN HEAD.

Cascaded CMC VLAN Details

Each CMC connects to a port on a top level switch.

The port is configured so that all traffic coming in and going to that port travels to VLAN RACKx by default.

The CMC gets its VLAN RACKx IP address using DHCP.

Compute Node VLAN Details

The compute nodes reside on the cascaded CMC switch (see "Cascaded CMC VLAN Details" on page 24).

For SGI ICE X systems, the compute node's BMC shares the physical Ethernet connection of the host interface. The compute node host and BMC traffic are commingled on VLAN RACKx. The rack leader controller (RLC) still keeps the BMCs and host interfaces on separate networks so that previous SGI ICE systems and SGI ICE X systems are managed similarly. However, both networks reside on VLAN RACKx.

Rack Leader Controller (RLC) Detail

RLC top level switch ports are configured for VLAN HEAD by default for all untagged packets.

The switch port is also configured to let in traffic for VLAN RACKx.

The switch port is configured in hybrid VLAN mode because there are both tagged and untagged packets in play.

The RLC has three management related IP addresses, as follows:

- An IP address on VLAN HEAD
- Two IP addresses on VLAN RACKx
  - BMC network
  - GBE network
- This approach keeps the management and IP networks similar to SGI ICE 8400 even though SGI ICE 8400 systems have distinct VLANs for BMC and GBE. These SGI ICE 8400 distinct VLANs are fully managed within the CMC for ICE 8400 system.

The RLC Ethernet interface for VLAN RACKx are configured with VLAN tagging and tie these networks to VLAN RACKx.

SGI ICE X IP Address Ranges and VLANs

This section describes SGI ICE X IP address ranges and VLANs for cluster management and application use.

Table 1-1 on page 25 shows cluster-wide IP address ranges for cluster management software.

Table 1-1 Cluster Management Cluster-wide IP Address Ranges wide

VLAN Name	Subnet Name	IP Range	Nodes
HEAD	head	172.23.0.0/16	System admin controller (SAC), service, rack leader controller (RLC)
HEAD	head-bmc	172.24.0.0/16	SAC, service, RLC BMCs (see Note 1)

Note 1: Separate IP subnet but lives on HEAD VLAN; super-netted

Table 1-2 on page 26 shows cluster-wide IP address ranges for cluster application software.

Table 1-2 Application Software Cluster-wide IP Address Ranges

VLAN Name	Subnet Name	IP Range	Nodes
IB0	ib-0	10.148.0.0/16	Service, some RLCs, compute blades (see Note 1)
IB1	ib-0	10.149.0.0/16	Service, some RLCs, compute blades (see Note 1)

Note 1: Only RLCs providing `ib` subnet services need to connect.

Table 1-3 on page 26 shows per-rack IP address ranges for cluster management software.

Table 1-3 Per-rack IP Address Ranges for Cluster Management Software

VLAN Name	Subnet Name	IP Range	Nodes
RACK1	gbe	192.168.159.0/24	RLC, CMCs, and compute nodes in Rack 1
RACK1	bmc	192.168.160.0/24	RLC, CMCs, and compute nodes BMCs in Rack 1 (see Note 1)

VLAN Name	Subnet Name	IP Range	Nodes
RACK2	gbe	192.168.159.0/24	RLC, CMCs, and compute nodes in Rack 2
RACK2	bmc	192.168.160.0/24	RLC, CMCs, and compute nodes BMCs in rack 2 (see Note 2)
RACKn	gbe	192.168.159.0/24	RLC, CMCs, and compute nodes in rack n
RACKn	bmc	192.168.160.0/24	RLC, CMCs, and compute nodes BMCs in rack n (See Note 3)

Note 1: Separate IP subnet but lives on RACK1 VLAN; super-netted.

Note 2:: Separate IP subnet but lives on RACK2 VLAN; super-netted

Note 3: Separate IP subnet but lives on RACKn VLAN; super-netted

SGI ICE X Naming Conventions

Some nNaming conventions for SGI ICE X have been carried forward from previous SGI ICE systems for consistency and for existing customers with SGI ICE hardware. This section covers some additions and small changes.

External InfiniBand switches have a new naming convention (`ibswitchX`).

Internal InfiniBand switches have names like `r1i0s0-bmc`, `r1i0s1-bmc`, and so on.

An individual rack unit (IRU) is called a blade enclosure (see "Blade Enclosure Assembly" on page 6) on SGI ICE X systems but the same syntax is used; that is, rack 1, blade enclosure 0, switch 1

For twin blades (available in a later release), the compute blades are divided between CMCs to maintain the original naming. Node A in twin blade A goes with CMC A. Node B goes with CMC B. Name CMC A and CMC B separate blade enclosures in system naming convention. When you have twin blades installed, you could have the following:

```
r1i0n0 twin blade 0, node a
r1i1n0 twin blade 0, node b
r1i0n1 twin blade 1, node a
r1i1n1 twin blade 1, node b
```

For SGI ICE X systems, a blade enclosure (previously called an individual rack unit (IRU) can contain 18 blades, for example:

```
r1i0n0...r1i0n17
```

A rack with twin blades will look like a system with eight blade enclosures instead of four blade enclosures and has a total of 144 blades.

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 system admin controller (SAC) and service node nodes can communicate with the rack leader controllers (RLCs) over the `VLAN_HEAD` virtual network. The system component terms used in this section are as follows:

Node	Refers to a building block within an SGI ICE system (see "System Nodes" on page 14)
Connection name	Denotes a resolvable name associated with an IP network
Node name	Represents system-wide unique identifier for the building blocks of the SGI ICE system. These IDs are partly not route-able. See "Non-resolvable Names" on page 30.

Hostname	Returns string of the hostname command. Is technically independent from the other names.

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

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

HEAD VLAN Network Connections

Table 1-4 on page 29 shows the HEAD VLAN network connection names.

Table 1-4 VLAN_HEAD Connections

Node	Connection Name
System admin controller (SAC)	admin
Service	serviceX serviceX-bmc
Rack leader controller (RLC)	rXlead rXlead-bmc

There is one SAC per system. You can have multiple service nodes labelled service0, service1, and so on. The BMC controllers for managed service nodes are accessible inside the network. BMCs for unmanaged service nodes are normally configured on the external network. For more information on managed service nodes, see "Installing Software on the Rack Leader Controllers (RLCs) and Service Nodes" on page 92.

VLAN RACKx Network Connections

Table 1-5 on page 30 shows the VLAN_GBE network connections.

Table 1-5 VLAN RACKx Network Connections.

Node	Connection Name	Node Name
Rack leader controller (RLC)	`lead-eth`	`rXlead`
Blade	`iYnZ-eth`	`rXiYnZ`

The VLAN RACKx is entirely internal to each rack . The naming scheme is replicated between each rack, so the name `i2n4-eth` (identifying the `VLAN RACKx` 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. Note that two IRUs on SGI ICE X systems are called a blade enclosure pair.

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

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

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

Non-resolvable Names

Sometimes a rack, an IRU, or a CMC needs to be uniquely identified within the SGI ICE X system. Table 1-6 on page 31 shows the names that may be used for this, but there is no IP address associated with them. Therefore, DNS lookup will not succeed for these names. The names are used by certain SGI ICE X management tools and are parsed internally to indicate which rack leader controller (RLC) to use in order to connect to the destination system.

Table 1-6 Non-resolvable Names

Node	Node Name
Rack	`rX`
IRU	`rXiY`
CMC	`rXiYc`

Hostnames

Hostnames are distinct from the non-resolvable names and are shown in Table 1-7 on page 31. 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 1-7 Hostnames

Node	Hostnames
System admin controller (SAC)	user assigned
Rack leader controller (RLC)	`rXlead`
Blade	`rXiYnZ`
CMC	`rXiYc`
Service	user assigned (see Note below)

Note: By default, the host name for service nodes follow the convention serviceX. However, host names of service nodes or SACs can be changed using the `cadmin` command (see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195).

The hostname gets the A record and name -ib0 gets a CNAME alias. Additionally, if you changed the hostname from the SMC for SGI ICE X node name, there will be CNAME alias for the
SMC for SGI ICE X node name, as well.

The zone looks similar to the following:

```
r1lead          IN      A       10.148.0.1
r1lead-ib0             IN      CNAME   r1lead.ice.mycompany.com.
r1lead-ib1             IN      A       10.149.0.1
r1i0n0                 IN      A       10.148.0.2
r1i0n0-ib0             IN      CNAME   r1i0n0.ice.mycompany.com.
r1i0n0-ib1             IN      A       10.149.0.2
r1i0n1                 IN      A       10.148.0.3
r1i0n1-ib0             IN      CNAME   r1i0n1.ice.mycompany.com.
r1i0n1-ib1             IN      A       10.149.0.3
[...]
```

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

InfiniBand Network

The InfiniBand fabric is connected to service nodes, rack leader controllers (RLCs), and compute nodes, but not to the system admin controller (SAC) or to the CMCs. Table 1-8 on page 33 shows InfiniBand names. There are two IB connections to each of the nodes that use it. Since IB is not local to each rack, you must use the fully-qualified, system-unique node name when specifying a destination interface. It may be necessary to alias the rXiYnZ names (currently non-resolvable) to rXiYnZ-ib0 if this is needed by MPI. Technically, rXiYnZ from an RLC points at the VLAN_GBE interface for the compute blade while from a service or compute blade, rXiYnZ points to the ib0 interface.

In DNS, the rXiYnZ name is the A record, with the -ib0 address, rXiYnZ-ib0, the CNAME alias to the rXiYnZ A record. The same applies to service nodes (see "Hostnames" on page 31).

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

Table 1-8 InfiniBand Names

Node	Connection Name	Node Name
Service	`serviceX-ib0` `serviceX-ib1`	`serviceX`
Rack leader controller (RLC)	`rXlead-ib0` `rXlead-ib1`	`rXlead`
Blade	`rXiYnZ-ib0` `rXiYnZ-ib1`	`rXiYnZ`

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

System Control Configuration

The system control network for an SGI ICE X system can be configured in the following ways:

- Standard configuration

  The standard configuration has a single GigE fabric with a single connection to the CMCs, rack leader controllers (RLCs), and system admin controller (SAC) (see Figure 1-10 on page 36).

- Redundant management network (RMN)

  The number of GigE switches in the system control network is doubled. An RMN also has the following:

  – The GigE switches are stacked (using stacking cables).

  – The links from the CMCs are doubled.

  – There are links from the SAC, RLCs, and most service nodes that are doubled (see Figure 1-11 on page 38) .

    At this time, BMC connections are not doubled in the redundant configuration. Certain failures may cause temporary inaccessibility to the BMCs but the host interfaces will stay alive.

- Time Synchronization Enabled

The standard GigE switches are replaced with GigE switches that will support 1588/clock sync.

- SAC high availability (HA)

  An HA system control network includes the following:

  - A second SAC
  - An SGI InfiniteStorage 5000 (IS5000) storage solution
  - A redundant GigE fabric (see Figure 1-12 on page 40)

- RLC high availability

  The number of RLCs in the system is doubled and the a redundant GigE fabric is added

Figure 1-9 on page 35 shows a simple (non-redundant) GigE management network, which is the minimum configuration of the management network.

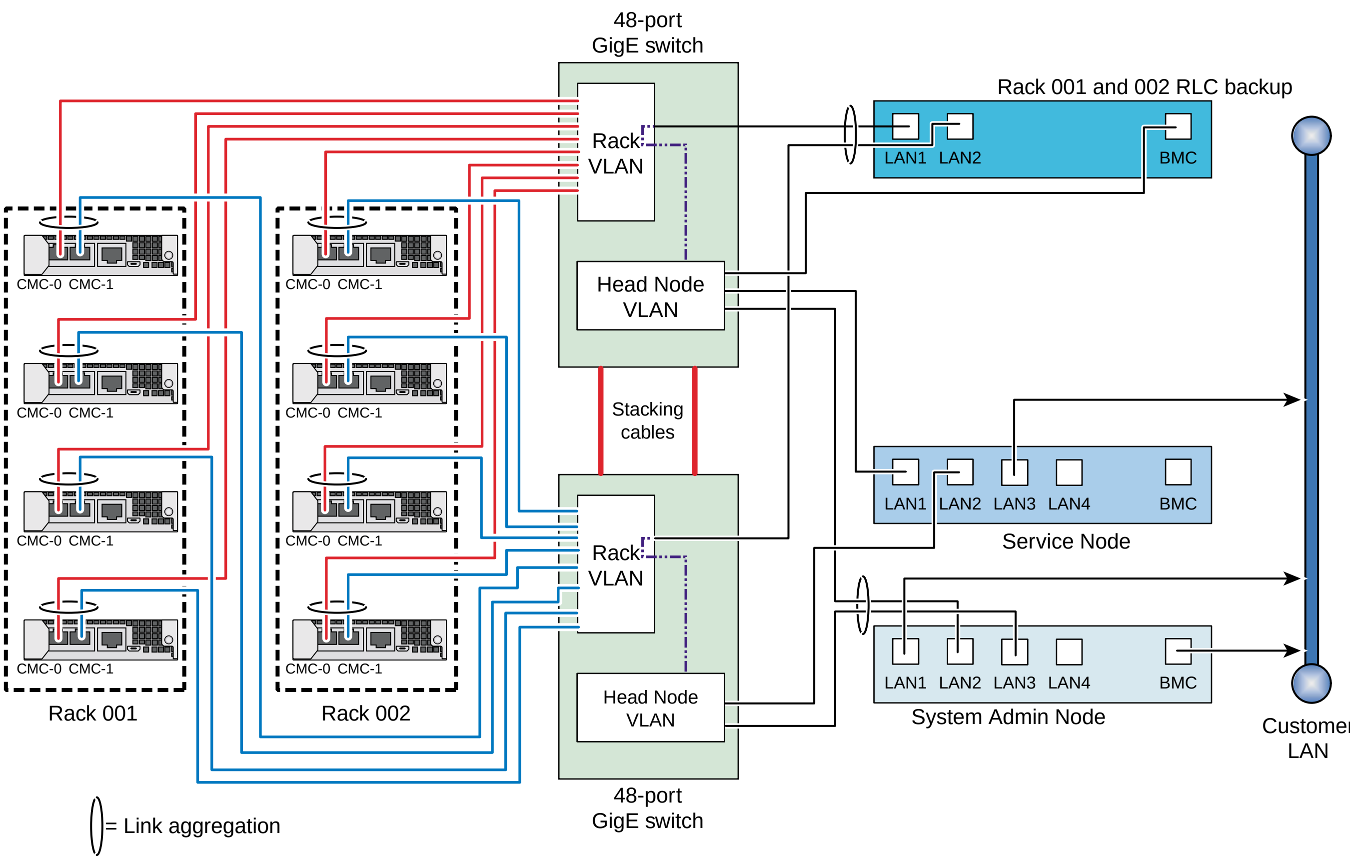

Figure 1-9

Figure 1-10 on page 36 shows a simple (non-redundant) GigE management network, which is the minimum configuration of the management network.

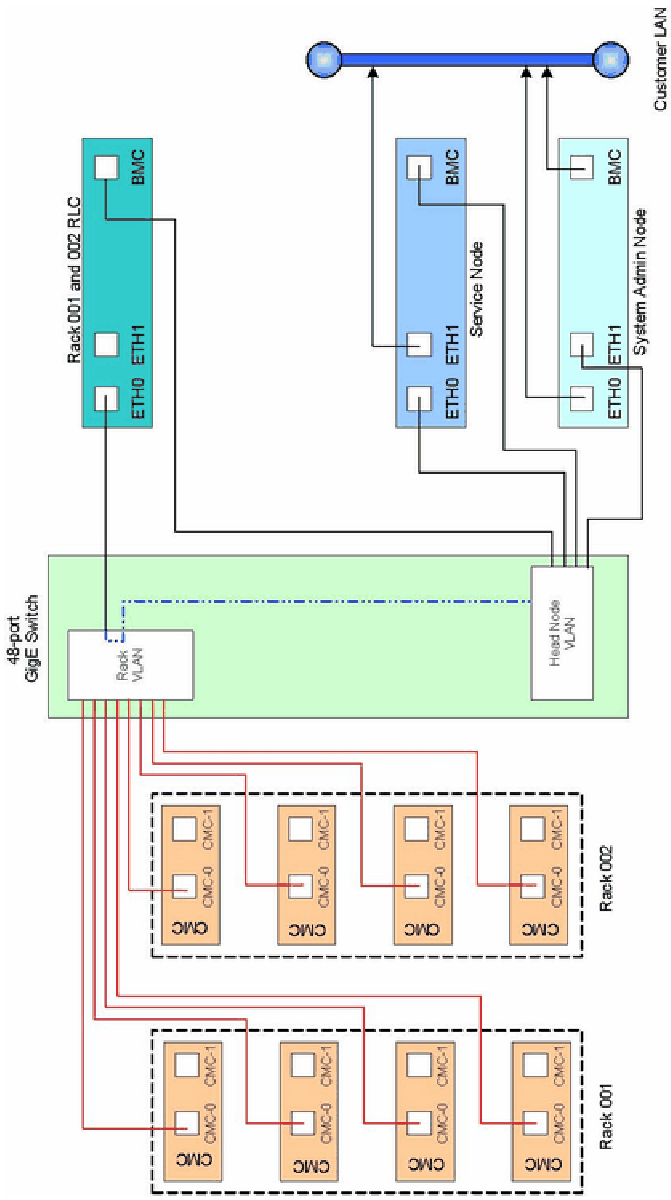

Figure 1-10 Standard (Non-Redundant) Management Network Topology

Figure 1-11 on page 38 shows a redundant management network (RMN). The RMN configuration is the default (normal) configuration for SGI ICE X systems. Note that the rack VLAN and the read VLAN are the same VLANs across both switches. When the two GigE switches are stacked, they essentially become a single large switch with one IP address.

Note: All service nodes in a redundant management network should connect to both management switches.

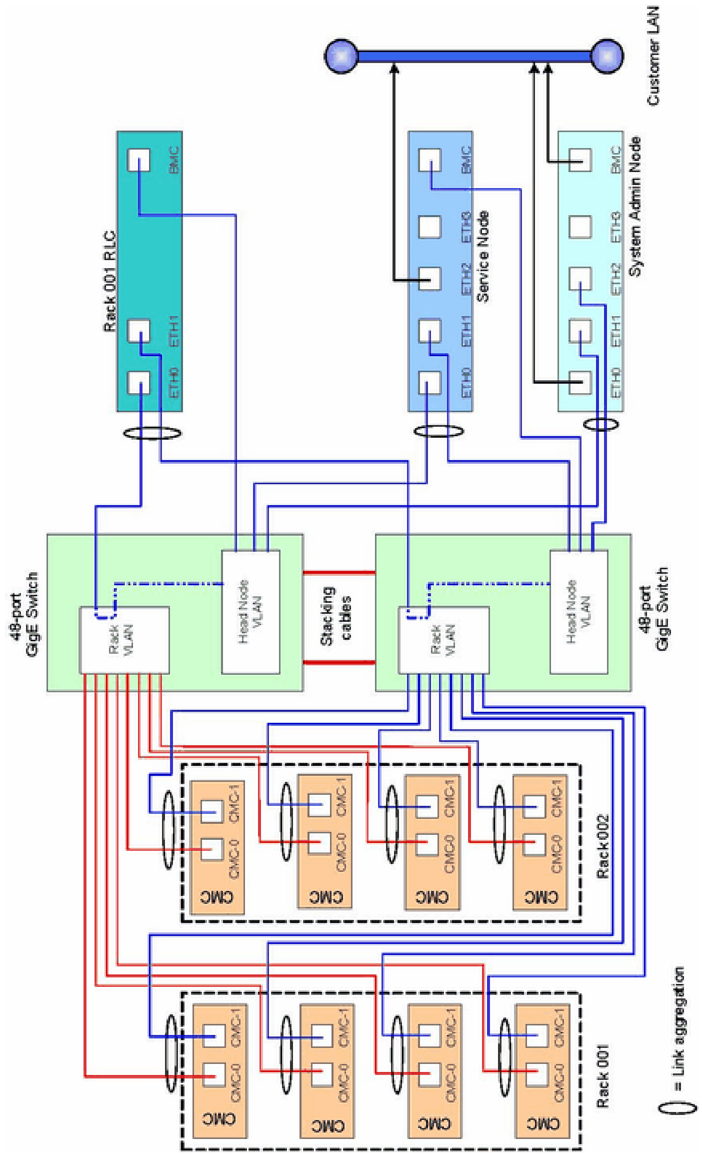

Figure 1-11 Redundant Management Network

A time synchronization enabled configuration uses IEEE 1588 Precision Time Protocol to synchronize the clocks in the compute blades to within 100 nanoseconds of each other.

All GigE switches in the system management network have to be replaced with GigE switches and related hardware that support 1588 time sync. The cabling does not change; just the switch type.

Time sync is used by test and measurement, power-line management, and industrial automation applications. It is also used by voice and video applications, or any application that requires a precision synchronous clock.

Figure 1-12 on page 40 shows a system control configuration with an HA SAC.

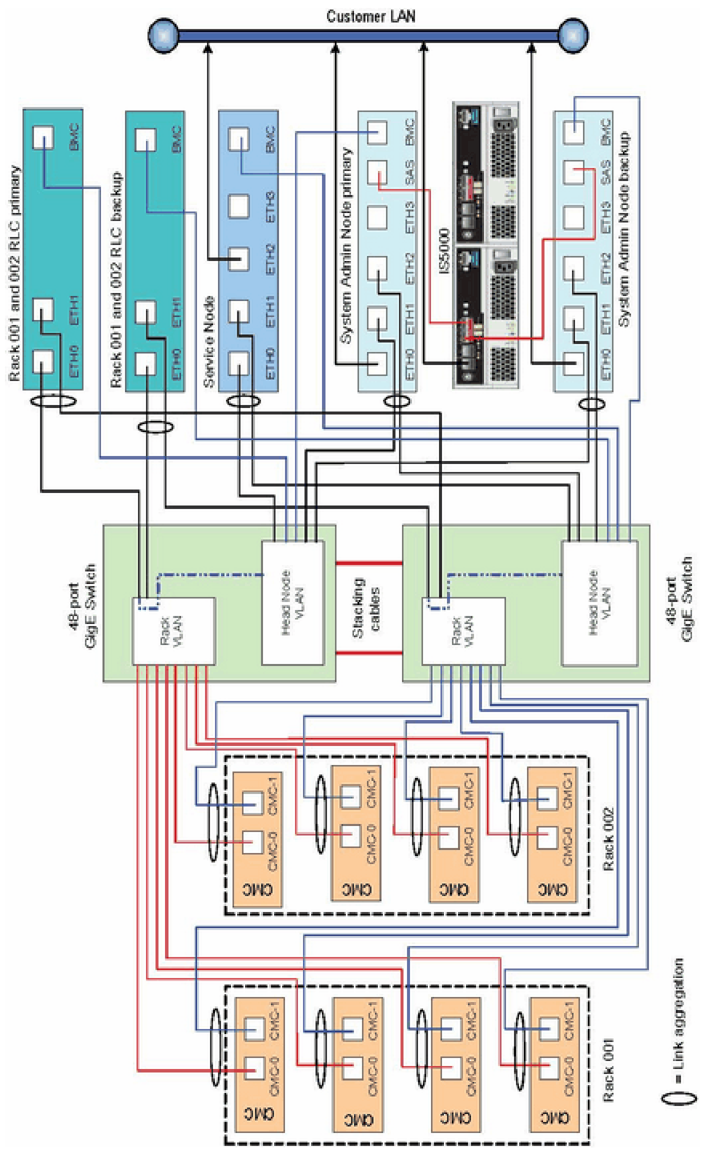

Figure 1-12 System Control Configuration with SAC High Availability

Figure 1-13 on page 41 shows a cascaded switch configuration.

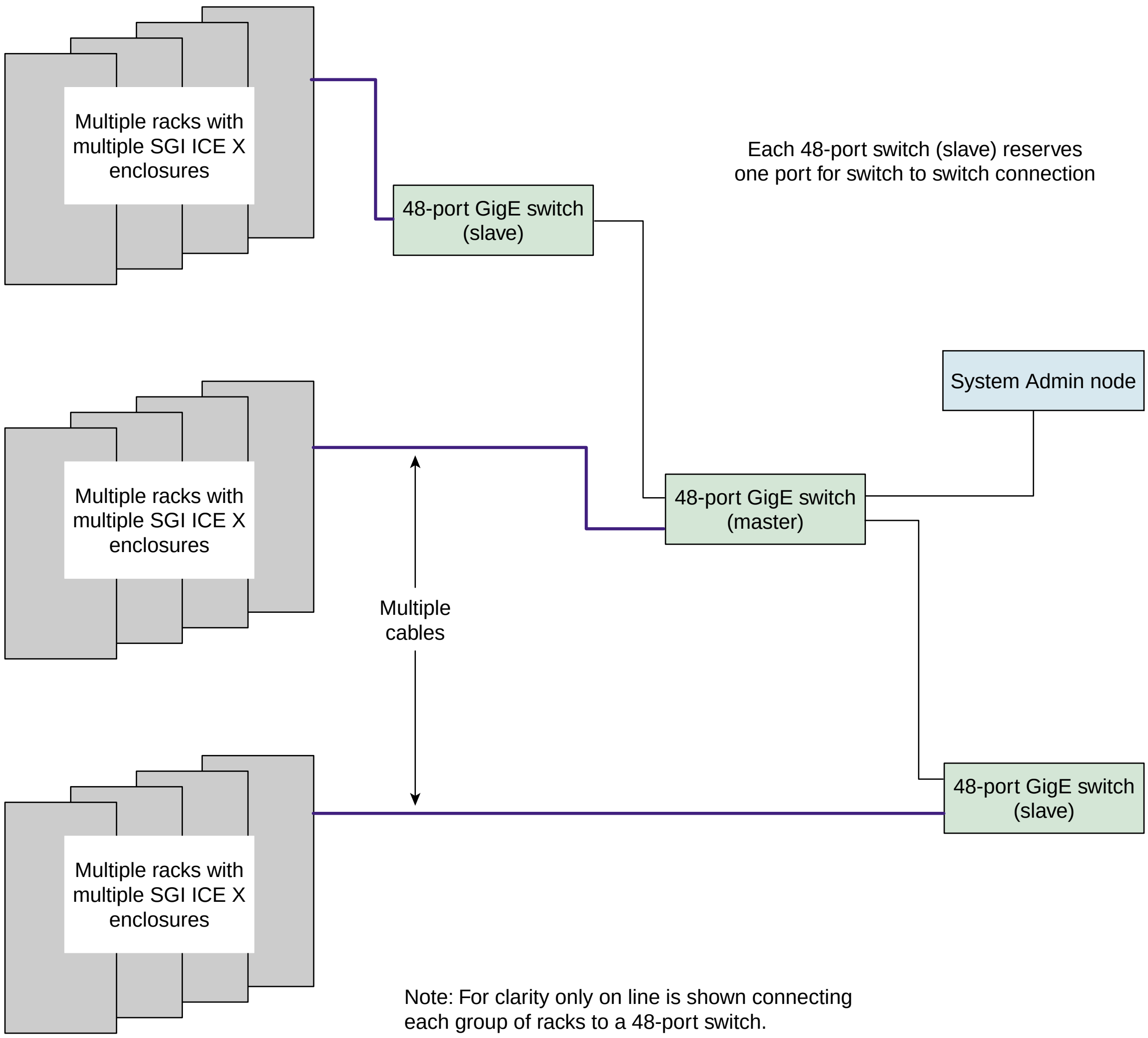

Figure 1-13 Cascaded Switch Network Configuration

Figure 1-14 on page 42 shows a redundant cascaded switch configuration.

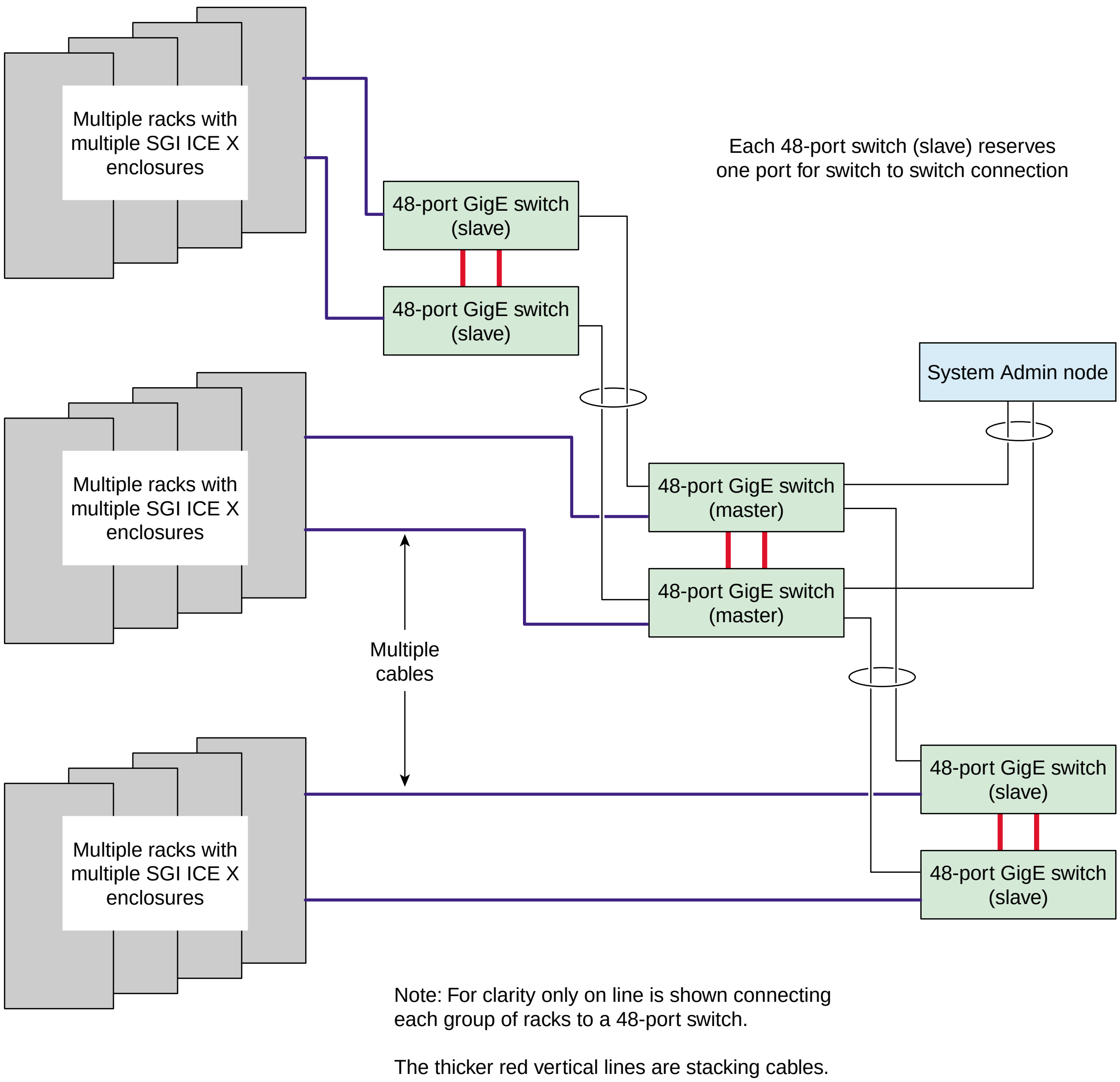

Figure 1-14 Redundant Cascaded Switch Configuration

Customizing a Factory-installed SGI ICE X System

Your SGI ICE X system was tested and configured at the factory. If you want to use the factory installation and minimize customization, use procedure in this chapter.

If you want to reinstall the operating system and all the rest of the software, use the procedure in the following chapter:

Chapter 3, "Installing and Configuring an SGI ICE X System" on page 47

Customizing a Factory-installed SGI ICE X System

This section describes how to configure an SGI ICE X system for your site. The procedure in this topic assumes that you want to preserve the factory-installed base configuration on the system admin controller (SAC) and to configure your system on top of that existing base configuration.

Procedure 2-1 To configure a factory-installed SGI ICE X System

1. Log into the SAC as the root user.

2. Use the `cpasswd` command to change the root passwords on all nodes.

   The `cpasswd` command changes the root password on the SAC and on all other nodes. To obtain a usage statement, type the following command:

   ```
   # cpasswd --h
   ```

3. Configure your house network on the first network interface card (NIC) on the SAC.

   The first NIC is `eth0`. Use the RHEL or SLES user interface to complete this step.

   For example, on a SLES SAC, start the YaST2 interface, and proceed to the **Network Settings** screen, and change one or more of the following settings:

   - For `eth0`, specify the following network settings: the fully-qualified domain name (FQDN), hostname, IP address, search domain, gateway, default route, netmask, DNS, and NTP server.

   - For the system itself, specify the time zone.

For information about how to change your network settings, see one of the following:

- "Installing SUSE Linux Enterprise Server (SLES)" on page 54.
- "Installing Red Hat Enterprise Linux (RHEL)" on page 58

Proceed as follows:

- If you did not change any network settings, the time zone, or the system-wide root password, proceed to the following step:

  Procedure 2-1, step 7 on page 44

- If you changed network settings, the time zone, or the system-wide root password, proceed to the following step:

  Procedure 2-1, step 4 on page 44

4. Type the following command to start the cluster configuration tool:

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

5. On the cluster configuration tool's main menu select **Reset the Database** and select **OK**.

   Follow the instructions on the screen to reset the database.

   For more information about the cluster configuration tool, see "Running the Cluster Configuration Tool" on page 61.

6. After the database is reset, back on the main menu, click **Initial Setup Menu**.

   Within the **Initial Setup Menu**, complete the options in the order that they appear.

Note: If the system issues a message about existing system images, proceed as follows:

- If you changed the root password or the time zone, do not choose to use the existing system images. The previous root password and time zone are stored in the existing images. Recreating the system images takes about 30 minutes.
- If you did not change the root password or the time zone, you can choose to use the existing system images.

---

7. Change the subdomain name.

Type the following command:

```
# cadmin --set-subdomain
```

For more information about the `cadmin` command, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195.

8. Run the `discover` command to configure the rack leader controllers (RLCs) and service nodes.

   For information about the `discover` command, see "`discover` Command" on page 84.

Installing and Configuring an SGI ICE X System

SGI installs operating system software on each ICE X system before factory shipment occurs. The topics in this chapter include the additional procedures that you need to complete in order to configure the system for your site.

If you want to completely reinstall the operating system and all other software, the topics in this chapter enable you to complete that task. For example, you might need to reinstall the operating system to meet site requirements or to recover a system in case of a disaster.

This chapter includes the following topics:

- .

Note: If you are upgrading from a prior release or installing SMC for SGI ICE X software patches, see "Installing SMC for SGI ICE Patches and Updating SGI ICE Systems " on page 122.

Performing a New Installation and Configuring the Software on an SGI ICE X System

Table 3-1 on page 47 shows the installation and configuration process for a situation in which you want to install the SGI ICE X system from scratch. In this case, you reinstall the operating system on the nodes and configure everything yourself.

Table 3-1 SGI ICE X System Installation and Configuration Process

Step	Task	See
1	Prepare to install the SGI ICE X software	"Preparing to Install Software on an SGI ICE X System" on page 49
2	Configure a static address for the baseboard management controller (BMC) on the system admin controller (SAC). If you plan to configure a highly available system admin controller (SAC), configure the BMC on each of the two SACs.	"Setting a Static IP Address for the Baseboard Management Controller (BMC) in the System Admin Node (SAC)" on page 51
3	(Optional) Configure a highly available system admin controller	"(Optional) Configuring a Highly Available (HA) System Admin Controller (SAC)" on page 52
4	Boot the system	"Booting the System" on page 52
5	Install the operating system on the system admin controller (SAC) node	"Installing the Operating System" on page 54
6	Run the cluster configuration tool. Complete the initial cluster set-up tasks, which include the following: • Set up software repositories for required and optional software • Install the SAC software • Configure network settings • Configure the NTP server • Set up the initial SAC infrastructure • Configure the house network DNS resolvers	"Running the Cluster Configuration Tool" on page 61
7	(Conditional) Customize the cluster configuration	"(Conditional) Customizing the Cluster Configuration" on page 68
8	Install the SGI Management Center (SMC) license key	"Installing the SGI Management Center License Key" on page 78
9	Sync the repository updates, apply the latest patches to the newly installed software, and clone the images	"Synchronizing the Software Repository, Installing Software Updates, and Cloning the Images" on page 78
10	Configure the switches	"Configuring the Switches" on page 79

Step	Task	See
11	Use the `discover` command to install and configure the rack leader controller and service node software	"`discover` Command" on page 84
12	Run the cluster configuration tool to configure the following: • (Optional) The backup DNS server • The InfiniBand network	.
13	(Optional) Configure the backup DNS server	.
14	Configure the InfiniBand network	.

Preparing to Install Software on an SGI ICE X System

The following procedure lists pre-installation tasks that you need to complete before you begin working with the SGI ICE X system.

Procedure 3-1 To prepare for an installation

1. Contact your site's network administrator, and obtain network information.

   Obtain the following information to use when you configure the baseboard management controller (BMC):

   - (Optional) The current IP address of the BMC on the system admin node (SAC). You can set the BMC address from a serial console if you do not have this information.
   - The address you want to set for the BMC.
   - The netmask you want to set for the BMC.
   - The default gateway you want to set for the BMC.

   Your network administrator can provide an IP address, a hostname, or a fully qualified domain name (FQDN) for each of the preceding addresses.

   Obtain the following information to use when you configure the network for the SGI ICE X system:

   - Hostname
   - Domain name

- IP address
- Netmask
- Default route
- Root password

Obtain the following information about your site's house network:

- IP addresses of the domain name servers (DNSs)

2. Familiarize yourself with the boot parameters, and determine which boot parameters you want to use.

   You can configure your SGI ICE X system to boot from one, two (default), three, four, or five partitions. This enables you to configure your SGI ICE X system as either a single-boot computer system or as a multiple-boot computer system. A multiple-boot computer system has two or more partitions, so it has more than one root directory (`/`) and more than one boot directory (`/boot`). In an SGI ICE X system, these root and boot directories are paired into multiple *slots*. A multiple-slot disk layout is also called a *cascading dual-root layout* or a *cascading dual-boot layout*.

   The installation procedure explains how to create a default, two-slot SGI ICE X system and directs you to use the `install` boot parameter. If you want to create only one slot, or if you want to create three or more slots, you need to specify different boot parameters.

   The installer creates the same disk layout on all compute nodes. For more information about boot parameters, disk layouts, and so on, see "Boot Parameters, Disk Partitioning, and Managing a Multiboot System" on page 139.

3. Obtain the following from your SGI representative:

   - The MAC file for your system. The MAC file contains MAC address information for the nodes. If you have these addresses, the node discovery process can complete more quickly.

Setting a Static IP Address for the Baseboard Management Controller (BMC) in the System Admin Node (SAC)

When you set the IP address for the BC on the SAC, you ensure access to the SAC when the site DHCP server is inaccessible. If you want to configure a highly available SAC, make sure to perform this topic's procedure on the BMCs on each of the two SACs.

The following procedure explains how to set a static IP address.

Procedure 3-2 To set a static IP address for the BMC on the SAC

1. Connect to the console on the SAC.

   You can make this connection in one of the following ways:

   - Use the terminal attached to the SAC.
   - Attach a keyboard, monitor, and mouse to the baseboard management controller (BMC) on the SAC.
   - Use a PC or workstation to connect to the BMC on the SAC over the network, and log in through the IPMI tool. This method assumes that you know the IP address of the BMC. Complete the following steps:

     1. Type the following command to obtain a console:

        `ipmitool ---H` *address* `---I lanplus ---U ADMIN ---P ADMIN sol activate`

        For *address*, type the IP address, hostname, or FQDN of the BMC on the SAC.

     2. Type the following command to ensure that the IPMI tool is enabled whenever you reboot the SAC:

        `chkconfig ipmi on`

     3. Type the following command to start the IPMI service:

        `service ipmi start`

     4. Type the following command to configure the network on the BMC:

        `ipmitool lan set ipaddr` *IP_addr* `netmask` *netmask_addr* `defgw` *gateway_addr*

For each of the addresses, specify the addresses you obtained from the network administrator in "Preparing to Install Software on an SGI ICE X System" on page 49.

2. Proceed to one of the following:

   - If you want to configure a highly available SAC, proceed to the following:

     "(Optional) Configuring a Highly Available (HA) System Admin Controller (SAC)" on page 52

   - If you want to configure a single SAC, proceed to the following:

     "Booting the System" on page 52

(Optional) Configuring a Highly Available (HA) System Admin Controller (SAC)

SGI enables you to configure the SAC and your rack leader controllers (RLCs) as highly available nodes in an SGI ICE X system. If you want to enable high availability, use the information in the following appendix:

Appendix B, "Installing a Highly Available System Admin Controller (SAC) or Rack Leader Controller (RLC)" on page 335

Booting the System

The following procedure explains how to boot the system and begin the installation.

Procedure 3-3 To boot the system

1. (Conditional) Power-off the system admin controller (SAC).

   Perform this step only if the SAC is powered on at this time.

2. Power-on the SAC.

   As Figure 3-1 on page 53 shows, the power-on button is on the right of the SAC.

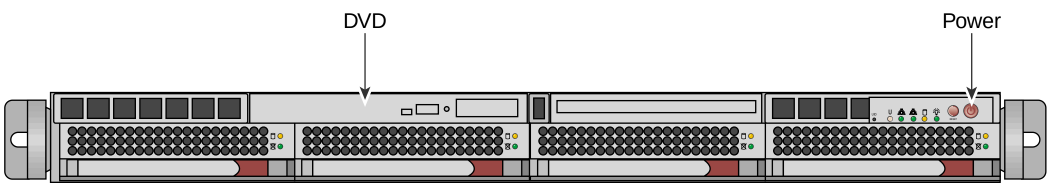

Figure 3-1 SAC Power On Button and DVD Drive

3. (Optional) Back up the cluster configuration snapshot in the `/opt/sgi/var/ivt` directory on the SAC.

   If you back up your system's factory configuration now, you can use the interconnect verification tool (IVT) to verify your hardware configuration later.

   You can use `ftp` or `scp` to copy the IVT files to another server at your site, or you can write the IVT files to a USB stick. For more information about IVT, see "Inventory Verification Tool" on page 274.

4. (Optional) Configure the system so that you can perform the installation from a VGA screen and can perform later operations from a serial console.

   If you want to enable this capability, perform the following steps:

   - Use a text editor to open file `/boot/grub/menu.lst`.
   - Search the file for the word `kernel` at the beginning of a line.
   - Add the following to the `kernel` line: `console=`*type*.

     For example:

```
kernel /boot/vmlinuz-2.6.16.46-0.12-smp root=/dev/disk/by-label/sgiroot console=ttyS1,38400n8
splash=silent showopts
```

   - Add the `console=`*type* parameter to the end of every `kernel` line.

   Later, if you want to access the SAC from only a VGA, you can remove the `console=` parameters.

5. Insert the SGI Admin Node Autoinstallation DVD into the DVD drive on the system admin controller (SAC).

   The autoinstallation message appear, and at the end is the `boot:` prompt.

6. At the `boot:` prompt, type **`install`** and, optionally, other boot parameters.

"Boot Parameters, Disk Partitioning, and Managing a Multiboot System" on page 139 explains the other, optional, boot parameters.

Monitor the installation. This can take several minutes.

7. Remove the operating system installation DVD.

8. At the # prompt, type **`reboot`**.

   This is the first boot from the SAC's hard disk.

9. Proceed to one of the following:

   - If you want to install SUSE Linux Enterprise Server (SLES), proceed to "Installing SUSE Linux Enterprise Server (SLES)" on page 54.
   - If you want to install Red Hat Enterprise Linux (RHEL), proceed to "Installing Red Hat Enterprise Linux (RHEL)" on page 58.

Installing the Operating System

The SGI ICE X platform supports both the SUSE Linux Enterprise Server (SLES) and Red Hat Enterprise Linux (RHEL) operating systems. Use one of the following procedures to install your operating system software on the system admin controller (SAC) node:

- "Installing SUSE Linux Enterprise Server (SLES)" on page 54
- "Installing Red Hat Enterprise Linux (RHEL)" on page 58

Installing SUSE Linux Enterprise Server (SLES)

The SLES YaST2 interface enables you to install the SLES operating system on the SGI ICE X system. To navigate the YaST2 modules, use key combinations such as `Tab` (forward) `Shift + Tab` (backward). You can use the arrow keys to move up, down, left, and right. To use shortcuts, press the `Alt` + the highlighted letter. Press `Enter` to complete or confirm an action. `Ctrl + L` refreshes the screen. For more information about navigation, see Appendix C, "YaST2 Navigation" on page 377.

The following procedure explains how to use YaST2 to install SLES 11 SP2 on an SGI ICE X system. Use the following keys to navigate the YaST2 interface:

Procedure 3-4 To install SLES 11 SP2 on an SGI ICE X system

1. On the **Language and Keyboard Layout** screen, complete the following steps:
   - Select your language
   - Select your keyboard layout
   - Select **Next**.
2. On the **Welcome** screen, select **Next**.
3. On the **Hostname and Domain Name** screen, complete the following steps:
   - Type the hostname for this SGI ICE X system.
   - Type the domain name.
   - Clear the box next to **Change Hostname via DHCP**. The box appears with an X in it by default, but you need to clear this box.
   - Select **Assign Hostname to Loopback IP**. Put an X in this box.
   - Select **Next**.
4. On the **Network Configuration** screen, complete the following steps:
   - Select **Change**. A pop-up window appears.
   - On the pop-up window, choose **Network Interfaces.**
5. On the **Network Settings** screen, complete the following steps:
   - Highlight the first network interface card that appears underneath **Name**.
   - Select **Edit**.
6. On the **Network Card Setup** screen, specify the system admin controller's (SAC's) house/public network interface.

   Figure 3-2 on page 56 shows the **Network Card Setup** screen.

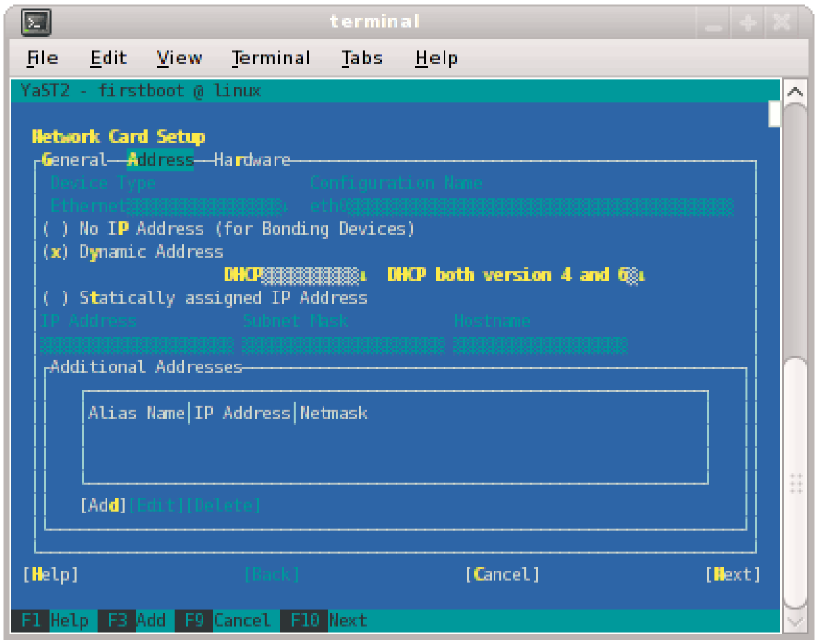

Figure 3-2 Network Card Setup Screen

Complete the following steps:

- Select **Statically Assigned IP Address**. SGI recommends a static IP address, not DHCP, for system admin nodes (SACs).
- In the **IP Address** field, type the system's IP address.
- In the **Subnet Mask** field, type the system's subnet mask.
- In the **Hostname** field, type the system's fully qualified domain name (FQDN). SGI requires you to type an FQDN, not the system's shorter hostname, into this field. For example, type `mysystem-admin.mydomainname.com`. Failure to supply an FQDN in this field causes the `configure-cluster` command to fail.
- Select **Next**.

You can specify the default route, if needed, in a later step.

7. On the **Network Settings** screen, complete the following steps:
   - Select **Hostname/DNS**.
   - In the **Hostname** field, type the system's fully qualified domain name (FQDN).
   - In the **Domain Name** field, type the domain name for your site.
   - Put an X in the box next to **Assign Hostname to Loopback IP**.
   - In the **Name Servers and Domain Search List**, type the name servers for your house network.
   - Back at the top of the screen, select **Routing**.

     The **Network Settings > Routing** screen appears.
   - In the **Default Gateway** field, type your site's default gateway.
   - Select **OK**.
8. On the **Network Configuration** screen, click **Next**.

   The **Saving Network Configuration** screen appears and saves your configuration.
9. On the **Clock and Time Zone** screen, complete the following steps:
   - Select your region.
   - Select your time zone.
   - (Optional) In the **Hardware Clock Set To** field, choose **Local Time** or accept the default of **UTC**.
   - Select **Next**.

   This step synchronizes the time in the BIOS hardware with the time in the operating system. Your choice depends on how the BIOS hardware clock is set. If the clock is set to GMT, which corresponds to UTC, your system can rely on the operating system to switch from standard time to daylight savings time and back automatically.
10. On the **Password for System Administrator "root"** screen, complete the following steps:

- In the **Password for root User** field, type the password you want to use for the root user for all compute nodes (SAC, rack leader controller (RLC), and service nodes) throughout the SGI ICE X system.
- In the **Confirm password** field, type the root user's password again.
- In the **Test Keyboard Layout** field, type a few characters.

  For example, if you specified a language other than English, type a few characters that are unique to that language. If these characters appear in this plain text field, you can use these characters in passwords safely.
- Select **Next**.

11. On the **User Authentication Method** screen, select one of the authentication methods and select **Next**.

    Typically, users accept the default (**Local**).

12. On the **New Local User** screen, create additional user accounts or select **Next**.

    If you do not create additional users, select **Yes** on the **Empty User Login** warning pop-up window, and select **Next**.

13. On the **Installation Completed** screen, select **Finish**.

14. Log into the SAC and confirm that the system is working as expected.

    If necessary, restart YaST2 to correct settings.

Installing Red Hat Enterprise Linux (RHEL)

.

Initial Configuration of a RHEL 6 System Admin Controller (SAC)

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

Procedure 3-5 Initial Configuration of a RHEL 6 SAC

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

1. Add the `IPADDR`, `NETMASK`, and `NETWORK` values appropriate for the public (house) network interface to the

/etc/sysconfig/network-scripts/ifcfg-eth0 file similar to the following example:

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

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

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

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

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

4. Set the SAC hostname, as follows:

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

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

```
search mydomain.com
nameserver 192.168.0.1
nameserver 192.168.0.25
```

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

```
# nscd -i hosts
```

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

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

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

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

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

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

   Confirm the `timezone`, as follows:

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

9. (Conditional) Edit file `/etc/ntp.conf` to direct requests to the NTP server at your site.

   Complete the following steps if you use the RHEL operating system and you want to direct requests to your site's NTP server instead of to the public time servers of the `pool.ntp.org` project:

   - Use a text editor to open file `/etc/ntp.conf`.
   - Insert a pound character (#) into column 1 of each of each line that includes `rhel.pool.ntp.org`.

     Note: Do not edit or remove entries that serve the cluster networks.

     The following is an example of a correctly edited file:

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

- Type the following command to restart the NTP server:

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

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

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

11. Run the `configure-cluster` command. See "Running the Cluster Configuration Tool" on page 61.

Running the Cluster Configuration Tool

The cluster configuration tool enables you to configure, or reconfigure, your SGI ICE X system. The procedure in this topic explains the general, required configuration steps for all SGI ICE X systems. If your SGI ICE X system includes optional components, or if your site has specific requirements that require further customization, later procedures explain how to use the cluster configuration tool to create a more customized environment.

The following are the required cluster configuration steps:

- Create repositories for software installation files and updates.
- Install the system admin node (SAC) cluster software.
- Configure the cluster subdomain and examine other network settings. The cluster subdomain is likely to be different from the `eth0` domain on the SAC itself.
- Configure the NTP server.
- Install the cluster's software infrastructure. This step can take 30 minutes.
- Configure the house network's DNS resolvers.

The following procedure explains how to perform the required configuration tasks.

Procedure 3-6 To run the cluster configuration tool

1. Locate your site's SGI software distribution DVDs or verify the path to your site's online software repository.

   You can install the software from either physical media or from an ISO on your network.

2. From the VGA screen, or through an `ssh` connection, log into the system admin controller (SAC) as the root user.

   SGI recommends that you run the cluster configuration tool either from the VGA screen or from an `ssh` session to the system admin controller (SAC). Avoid running the `configure-cluster` command from a serial console.

3. Type the following command to start the cluster configuration tool:

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

4. On the cluster configuration tool's **Initial Configuration Check** screen, select **OK** on the initial pop-up window.

   Figure 3-3 on page 62 shows the initial screen and the pop-up.

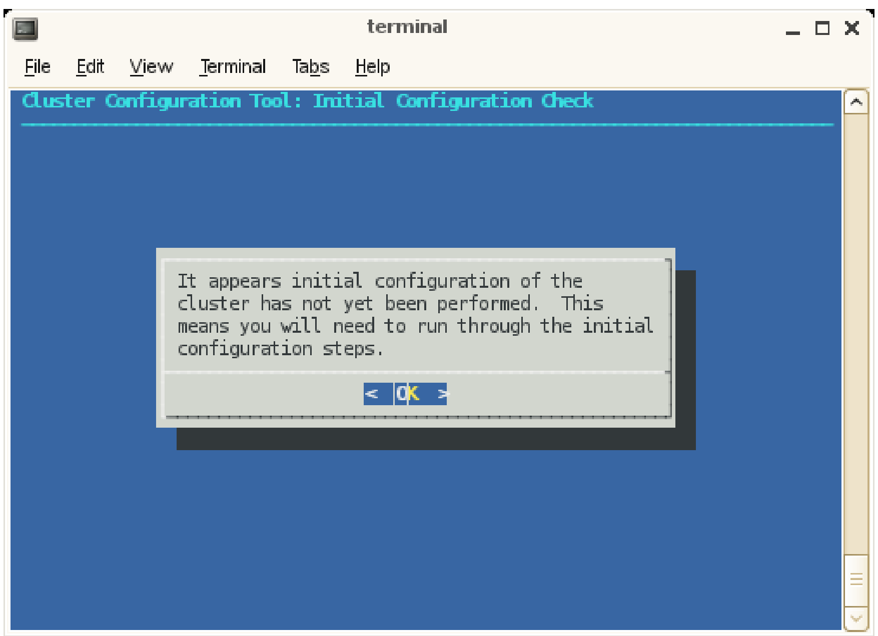

Figure 3-3 Initial Configuration Check Screen

The cluster configuration tool recognizes a configured cluster. If you start the tool on a configured SGI ICE X system, it opens into the **Main Menu**.

5. On the **Initial Cluster Setup** screen, select **OK** on the initial pop-up window.

   Figure 3-4 on page 63 shows the initial screen and the pop-up.

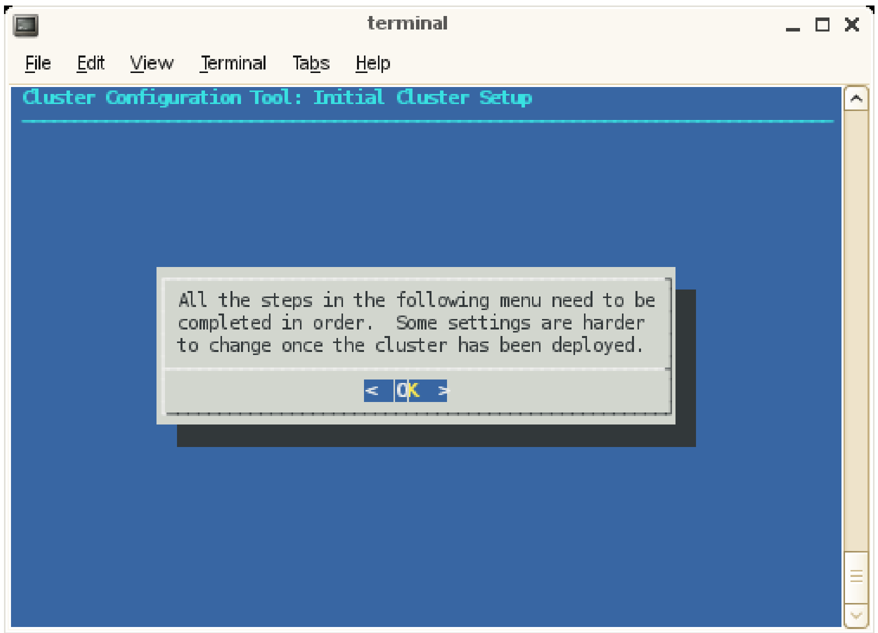

Figure 3-4 Initial Cluster Setup Screen with the initial pop-up window

6. On the **Initial Cluster Setup** screen, select **R Repo Manager: Set Up Software Repos**, and click **OK**.

   Figure 3-5 on page 64 shows the **Initial Cluster Setup** screen with the task menu. This procedure guides you through the tasks you need to perform for each of the menu selections on the **Initial Cluster Setup** screen.

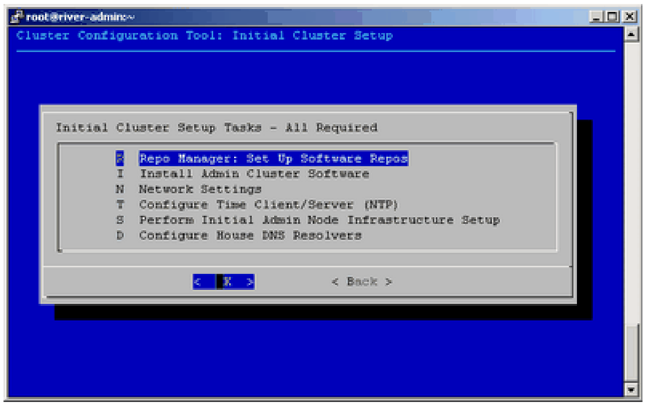

Figure 3-5 **Initial Cluster Setup Tasks** Screen

The next few steps create software repositories for the initial installation packages and for updates. You need to create repositories for the following software:

- The operating system software, either RHEL or SLES
- SGI Foundation Suite
- SGI Management Center (SMC) for SGI ICE X
- Additional software packages for which you hold licenses, such as the Message Passing Toolkit, the SGI Performance Suite, and any others

7. On the **Repo Manager** screen, respond to the pop-up windows as follows:
   - On the **One or more ISOs were ...** pop-up window, select **Yes**.
   - On the **Repositories are created ...** pop-up window, press **Enter**.
   - On the **You will now be prompted ...** pop-up window, select **OK**.
   - On the **Would you like to register ...** pop-up window, select **Yes**.
   - On the last pop-up window, specify whether you want to install from DVDs or from an ISO image.
8. Perform one of the following step sequences to create the software repositories:

- To create the repositories from DVDs, complete the following sequence of steps:

  1. Select **Insert DVD**.
  2. Insert a DVD.
  3. Select **Mount inserted DVD**.
  4. On the **Media registered successfully with crepo ...** screen, select **OK**, and eject the DVD.
  5. On the **Would you like to register ...** pop-up window, select **Yes** if you have more software that you need to register.

     If you select **Yes**, repeat the preceding this sequence for the next DVD.

     If you select **No**, proceed to the next step.

- To create the repositories from ISO images, complete the following sequence of steps:

  1. Select **Use custom path/URL**.
  2. On the **Please enter the full path to the mount point or the ISO file ...** screen, type the full path in *server_name*:*path_name*/*iso_file* format. This field also accepts a URL or an NFS path. Select **OK** after typing the path.
  3. On the **Media registered successfully with crepo ...** screen, select **OK**.
  4. On the **Would you like to register ...** pop-up window, select **Yes** if you have more software that you need to register.

     If you select **Yes**, repeat the preceding tasks in this sequence for the next DVD.

     If you select **No**, proceed to the next step.

9. On the **Initial Cluster Setup** screen, select **I Install Admin Cluster Software**, and select **OK**.

   This step installs the cluster software that you wrote to the repositories.

10. On the **Initial Cluster Setup** screen, select **N Network Settings**, and select **OK**.

11. On the **Cluster Network Settings** screen, select **C Configure Subnet Addresses**, and select **OK**.

12. On the **Warning: Changing the subnet IP addresses ...** screen, click **OK**.
13. Review the settings on the **Subnet Network Addresses** screen, and modify these settings only if absolutely necessary.

    Select either **OK** or **Back** if you accept the defaults.

    If your site has network requirements that conflict with the defaults, you need to change the network settings. On the **Update Subnet Addresses** screen, the **Head Network** field shows the SAC's IP address. SGI recommends that you do not change the IP address of the SAC or rack leader controllers (RLCs) if at all possible. You can change the IP addresses of the InfiniBand network (**IB0** and **IB1**) to match the IP requirements of the house network, and then select **OK**.
14. On the **Cluster Network Settings** screen, select **D Configure Cluster Domain Name**, and select **OK**.
15. On the **Please enter the domain name for this cluster.** pop-up window, type the domain name, and select **OK**.

    The domain you type becomes a subdomain to your house network..

    For example, type `ice.americas.sgi.com`.
16. On the **Cluster Network Settings** screen, select **Back**.
17. On the **Initial Cluster Setup** screen, select **T Configure Time Client/Server (NTP)**, and select **OK**.
18. On the pop-up window that begins with **This procedure will replace your ntp configuration file ...**, select **Yes**.
19. On the pop-up window that begins with **A new ntp.conf has been put in to position ...**, select **OK**.

    On the subsequent screens, you set the SAC as the time server to the SGI ICE X system. The cluster configuration tool screens differ for the RHEL and SLES operating systems. On RHEL platforms, you return to the **Initial Cluster Setup** menu. On SLES platforms, use the SLES documentation to guide you through the NTP configuration screens.
20. On the **Initial Cluster Setup** menu, select **S Perform Initial Admin Node Infrastructure Setup**, and select **OK**.
21. On the pop-up window that begins with **A script will now perform the initial cluster ...**, select **OK**.

22. On the **Admin Infrastructure One Time Setup** screen, in the **Initial Cluster Setup Complete** pop-up, select **OK**.

    This step runs a series of scripts that configure the SAC on the SGI ICE X system. The scripts also create the root images for the RLC, service, and compute nodes. The scripts run for approximately 30 minutes. At the end, the script issues a line that includes **install-cluster completed** in its output. This step takes about 30 minutes.

    The final output of the script is as follows:

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

    The output of the `mksiimage` commands are stored in a log file at the following location:

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

23. On the **Initial Cluster Setup** menu, select **D Configure House DNS Resolvers**, and select **OK**.

    Figure 3-6 on page 67 shows the **Configure House DNS Resolvers** screen.

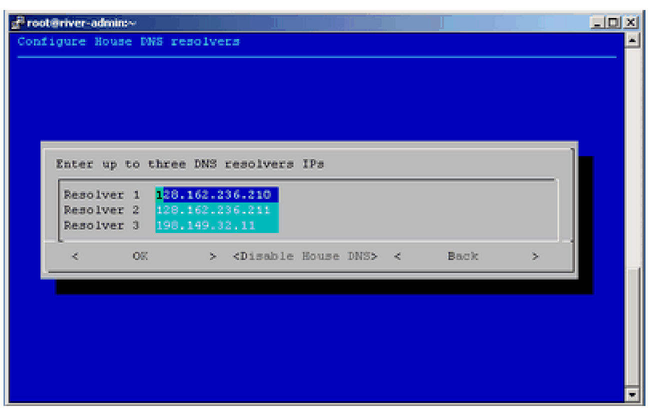

Figure 3-6 Configure House DNS Resolvers Screen

The system autopopulates the values on the **Configure House DNS Resolvers** screen to match the DNS specifications on the SAC. The DNS resolvers you specify here enable the service nodes to resolve host names on your network. You can set the DNS resolvers to the same name servers used on the SAC itself.

Complete the following steps:

1. Perform one of the following actions:
   - To accept these settings, select **OK**.
   - To change the settings, type in different IP addresses, and select **OK**.
   - To disable house network resolvers, select **Disable House DNS**.
2. On the **Setting DNS Forwarders to ...** pop-up window, select **Yes**.
3. On the **Initial Cluster Setup complete** pop-up window, select **OK**.
4. On the **Initial Cluster Setup** screen, select **Back**.
5. Proceed to "(Conditional) Customizing the Cluster Configuration" on page 68.

(Conditional) Customizing the Cluster Configuration

This topic explains how to use the cluster configuration tool to enable features for specific situations. The features you need to enable depend on your hardware platform's features and your site requirements. When you use the cluster configuration tool, you set global values that apply to all nodes that you discover after you set the value. When you perform an initial installation, none of the nodes have been discovered at this point in the configuration procedure, so all nodes you discover later are set the same. However, if you use the cluster configuration tool to change values on a system that is already configured, you might need to reset values on older, existing nodes that you configured previously. You can use commands to reset the values on previously configured nodes.

Table 3-2 on page 68 shows the cluster configuration tool **Main Menu** options and explains those that you need to configure.

Table 3-2 Cluster Configuration Tool Main Menu Options

Menu Selection	Platform Notes	For More Information
B Configure Backup DNS Server	Optional on all platforms. This menu option enables you to configure one of your service nodes as a DNS server. If the DNS on your system admin controller (SAC) is unavailable, the service node you configure here can act as a DNS server for your system.	tbd
M Configure Redundant Management Network	Enable on all SGI ICE X platforms. On SGI ICE X and SGI ICE 8400 systems, the default is **yes** (enabled). On SGI ICE 8200 systems, the default is **yes** (enabled), but you need to set this to **no** (disabled). Enables a secondary network from the SAC, RLC, and service nodes to the cluster network.	tbd
S Configure Switch Management Network	Default is **no** (disabled). Set to **yes** (enabled) on all SGI ICE X platforms. Accept the default (**no**) on SGI ICE 8400 and SGI ICE 8200 platforms. This selection enables or disables link aggregation within the cluster. The Ethernet switch controls all VLANs and trunking within the cluster.	tbd
N Configure MCell Network	Enable on all SGI ICE X platforms that are equipped with MCells. Enables the separate network that the MCells require.	tbd
Q Configure MySql Replication	Default is **yes** (enabled). Enable on very large systems. When enabled, the cluster's MySQL database keeps the leader and server nodes synchronized.	tbd
U Configure Default Max Rack IRU Setting	Verify the setting on all platforms. When set appropriately, it minimizes the time it takes to distribute software to the blades.	tbd
C Configure blademond rescan interval	Optional on all platforms. Enables automatic blade discovery.	tbd

The following procedure explains how to use the cluster configuration tool to configure additional features on your SGI ICE system.

Procedure 3-7 To configure system-wide values for specific configurations

1. Log into the system admin controller (SAC) as the root user, and type the following command to start the cluster configuration tool:

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

2. Choose menu selections from the **Cluster Configuration Tool: Main Menu** that you want to configure for your system.

   Table 3-2 on page 68 summarizes the menu items, and the following topics provide additional information:

   - "Configuring a Backup Domain Name System (DNS) Server" on page 70
   - "Configuring a Redundant Management Network (RMN)" on page 72
   - "Configuring a Switch Management Network" on page 73
   - "Configuring an MCell Network" on page 74
   - "Configuring MySQL Database Server Replication" on page 74
   - "Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting" on page 76
   - "Configuring the `blademond` Rescan Interval" on page 77

3. Proceed to the following:

   "Installing the SGI Management Center License Key" on page 78

Configuring a Backup Domain Name System (DNS) Server

When you configure a backup DNS, the compute nodes can use a service node as a secondary DNS server if the rack leader controller (RLC) is not available.

The following procedure explains how to configure a service nodes to act as a DNS when the RLC is down or being serviced.

Procedure 3-8 To enable a backup DNS

1. (Conditional) Log into the system admin controller (SAC) as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. (Optional) Type the following command to retrieve a list of available service nodes:

   ```
   # cnodes --service
   ```

3. On the **Main Menu** screen, select **B Configure Backup DNS Server (optional)**, and select **OK**.

4. On the pop-up window that appears, type the identifier for the service node that you want to designate as the backup DNS, and select **OK**.

   Figure 3-7 on page 71 shows how to specify `service0` as the backup DNS.

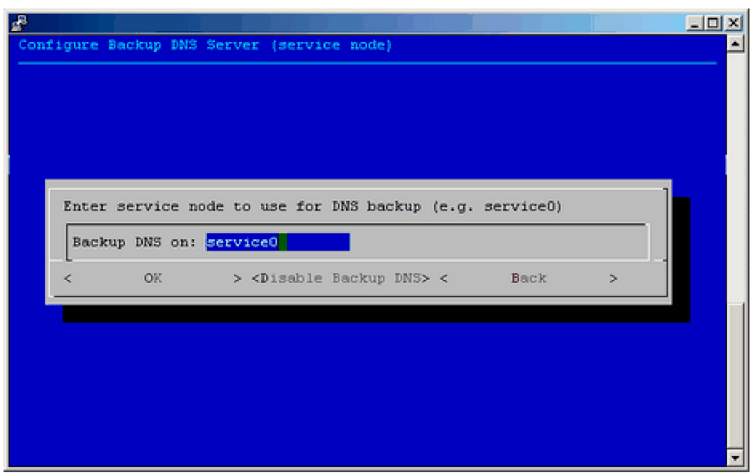

Figure 3-7 Configure Backup DNS Server (service node) pop-up window

To disable this feature, select **Disable Backup DNS** from the same menu and select **Yes** to confirm your choice. For information about how to use commands to enable or disable this feature, see the following:

"Enabling or Disabling the Backup Domain Name Service (DNS) on an SGI ICE X Cluster" on page 138

Configuring a Redundant Management Network (RMN)

An RMN is a secondary network from the nodes to the cluster network. The RMN is enabled by default for all platforms. On SGI ICE X systems and SGI ICE 8400 systems, make sure the RMN is enabled. On SGI ICE 8200 systems, make sure to disable the RMN.

When an RMN is enabled for a node, the default Linux bonding mode for RLCs and service nodes is 802.3ad link aggregation. The RMN has the following additional characteristics:

- The GigE switches are doubled in the system control network and stacked (using stacking cables).
- The links from the chassis management controllers (CMCs) are doubled.
- Some links from the system admin controller (SAC), rack leader controllers (RLCs), and most service nodes are doubled.
- Baseboard management controller (BMC) connections are not doubled, which means that certain failures can cause temporary inaccessibility to the BMCs. During these failures, the host interfaces remain accessible.

When you use the cluster configuration tool to configure an RMN, the system enables an RMN for all nodes that you discover after you enable the setting. If you have existing nodes in the cluster without an RMN, those existing nodes are not changed. The following procedure explains how to configure an RMN from the cluster configuration tool.

Procedure 3-9 To enable the RMN from the cluster configuration tool

1. (Conditional) Log into the system admin controller (SAC) as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **M Configure Redundant Management Network (optional)**, and select **OK**.

3. On the pop-up window that appears, select **Y yes** (default), and select **OK**.

For information about how to use commands to enable or disable the RMN, see the following:

"Enabling or Disabling the Redundant Management Network (RMN)" on page 137

For a diagram that shows an RMN, see Figure 1-11 on page 38. For information about link aggregation, see "Link Aggregation, Rack Leader Controllers (RLCs), and Service Nodes" on page 101.

Configuring a Switch Management Network

On an SGI ICE X system, enable the switch management network. On SGI ICE 8400 and SGI ICE 8200 systems, disable the switch management network. If your cluster mixes SGI ICE X racks with either SGI ICE 8400 or SGI ICE 8200 racks, use the cluster configuration tool to enable the switch management network in the cluster.

The system software attempts to set this value automatically, but you can use the procedures in this topic to verify or reset the value. In an SGI ICE X cluster, the switch management network enables the Ethernet switch to control all VLANs and trunking.

The following procedure explains how to enable the switch management network.

Procedure 3-10 To enable the switch management network from the cluster configuration tool

1. (Conditional) Log into the system admin controller (SAC) as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **S Configure Switch Management Network (optional)**, and select **OK**.

3. On the pop-up window that appears, select **Y yes**, and select **OK**.

   Figure 3-8 on page 74 shows the selection pop-up window:

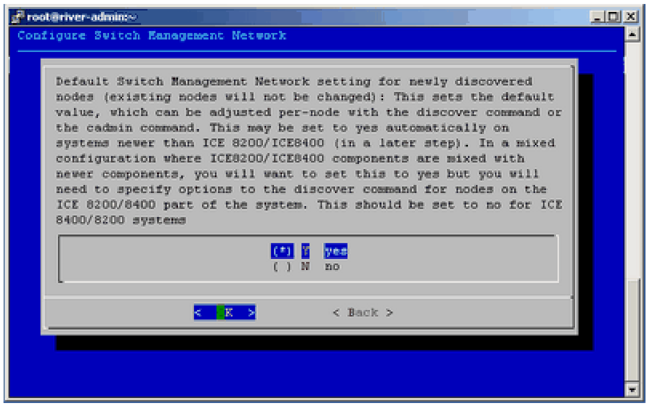

Figure 3-8 **Configure Switch Management Network** pop-up window

Configuring an MCell Network

Perform the procedure in this topic if your SGI ICE X system includes MCells.

The MCell network is the internal network that powers the MCell cooling system. The following procedure explains how to enable the MCell network.

Procedure 3-11 To enable MCells from the cluster configuration tool

1. (Conditional) Log into the system admin controller (SAC) as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **N Configure MCell Network (optional)**, and select **OK**.

3. On the pop-up window that appears, select **Y yes**, and select **OK**.

Configuring MySQL Database Server Replication

SGI recommends that you enable MySQL replication on very large systems to keep the internal cluster database synchronized. The master MySQL database server resides on the system admin controller (SAC). When you enable replication, data from the master MySQL database server to be replicated to one or more MySQL database slaves on the rack leader controller (RLCs) and service nodes). If your site has a large number of racks, using this feature can reduce the amount of contention for database resources on the SAC.

If the database becomes corrupt, you can disable replication during the debugging session and reenable it later.

The following procedure explains how to enable MySQL database replication.

Procedure 3-12 To enable MySQL database replication from the cluster configuration tool

1. (Conditional) Log into the SAC as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **Q Configure MySQL Replication (optional)**, and select **OK**.

3. On the pop-up window that appears, select **Y yes**, and select **OK**.

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

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

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

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

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

```
my_sql_replication      : yes
```

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

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

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

Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting

You can configure the maximum number of blade enclosures that an individual rack leader controller (RLC) can manage. When you set this to a value that is appropriate to your system size, it takes less time to distribute new software images to the blades in an enclosure. If you change this value, the system assigns the new value to any nodes that you discover.

Procedure 3-13 To configure the default maximum IRU setting from the cluster configuration tool

1. (Conditional) Log into the SAC as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **U Configure Default Max Rack IRU Setting (optional)**, and select **OK**.

3. On the pop-up window that appears, type `8` or `4`, and select **OK**.

   On SGI ICE X platforms, set to this value to 8. On SGI ICE 8200 or SGI ICE 8400 platforms, set this value to 4.

4. (Conditional) Use the cadmin command to change the maximum number of IRUs managed by existing, configured RLCs.

   Perform this step if your system includes configured RLCs that have a different number of IRUs configured as their maximum.

Type the following command to retrieve the current setting:

```
# cadmin --show-max-rack-irus
```

Configuring the `blademond` Rescan Interval

When enabled, the system checks every two minutes for changes to the number of blades in the system. If you remove or add a new blade, the system automatically detects this change, updates the system, and integrates the change on the rack. By default, the interval between checks is set to `120`, which is two minutes.

Procedure 3-14 To configure the `blademond` rescan interval from the cluster configuration tool

1. (Conditional) Log into the SAC as the root user, and type the following command to start the cluster configuration tool:

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

   Perform this step only if the cluster configuration tool is not running at this time.

2. On the **Main Menu** screen, select **C Configure blademond rescan interval (optional)**, and select **OK**.

3. On the pop-up window that appears, accept the default of `120`, which is two minutes, and select **OK**.

   Alternatively, type a different value and select **OK**.

4. On the **Main Menu** screen, select **U Configure Default Max Rack IRU Setting (optional)**, and select **OK**.

5. Visually inspect the pop-up window that appears and verify that the maximum IRU setting is appropriate for your system.

   On SGI ICE X platforms, set to this value to 8. On SGI ICE 8200 or SGI ICE 8400 platforms, set this value to 4.

   When the maximum IRU setting is configured correctly, the system manages the changes to your system more efficiently.

   For more information about this setting, see "Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting" on page 76.

Installing the SGI Management Center License Key

The SGI Management Center (SMC) software runs on the system admin controller (SAC). SMC provides a graphical user interface for system configuration, operation, and monitoring.

For more information about using SMC, see *SGI Management Center (SMC) System Administrator Guide*.

For more information about licensing, see the licensing FAQ on the following website:

http://www.sgi.com/support/licensing/faq.html

The following procedure explains how to obtain and install the license key for SMC.

Procedure 3-15 To license the SMC software

1. Use a text editor to open file `/etc/lk/keys.dat`.
2. Copy and paste the license key string exactly as it was given to you.
3. Save the file.
4. Type the following command to restart the SMC daemon:

   ```
   # service mgr restart
   ```

5. Proceed to the following:

   "Synchronizing the Software Repository, Installing Software Updates, and Cloning the Images" on page 78

Synchronizing the Software Repository, Installing Software Updates, and Cloning the Images

The following procedure explains how to update the software in the repositories that you created with the cluster configuration tool.

Procedure 3-16 To update the software

1. Log into the system admin controller (SAC), and type the following command to retrieve information about the network interface card (NIC) bonding method on the SAC:

   ```
   # cadmin --show-mgmt-bonding --node admin
   ```

The command returns `802.3ad` if bonding is set appropriately..

If the command does not return `802.3ad`, type the following commands to set the bonding appropriately, and reboot the system:

```
# cadmin --set-mgmt-bonding --node admin 802.3ad
# reboot
```

`discover_skip_switchconfig=YES`. This setting is `YES` if manufacturing configured the switch. If you don't know the status, set this to `NO`.

`mgmt_bonding=802.3ad`. On SGI ICE X SAC and service nodes, when set to `active-backup`, the system uses the first NIC card unless the first NIC card is unavailable. The system uses the second NIC until the first NIC can be used again.

2. Type the following command to retrieve the new images:

   ```
   # sync-repo-updates
   ```

3. Type the following command to install the images:

   ```
   # cinstallman --update-image
   ```

4. Type the following command to create the new software images on the system:

   ```
   # create-default-sgi-images
   ```

5. Proceed to the following:

   - "Configuring the Switches" on page 79

Configuring the Switches

The `discover` command initializes and configures the system components for the SGI ICE system. You use the `discover` command to configure the SGI ICE system's management switches first, and then if you have MCells, you configure the MCell switches. After you configure the switches, you can configure the nodes.

The following procedure is an overview of the switch configuration process.

Procedure 3-17 To configure SGI ICE X switches

1. Use one of the following procedures to configure the management switches:

   - "Configuring Management Switches With a MAC File" on page 80

- "Configuring Management Switches Without a MAC File" on page 82

The procedures differ depending on whether you have a media access control (MAC) file or not. The MAC file shows the MAC addresses of the components in your environment. Switch discovery and configuration can complete more quickly if you obtain this file. Without this file, you need to power cycle each switch manually.

The following is an example MAC file:

```
r1lead 00:30:48:9e:f2:59 00:30:48:F2:7E:A2
r2lead 00:25:90:01:4e:3c 00:25:90:01:6c:cc
service0 00:25:90:00:3b:8f 00:25:90:01:6e:9e
mgmtsw0 00:26:f3:c3:7a:40 00:26:f3:c3:7a:40
```

The content of a MAC file is as follows:

Column	Content
1	The component's ID.
2	For nodes, column 2 contains the MAC address of baseboard management controller (BMC) on the component. For switches, column 2 contains the MAC address of the first network interface card (NIC), `eth0`. Switches do not have a BMC.
3	The MAC address of the first network interface card (NIC), `eth0`. For switches, columns two and three are identical in the MAC file.

2. (Conditional) Use the following procedure to configure MCell switches:

   "(Conditional) Configuring MCell Switches" on page 84

Configuring Management Switches With a MAC File

```
interface ethernet 2/47
no lacp
shutdown
switchport allowed vlan add 3 tagged
interface ethernet 1/47
no lacp
```

```
switchport allowed vlan add 3 tagged
lacp
interface ethernet 2/47
lacp
no shutdown
end
from:http://linux.engr.sgi.com/wiki/index.php/CB3_Switch_Configuration#configure_p
```

The following procedure explains how to configure your switches when you have each switch's MAC information in a MAC file.

Procedure 3-18 To configure switches — with a MAC file

1. Gather information about the switches in your SGI ICE X system.

   Visually inspect your system. Note the types of switches you have and their identifiers. At a minimum, you have one management switch. You might also have InfiniBand switches and management switches attached to MCells. In this procedure, you configure only the management switches.

2. Log in as root to the system admin controller (SAC), and write the MAC file to a location on your SAC.

   For example, write it to `/var/tmp/mac_file`.

3. Power-on all the management switches.

4. Type the following command:

   # **discover --mgmtswitch *num* --macfile *path***

   For *num*, type the identifier for one of the switches. Visually inspect the outside of each switch to determine its identifier.

   For *path*, type the full path to the location of the MAC file.

   For example:

   # **discover --mgmtswitch 0 --macfile /var/tmp/mac_file**

5. Repeat the preceding command for each switch attached to your SGI ICE system.

6. Type the following command to retrieve information about the switches you discovered, and examine the output to confirm that all switches are included:

   # **cnodes --mgmtswitch**

7. After all switches have been discovered, proceed to the following:

Configuring Management Switches Without a MAC File

The following procedure explains how to configure your switches when you do not have the switch MAC information in a MAC file.

Procedure 3-19 To configure switches — without a MAC file

1. Gather information about the switches in your SGI ICE X system.

   Visually inspect your system. Note the types of switches you have and their identifiers. At a minimum, you have one management switch. You might also have InfiniBand switches and MCell switches. In this procedure, you configure only the management switches.

2. For each management switch stack, verify that only one cable goes from the first switch stack to the second switch stack.

   This cable should connect the master switch in the upper switch to the master switch in switch stack immediately below. Each switch can have a cable plugged into its slave switch, but make sure the cables that connect the slave switches to each other are unplugged. This prevents looping.

3. Log in as the root user to the system admin controller (SAC), and type the following command:

   `# discover --mgmtswitch` *switch_ID*

   For *switch_type*, specify `mgmtswitch` or `ibswitch`.

   For example, the following command discovers management switch 0:

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

4. When prompted, connect the switch to a power source.

   The command discovers the MAC address of the switch after you connect the switch to a power source.

Note: Do not power-on the switch at this time. Only connect the switch to a power source.

---

5. Type the following command to save the MAC address to your MAC file:

   # **discover --show-macfile >** ***path***

   For *path*, type the full path to the location of the MAC file. For example, `/var/tmp/mac_file`.

6. Repeat the preceding steps for each switch that is attached to your SGI ICE system.

7. Type the following command to retrieve information about the switches that you discovered:

   # **cnodes --all**

   If the output is very long, direct the output to a file that you can examine with a text editor. For example:

   # **cnodes --all > switch_file**

8. After all switches have been discovered, proceed to the following:

Discovering Cascaded LG-E Switches

This section describes how to discover LG-E (LG-Ericsson) switches when cascading them.

Procedure 3-20 Discovering Cascaded LG-E Switches

When cascading LG-E switches, since the system topology uses stacking and one or more stack pairs are added to the configuration, to avoid looping, perform the following steps:

1. Stack both switches but do not connect them to the stacked top level switches.

2. Power on the switches and wait until one switch becomes the master and the other switch becomes a slave.

3. Connect the master switch to a top level master switch, for example, from port 1/48 on the cascaded master switch to port 1/48 on the top level master switch.

4. Use the `discover` command (see "`discover` Command" on page 84), to discover the switch pair. During the `discover` process, the `switchConfig` API

is called; it will set Link Aggregation Control Protocol (LACP) for port 1/48 and 2/48 on the cascaded switches.

5. Connect port 2/48 of the cascaded slave switch to port 2/48 on the top level slave switch.

Since LACP is already configured, no loop is created.

(Conditional) Configuring MCell Switches

Perform the procedure in this topic if you have an SGI ICE X system that includes MCells.

The following procedure explains how to configure the switches attached to the MCells.

Procedure 3-21 To configure MCell switches

1. Gather information about the MCell switches in your SGI ICE X system.

   Visually inspect your system. Note the switches identifiers, and note the port identifiers.

2. Log in as the root user to the system admin controller (SAC), and type the following command:

   `# switchconfig -s mgmtsw`*num* `-p` *port_num*

   For *num*, type the ID number of the management switch to which the cooling distribution unit (CDU) is attached.

   For *port_num*, type the port number.

3. Repeat the preceding step for each cooling distribution unit (CDU) and each cooling rack controller (CRC) attached to your system.

4. After all switches have been discovered, proceed to the following:

`discover` Command

The `discover` command is used to discover rack leader controllers (RLCs) and service nodes (and their associated BMC controllers) in an entire system or in a set of one or more racks that you select. Rack numbers generally start at one. Service nodes generally start at zero. The `discover` command is also used to discover external InfiniBand switches and system management switches.

Caution: It is best to discover system management switches prior to any other component. That is because, as you discover node types, the tool automatically reconfigures the switch to operate properly as it proceeds.

When you use the `discover` command to perform the discovery operation on your SGI ICE X system, you will be prompted with instructions on how to proceed (see "Installing Software on the Rack Leader Controllers (RLCs) and Service Nodes" on page 92).

When using the `--delrack` and `--delservice` options, the node is not removed completely from the database but it is marked with the administrative status `NOT_EXIST`. When you go to discover a node that previously existed, you now get the same IP allocations you had previously and the node is then marked with the administrative status of `ONLINE`. If you have a service node, for example, `service0`, that has a custom host name of "`myhost`" and you later go to delete `service0` using the `discover --delservice` command, the host name associated with it will still be present. This can cause conflicts if you wish to reuse the custom host name "`myhost`" on a node other than service0 in the future. You can use the `cadmin --db-purge --node service0` command that will remove the node entirely from the database (for more information, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195). You can then reuse the “`myhost`” name.

There is a new hardware typed named `generic`. This hardware type has its MAC address discovered, but it is for devices that only have a single MAC address and do not need to be managed by SMC for SGI ICE X software. The likely usage scenario is Ethernet switches that extend the management network that are necessary in large SGI ICE X configurations.

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

- The management switches should be the first hardware discovered in the system.
- The management switches should both start with their power cords unplugged (analogous to how SMC for SGI ICE X discovers RLCs and service nodes).

- The external switches can be given higher numbered service numbered if your site does not want them to take lower numbers.
- You can also elect to give these switches an alternate host name using the `cadmin` command after discovery is complete.
- Examples of using the `discover` command `generic` hardware type are, as follows:

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

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

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

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

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

Options:
  --delrack NUM[,FLAG]...        mark rack leaders as deleted
  --delservice NUM               mark a service node as deleted
  --delibswitch NUM              mark an external ib switch as deleted
  --delmgmtswitch NUM            mark a mgmt network switch as deleted
  --force                        avoid sanity checks that require input
  --ignoremac MAC                ignore the specified MAC address
  --macfile FILE                 read mac addresses from FILE
  --rack NUM[,FLAG]...           discover a specific rack or set of racks
  --rackset NUM,COUNT[,FLAG]...  discover count racks starting at #
  --service NUM[,FLAG]...        discover the specified service node
  --ibswitch NUM[,FLAG]...       discover the specified external ib switch
  --mgmtswitch NUM[,FLAG]...     discover the specified mgmt switch
  --show-macfile                 print output usable for --macfile to stdout

Details:
  Any number of management switches, racks, service nodes, or external
  switches can be discovered in one command line.  Rack numbers generally
```

```
  start at 1, service nodes, management switches, and infiniband switches
  generally start at 0.  An existing node can be re-discovered by re-running
  the discover command. An easier way to simply re-image a node is by
  using the cinstallman command, see the --next-boot and --assign-image
  options.

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

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

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

  The --macfile option can be used instead of discovering MACs by power cycling.
  All MACs to be discovered must be in the file.  External switches should
  simply repeat the same MAC twice in this file.  File format:
        Example file contents:
    r1lead 00:11:22:33:44:55 66:77:88:99:EE:FF
    service0 00:00:00:00:00:0A 00:00:00:00:00:0B
    extsw1 00:00:00:00:00:11 00:00:00:00:00:11

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

Switch Type Flags:
  voltaire-isr-9288 voltaire-isr-9096 voltaire-isr-9024 voltaire-isr-2004
  voltaire-isr-2012 voltaire4036 mellanox5030 mellanox5600


Other Flags:
```

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

Examples:
  Discover a top level management switch
    # discover --mgmtswitch 0
  You can later use the 'cadmin' command to give it a custom hostname if you
  so choose.

  Discover rack 1 and service node 0:
    # discover --rack 1 --service 0

  Discover service 0, using myimage and disabling redundnat_mgmt_network.
    # discover --service 0,image=myimage,redundant_mgmt_network=no

  Discover racks 1 and 4, service node 1, ignores MAC address 00:04:23:d6:03:1c:
    # discover --ignoremac 00:04:23:d6:03:1c --rack 1 --rack 4 --service 1

  Discover racks 1-5, service node 0-2, where service node 1 is Altix 450
  hardware and service node 2 is "other":
    # discover --rackset 1,5 --service 0,xe240 --service 1,altix450 --service 2,other

  Discover an external ib switch, corresponding to the voltaire-isr-9024
  hardware and IB0 fabric.
    # discover --ibswitch 0,voltaire-isr-9024,net=ib0,type=spine
  You can later use the 'cadmin' command to give it a custom hostname if you
  so choose.

  Discover a switch used to extend the Tempo management network - a generic
  device.
   # discover --service 99,generic
```

```
Discover two leaders for rack 1 (High Availability):
  # discover --rack 1,ha=all

Discover r1lead1 (High Availability):
  # discover --rack 1,ha=1

Discover r1lead2 (High Availability):
  # discover --rack 1,ha=2

Discover two leaders per rack for racks 1, 2, and 3 (High Availability):
  # discover --rackset 1,3,ha=all

Delete r1lead1 (High Availability):
  # discover --delrack 1,ha=1

Delete r1lead2 (High Availability):
  # discover --delrack 1,ha=2
```

EXAMPLES

Example 3-1 `discover` Command Examples

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

To discover a top level management switch, perform the following:

```
admin:~ # /opt/sgi/sbin/discover --mgmtswitch 0
```

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

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

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

In this example, service node `0` is an SGI Rackable C2108-TY10 system.

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

```
admin:~ # /opt/sgi/sbin/discover --rackset 1,5 --service0,c2108 --service 1,altix450 --service 2,other
```

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

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

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

Note: You may direct a service node to image itself with a custom image later, without re-discovering it. See "`cinstallman` Command" on page 154.

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

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

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

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

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

An example command is, as follows:

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

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

To remove a switch, perform the following:

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

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

An example command is, as follows:

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

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

```
admin:~ # discover --service0,xe500,redundant_mgmt_network=no
```

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

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

Configuring the Rack Leader Controllers (RLCs) and Service Nodes

The `discover` command identifies and configures the RLCs and service nodes on the SGI ICE X system.

Installing SMC for ICE X System Admin Controller (SAC) Software

.

`cmcdetectd` Daemon

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

The `cmcdetectd` daemon performs the following:

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

- Once moved, the dynamic host configuration protocol (DHCP) requests are directed to the rack VLAN for a given rack and detected by the rack leader controller (RLC) when it is discovered.
- If the `cmcdetectd` daemon detects any SGI ICE X CMCs, it automatically sets to true the switch management network variable to true (see "Configuring a Switch Management Network" on page 73).
- If you install a second slot or re-install a system, the switch is already configured and the `cmcdetectd` daemon does not see the requests any more. It is a good practice to manually configure the switch management network setting using the `configure-cluster` option.

Installing Software on the Rack Leader Controllers (RLCs) and Service Nodes

The `discover` command, described in "`discover` Command" on page 84, sets up the RLC 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 RLC and then how to install software on the RLCs.

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

To change the `dhcp` option code number used for this operation, see the `cadmin --set-dhcp-option` option. The `--show-dhcp-option` will show the current value. For more information on the using the `cadmin` command, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195.

Procedure 3-22 Installing Software on the RLCs and Service Nodes

To install software on the RLCs, perform the following steps:

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

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

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

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

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

   When the node has electrical power, the BMC starts up even though the system is not powered on. The BMC does a network DHCP request that the discover script intercepts and then configures the cluster database and DHCP with the MAC address for the BMC. The BMC then retrieves its IP address. Next, this script instructs the BMC to power up the node. The node performs a DHCP request that the script intercepts and then configures the cluster database and DHCP with the MAC address for the node. The RLC 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 RLC and start the process all over.

   You may watch install progress by using the `console` command. For example, `console r1lead` connects you to the console of the `r1lead` so that you can watch installation progress. The sessions are also logged. For more information on the `console` command, see "Console Management" on page 201.

3. Using the identify light, you can configure all the RLCs and service nodes in the cluster without having to go back and fourth to and from your workstation

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

4. Shortly after the `discover` command reports that discovery is complete for a given node, that node installs itself. If you supplied multiple nodes on the `discover` command line, it is possible multiple nodes could be in different stages of the imaging/installation process at the same time. When the RLC boots up for the first time, one process it starts is the `blademond` process. This process discovers the IRUs and attached blades and sets them up for use. The `blademond` process is described in "Configuring the `blademond` Rescan Interval" on page 77, including which files to watch for progress and includes a `blademond --help` statement.

   Some sites choose to turn the `blademond` daemon off and only enable it periodically. The `blademond --scan-once` option allows you to easily run `blademond` once from the command line and watch the output.

   Note: You would never run `blademond --scan-once` command if `blademond` is already running as a daemon on your system.

   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 RLC 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 96.

Discovering Compute Nodes

This section describes how to discover compute nodes in your SGI ICE X system. The `blademond` daemon that runs on the rack leader controllers (RLCs) calls the `discover-rack` command to discover a rack and integrate new compute nodes (blades). For more information, see "Configuring the `blademond` Rescan Interval" on page 77.

Procedure 3-23 Discovering Compute Nodes

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

1. Complete the steps in "Installing Software on the Rack Leader Controllers (RLCs) and Service Nodes" on page 92.
2. For instructions on how to configure, start, verify, or stop the InfiniBand Fabric management software on your SGI ICE X system, see Chapter 5, "System Fabric Management" on page 227.

Note: The InfiniBand fabric does not automatically configure itself. For information on how to configure and start up the InfiniBand fabric, see Chapter 5, "System Fabric Management" on page 227.

Service Node Discovery, Installation, and Configuration

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

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

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

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

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

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

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

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

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

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

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

You can use the `cadmin --set-dhcp-option {value}` option, to change the SMC for SGI ICE X DHCP option identifier so it is different from your house network. For more information on the `cadmin` command, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195.

InfiniBand Configuration

Before you start configuring the InfiniBand network, you need to ensure that all hardware components of the cluster have been discovered successfully, that is, system admin controller (SAC), rack leader controller (RLC), service and compute nodes. You also need to be finished with the cluster configuration steps in "Running the Cluster Configuration Tool" on page 61.

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

```
% cattr set disableIbSwitchMonitoring true
```

To configure the InfiniBand network, start the `configure-cluster` command again on the system admin controller (SAC). Since the **Initial Setup** has been done already, you can now use the **Configure InfiniBand Fabric** option to configure the InfiniBand fabric as shown in Figure 3-9.

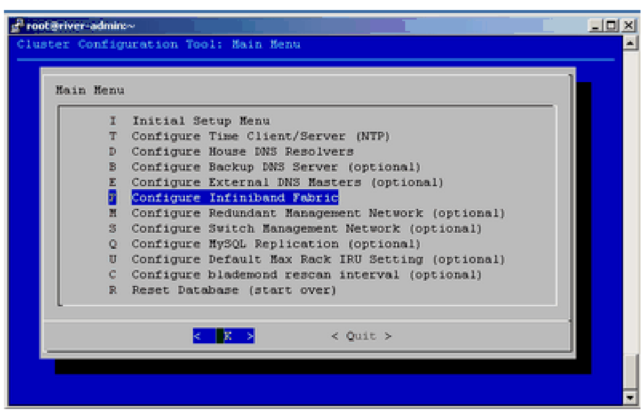

Figure 3-9 Configure InfiniBand Fabric from Cluster Configuration Tool

Select the **Configure InfiniBand Fabric** option, the InfiniBand Fabric Management tool appears, as shown in Figure 3-10.

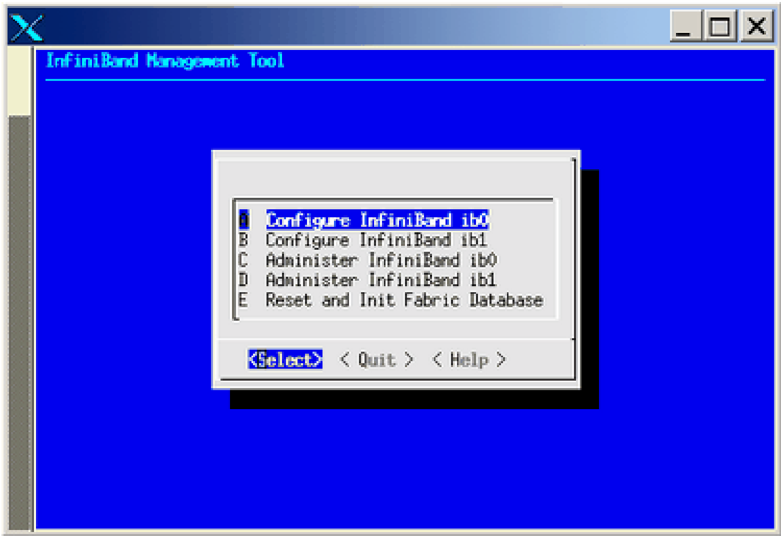

Figure 3-10 InfiniBand Management Tool Screen

Use the the online help available with this tool to guide you through the InfiniBand configuration. After configuring and bringing up the InfiniBand network, select the **Administer InfiniBand ib0** option or the **Administer InfiniBand ib1** option, the **Administer InfiniBand** screen appears as shown in Figure 3-11. Verify the status using the **Status** option.

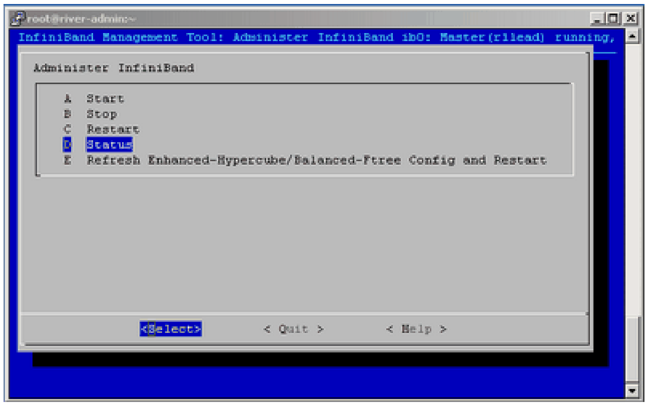

Figure 3-11 Administer InfiniBand GUI

The **Status** option returns information similar to the following:

```
Master SM
Host = r1lead
Guid = 0x0002c9030006938b
Fabric = ib0
Topology = hypercube
Routing Engine = dor
OpenSM = running
```

Press the `Enter` key to return to the `configure-cluster` GUI.

Configuring the Service Node

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

- "Link Aggregation, Rack Leader Controllers (RLCs), and Service Nodes" on page 101
- "Service Node Configuration for NAT" on page 101
- "Using External DNS for Compute Node Name Resolution" on page 104
- "Service Node Configuration for DNS" on page 105
- "Service Node Configuration for NFS " on page 106

- "Service Node Configuration for NIS for the House Network" on page 106

Link Aggregation, Rack Leader Controllers (RLCs), and Service Nodes

Link aggregation is initiated for an SGI ICE X cluster when one of the following events occurs:

- The **Configure Switch Management Network** option on the **Cluster Configuration Tool: Main Menu** is used to configure the switch management network (see "Configuring a Switch Management Network" on page 73).
- The `cmcdetectd` daemon finds SGI ICE X CMCs and turns link aggregation on automatically.

There are two potential types of interface bonding in play on an SGI ICE X and the default is link aggregation 802.3ad. The two bonding modes are, as follows:

- Active-backup, which is what we used in SGI ICE 8400 in redundant management network setups
- Link aggregation 802.3ad (The SGI ICE X default)

In order for link aggregation 802.3ad to work properly, the target service node must have a BMC with a dedicated Ethernet connection. If the BMC is in-band, sharing the physical wire with `eth0`, when link aggregation is negoiated, the BMC will disappear.

In order to force active-backup bonding and avoid this problem, you can perform one of the following:

- Use the `discover` command with the `mgmt_bonding=active-backup` flag
- In the case where a node is already installed, use the `cadmin` command with the `--set-mgmt-bonding` option.

You can also use the `cadmin` command to set link aggregation 802.3ad on the system admin controller (SAC) to increase bandwith capacity between the SAC, the RLC, and service nodes.

Service Node Configuration for NAT

You may want to reach network services outside of your SGI ICE X 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 3-24 Service Node Configuration for NAT

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

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

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

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

   ---

2. Update all of the compute node images with default route configured for NAT.

   SGI recommends a script on the system admin controller (SAC) in `/opt/sgi/share/per_host_customization/global/sgi-static-routes` that can customize the routes based upon rack, IRU, and slot of the compute blade. Some examples are available in that script.

3. Use the `cimage --push-rack` command to propagate the changes to the proper location for compute nodes to boot. For more information on using the `cimage` command, see "`cimage` Command" on page 169 and "Customizing Software On Your SGI ICE X System" on page 161.

4. Use the `cimage --set` command to select the image.

5. Reboot/reset the compute nodes using that desired image.

6. Once the service node(s) has NAT enabled, is attached to an operational house network, and the compute nodes are booted from an image which sets their routing to point at the service node, test the NAT operation by using the `ping`(8) command to ping known IP addresses on the house network from an interactive session on the compute blade.

7. See the troubleshooting discussion that follows.

Troubleshooting Service Node Configuration for NAT

Troubleshooting can become very complex. The first steps are to determine that the service node(s) are correctly configured for the house network and can ping the house IP addresses. Good choices are house name servers possibly found in the /etc/resolv.conf or /etc/name.d.conf files on the system admin controller (SAC). 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 3-2 tcpdump Command Examples

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

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

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

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

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

Using External DNS for Compute Node Name Resolution

You may want to configure service node(s) to act as NAT gateways for your cluster (see "Service Node Configuration for NAT" on page 101) and to have the host names for the compute nodes in the cluster resolve through external DNS servers.

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

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

To do this, use the `configure-cluster` tool (see "Running the Cluster Configuration Tool" on page 61). The menu item that handles this operation is **Configure External DNS Masters (optional)**, as shown in Figure 3-12 on page 104.

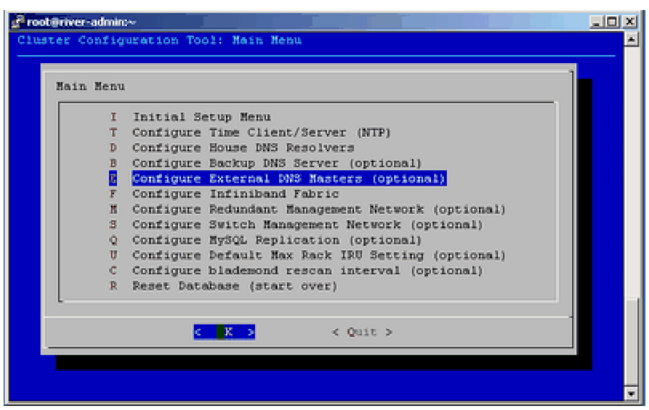

Figure 3-12 Configure External DNS Masters Option Screen

From the **Configure External DNS Master(s)** screen, click the `Yes` button, as shown in Figure 3-13 on page 105.

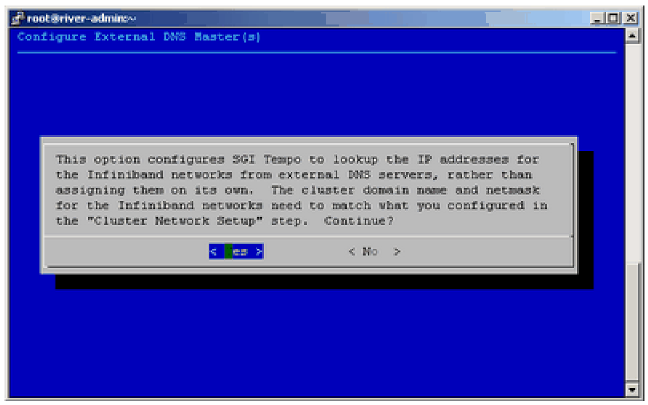

Figure 3-13 Configure External DNS Master(s) Screen

Some important considerations are, as follows:

- It is important to note that if you choose to use external DNS, you need to make this change **before** discovering anything. The change is **not** retroactive. If you have already discovered some nodes, then turn on external DNS support, the IP addresses assigned by SMC for SGI ICE X for the nodes already discovered will remain.
- This is an optional feature that only a small set of customers will need to use. It should not be used by default.
- This feature only makes sense if the compute nodes can reach the house network. This is not the default case for SGI ICE X systems.
- It is assumed that you have already configured a service node to act as a NAT gateway to your house network (see "Service Node Configuration for NAT" on page 101) and that the compute nodes have been configured to use that service node as their gateway.

Service Node Configuration for DNS

For information on setting up DNS, see Figure 3-6 on page 67.

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 3-25 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 file `/opt/sgi/share/per_host_customization/global/sgi-fstab` on the system admin controller (SAC), or edit an image-specific script.
2. Add the mount point, push the image, and reset the node.
3. The server's export should get mounted. If it is not, use the technique for troubleshooting outlined in "Troubleshooting Service Node Configuration for NAT" on page 103.

Service Node Configuration for NIS for the House Network

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

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

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

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

Procedure 3-26 NIS with Compute Nodes Directly Accessing the House NIS Infrastructure

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

1. In this case, you do not have to set up any additional NIS servers. Instead, each service node and compute node should be configured to bind to the existing house network servers. The nodes should already have the `ypbind` package

installed. The following steps should work with most Linux distributions. You may need to vary them slightly to meet your specific needs.

2. For service nodes, the instructions are very similar to those found in "Setting Up a SLES Service Node as a NIS Client" on page 117.

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

Procedure 3-27 NIS with a Service Node as a NIS Slave Server to the House NIS Master

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

1. Any service nodes that are NOT acting as an NIS slave server can be pointed at the existing house network NIS servers as described in Procedure 3-26, page 106. This is because they have house interfaces.

2. One (or more) service node(s) should be then be configured as NIS slave server(s) to the existing house network NIS Master server.

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

Setting Up an NFS Home Server on a Service Node for Your SGI ICE X 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 SGI ICE X system could create network bottlenecks that the hierarchical design of SGI ICE X 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 106.

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

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

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

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

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

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

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

```
# ls -l /dev/disk/by-id
total 0
lrwxrwxrwx 1 root root  9 2008-03-20 04:57 ata-MATSHITADVD-RAM_UJ-850S_HB08_020520 -> ../../hdb
lrwxrwxrwx 1 root root  9 2008-03-20 04:57 scsi-3600508e000000000307921086e156100 -> ../../sda
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part5 -> ../../sda5
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e000000000307921086e156100-part6 -> ../../sda6
lrwxrwxrwx 1 root root  9 2008-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a -> ../../sdb
lrwxrwxrwx 1 root root 10 2008-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a-part1 -> ../../sdb1
lrwxrwxrwx 1 root root  9 2008-03-20 09:57 usb-PepperC_Virtual_Disc_1_0e159d01a04567ab14E72156DB3AC4FA -
```

In the output, above, you can see that ID scsi-3600508e000000000307921086e156100 is in use by your system disk because it has a symbolic link pointing back to ../../sda. So do not consider that device.nThe other disk in the listing has ID scsi-3600508e0000000008dced2cfc3c1930a and happens to be linked to /dev/sdb.

Therefore, you know the by-id name you should use for your data is /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a because it is not connected with sda, which we found in the first ls example happened to be the root disk.

Partitioning, Creating, and Mounting Filesystems

Procedure 3-28 Partitioning and Creating Filesystems for an NFS Home Server on a Service Node

The following example uses /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a ID as the empty disk on which you will put your data. It is very important that you know this for sure. In "Setting Up an NFS Home Server on a Service Node for Your SGI ICE X System", an example is provided that allows you to determine where your root disk is located so you can avoid accidently destroying it. Remember, in some cases, /dev/sdb will be the root drive and /dev/sda or /dev/sdc may be the data drive.

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

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

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

1. Use the `parted`(8) utility, or some other partition tool, to create a partition on `/dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a`. The following example makes one filesystem out of the disk. You can use the `parted` utility interactively or in a command-line driven manner.

2. Make a new `msdos` label, as follows:

```
# # parted /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a mkpart primary ext2 0 249GB
Information: Don't forget to update /etc/fstab, if necessary.
```

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

```
# # parted /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a print
Disk geometry for /dev/sdb: 0kB - 249GB
Disk label type: msdos
Number  Start   End     Size    Type      File system  Flags
Information: Don't forget to update /etc/fstab, if necessary.
```

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

```
# # parted /dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a mkpart
primary ext2 0 249GB
Information: Don't forget to update /etc/fstab, if necessary.
```

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

```
# udevtrigger
```

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

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

An `ext3` example is, as follows:

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

An xfs example is, as follows:

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

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

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

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

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

---

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

```
# mount -a
```

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

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

   d.

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

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

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

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

---

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

Note: SGI recommends that you always work on clones of the SGI-supplied compute image so that you always have a base to copy to fall back to if necessary. For information on cloning a compute node image, see "Customizing Software Images" on page 166.

---

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 SAC. Note that the rest of the examples will resume using `/home`.

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

b. Add the `/home` filesystem to the compute nodes. SGI supplies an example script for managing this. You just need to add your new mount point to the `sgi-fstab` post-host-customization script.

c. Use a text editor to edit the following file:

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

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

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

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

---

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

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

Using `--push-rack` on an image that is already on the RLC 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 169.

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

For information on centrally managed user accounts, see "Setting Up a NIS Server for Your SGI ICE X System" on page 114. 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 RLCs. If you have more than one home server, you need to export all home filesystems on all home servers to the server acting as the NIS master. You also need to export the filesystems to the NIS master using the `no_root_squash` `exports` flag.

Home Directories on NAS

If you want to use NAS server for scratch storage or make home filesystems available on NAS, you can follow the instructions in "Setting Up an NFS Home Server on a Service Node for Your SGI ICE X System" on page 107. In this example, you need to replace `service0-ib1` with the `ib1` InfiniBand host name for the NAS server and you need to know where on the NAS server the home filesystem is mounted to craft the `sgi-fstab` script properly.

RHEL Service Node House Network Configuration

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

Some helpful hints are, as follows:

- On service nodes, the cluster interface is `eth0`. Therefore, do not configure this interface as it is already configured for the cluster network.
- Do not make the public interface a `dhcp` client as this can overwrite the `/etc/resolv.conf` file.

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

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

Setting Up a NIS Server for Your SGI ICE X System

This section describes how to set up a network information service (NIS) server running SLES 11 for your SGI ICE X 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 106. This section covers the following topics:

- "Tasks You Should Perform After Changing a Rack Leader Controller (RLC)" on page 122

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

Setting Up a SLES Service Node as a NIS Master

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

Procedure 3-29 Setting Up a SLES Service Node as a NIS master

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

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

---

1. Start up YaST, as follows:

   ```
   # yast nis_server
   ```

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

Note: You are now in the **NIS Master Server Slaves Setup**. Just now, you can enter the already defined rack leader controllers (RLSs) here. If you add more RLCs or re-discover RLCs, you will need to change this list. For more information, see "Tasks You Should Perform After Changing a Rack Leader Controller (RLC)" on page 122.

---

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

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

Setting Up a SLES Service Node as a NIS Client

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

Note: You do not do this on the NIS Master service node that you already configured as a client in "Setting Up a SLES Service Node as a NIS Master" on page 115.

Procedure 3-30 Setting Up a SLES Service Node as a NIS Client

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

1. Enable `ypbind`, perform the following:

   ```
   # chkconfig ypbind on
   ```

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

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

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

   ```
   ypserver r1lead
   ypserver r2lead
   ```

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

   ```
   # rcypbind start
   ```

   The service node is now bound.

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

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

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

This section provides two sets of instructions for setting up rack leader controllers (RLCs) as NIS slave servers. It is possible to make all these adjustments to the RLC 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 RLC 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 3-31 Setting up an RLC as a NIS Slave Server and Client

Use the following set of commands from the system admin controller (SAC) to set up an RLC as a NIS slave server and client.

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

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

Setting up the SLES Compute Nodes to be NIS Clients

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

Procedure 3-32 Setting up the Compute Nodes to be NIS Clients

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

1. Create a compute node image clone. SGI recommends that you always work with a clone of the compute node images. For information on how to clone the compute node image, see "Customizing Software Images" on page 166.

2. Change the compute nodes to use the cloned image/kernel pair, as follows:

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

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

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

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

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

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

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

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

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

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

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

NAS Configuration for Multiple IB Interfaces

The NAS cube needs to get configured with each InfiniBand fabric interface in a separate subnet. These fabrics will be separated from each other logically, but attached to the same physical network. For simplicity, this guide assumes that the `-ib1` fabric for the compute nodes has addresses assigned in the `10.149.0.0/16` network. This guide also assumes the lowest address the cluster management software has used is `10.149.0.1` and the highest is `10.149.1.3` (already assigned to the NAS cube).

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

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

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

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

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

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

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

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

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

Then add the following to the end of the file:

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

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

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

Creating User Accounts

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

Procedure 3-33 Creating User Accounts on a NIS Server

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

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

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

3. Provide the user a password, as follows:

   ```
   # passwd juser
   ```

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

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

Tasks You Should Perform After Changing a Rack Leader Controller (RLC)

If you add or remove an RLC, for example, if you use `discover` command to discover a new rack of equipment, you will need to configure the new RLC to be a NIS slave server as described in "Setting Up a SLES Service Node as a NIS Client" on page 117.

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

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

Installing SMC for SGI ICE Patches and Updating SGI ICE Systems

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

Note: To use the Subscription Management Tool (SMT) and run the `sync-repo-updates` script you must register your system with Novell using **Novell Customer Center Configuration**. This is in the **Software** category of YaST (see "Register with Novell " on page 125 and "Configuring the SMT Using YaST" on page 126).

Overview of Installing SMC for SGI ICE Patches

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

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

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

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

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

This following sections describe these steps, as follows:

- "Update the Local Package Repositories on the System Admin Controler (SAC)" on page 124
- "Installing Updates on Running System Admin Controller (SAC), Rack Leader Controller (RLC), and Service Nodes " on page 130

Update the Local Package Repositories on the System Admin Controler (SAC)

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

Mirroring Distribution Updates

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

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

Update the SGI Package Repositories on the System Admin Controller (SAC)

SGI provides a `sync-repo-updates` script to help keep your local package repositories on the SAC synchronized with available updates for the SMC for SGI ICE X, SGI Foundation, SGI Performance Suite, and SLES products. The script is located in `/opt/sgi/sbin/sync-repo-updates` on the SAC.

The `sync-repo-updates` script requires your Supportfolio user name and password. You can supply this on the command line or it will prompt you for it. With this login information, the script contacts the SGI update server and downloads the updated packages into the appropriate local package repositories.

For SLES, if you installed and configured the `SMT` tool as described in "SLES System Admin Controllers (SACs): Update the SLES Package Repository" on page 125, the `sync-repo-updates` script will also download any updates to SLES from the Novell update server. When all package downloads are complete, the script updates the repository metadata.

Once the script completes, the local package repositories on the SAC should contain the latest available package updates and be ready to use with the `cinstallman` command.

The `sync-repo-updates` script operates on all repositories, not just the selected reposistory.

Note: You can use the `crepo` command to set up custom repositories. If you add packages to these custom repositories later, you need to use the `yume --prepare --repo` command on the custom repository so that the metadata is up to date. Run the `cinstallman --yum-node --node admin clean all` command and then the `yum/yume/cinstallman` command.

SLES System Admin Controllers (SACs): Update the SLES Package Repository

In 1.8 (or later), SLES updates are mirrored to the SAC using the SUSE Linux Enterprise Subscription Management Tool. The Subscription Management Tool (SMT) is used to mirror and distribute updates from Novell. SMC for SGI ICE software only uses the mirror abilities of this tool. Mechanisms within SMC for SGI ICE are used to deploy updates to installed nodes and images. SMT is described in detail in the SUSELinux Enterprise *Subscription Management Tool Guide*. A copy of this manual is in the `SMT_en.pdf` file located in the `/usr/share/doc/manual/sle-smt_en` directory on the SAC. Use the `scp`(1) command to copy the manual to a location where you can view it, as follows:

```
admin :~ # scp /usr/share/doc/manual/sle-smt_en/SMT_en.pdf user@domain_name.mycompany.com:
```

Register with Novell

Register your system with Novell using **Novell Customer Center Configuration**. This is in the **Software** category of YaST. When registering, use the email address that is already on file with Novell. If there is not one on file, use a valid email address that you can associate with your Novell login at a future date.

The SMT will not be able to subscribe to the necessary update channels unless it is configured to work with a properly authorized Novell login. If you have an activation code or if you have entitlements associated with your Novell login, the SMT should be able to access the necessary update channels.

More information on how to register, how to find activation codes, and how to contact Novell with questions about registration can be found in the YaST help for Novell Customer Center Configuration.

Configuring the SMT Using YaST

At this point, your system admin controller (SAC) should be registered with Novell. You should also have a Novell login available that is associated with the SAC. This Novell login will be used when configuring the SMT described in this section. If the Novell login does not have proper authorization, you will not be able to register the appropriate update channels. Contact Novell with any questions on how to obtain or properly authorize your Novell login for use with the SMT.

Procedure 3-34 Configuring SMT Using YaST

Note: In step 8, a window pops up asking you for the Database root password. View the file `/etc/odapw`. Enter the contents of that file as the password in the blank box.

To configure SMT using YaST, perform the following steps:

1. Start up the YaST tool, as follows:

   ```
   admin:~ # yast
   ```

2. Under **Network Services**, find **SMT Configuration**
3. For **Enable Subscription Management Tool Service (SMT)**, check the box.
4. For **NU User**, enter your Novell user name.
5. For **NU Password**, enter your Novell password.

   Note: It is the mirror credentials you want. You can have a login that gets updates but cannot mirror the repository.

   ---

6. For **NU E-Mail**, use the email with which you registered.
7. For your **SMT Server URL**, just leave the default.

   It is a good idea to use the test feature. This will at least confirm basic functionality with your login. However, it does not guarantee that your login has access to all the desired update channels.

   Note that **Help** is available within this tool regarding the various fields.

8. When you click **Next**, a window pops up asking for the Database root password. View the file `/etc/odapw`. Enter the contents of that file as the password in the blank box.

   A window will likely pop up telling you that you do not have a certificate. You will then be given a chance to create the default certificate. Note that when that tool comes up, you will need to set the password for the certificate by clicking on the certificate settings.

Setting up SMT to Mirror Updates

This section describes how to set up SMT to mirror the appropriate SLES updates.

Procedure 3-35 Setting up SMT to Mirror Updates

To set up SMT to mirror updates, from the system admin controller (SAC), perform the following steps:

1. Refresh the list of available catalogs, as follows:

   ```
   admin:~ # smt-ncc-sync
   ```

2. Look at the available catalogs, as follows:

   ```
   admin:~ # smt-catalogs
   ```

   In that listing, you should see that the majority of the catalogs matching the SAC distribution (distro) **sles11**) have "**Yes**" in the "**Can be Mirrored**" column.

3. Use the `smt-catalogs -m` command to show you just the ones that you are allowed to mirror.

4. From the **Name** column, choose the entities with the ending of **`-Updates`** matching channels matching the installed distro. For example, if the base distro is SLES11, you might choose:

   ```
   SLE11-SMT-Updates
   SLE11-SDK-Updates
   SLES11-Updates
   ```

5. This step shows how you might enable the catalogs. Each time, you will be presented with a menu of choices. Be sure to select only the **x86_64 version** and if given a choice between **sles** and **sled**, choose **sles**, as follows:

```
admin:~ # smt-catalogs -e SLE11-SMT-Updates
admin:~ # smt-catalogs -e SLE11-SDK-Updates
admin:~ # smt-catalogs -e SLES11-Updates
```

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

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

Warning: SMC for SGI ICE does not map the concept of channels on to its repositories. This means that any channel you subscribe to will have its RPMs placed into the distribution repository. Therefore, only subscribe the SMC for SGI ICE X SAC to channels related to your SMC for SGI ICE X cluster needs.

Downloading the Updates from Novell and SGI

At this time, you should have your update channels registered. From here on, the `sync-repo-updates` script will do the rest of the work. That script will use SMT to download all the updates and position those updates in to the existing repositories so that the various nodes and images can be upgraded.

Run `/opt/sgi/sbin/sync-repo-updates` script.

After this completes, you need to update your nodes and images (see "Installing Updates on Running System Admin Controller (SAC), Rack Leader Controller (RLC), and Service Nodes " on page 130).

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

RHEL System Admin Controller (SAC): Update the RHEL Package Repository

This section describes how to keep your packages up to date on RHEL-based SACs. The general idea is that we download all updates in to the RHEL repository, and then use SGI Management Center for ICE tools to deploy the updates to nodes and images.

Perform the following:

- Register with RHN. This can be done, as follows:

  ```
  # rhn_register
  ```

- Once registered, the `sync-repo-updates` command will synchronize the latest version of update packages in to the RHEL 6 repository on the system.

  ```
  # sync-repo-updates
  ```

Update Distros That Do Not Match the System Admin Controller (SAC)

In situations where you have software distributions (distros) present that do not match the distro installed on the SAC, you have to arrange to download the updates on your own.

SLES

The instructions provided earlier show how to set up Novell SMT for the system admin controller (SAC). You could use similar ideas to configure your own SMT server somewhere on your network. Once the RPMs are staged on that server, you can copy them to the SAC using `rsync` or some other similar transport method. Remember to update the repository metadata after you update the packages. For example:

```
# yume --prepare --repo /tftpboot/distro/sles11sp1
```

RHEL

You can register with RHN on a RHEL server on your network. Then, you can look at the `/opt/sgi/sbin/sync-repo-updates` script to see how it stages the packages (search for `RHN` in that file). Following that example, you can set up a server on your house network to stage the files. Then you can copy the staged packages to the system admin controller (SAC) into the matching distro repository. You need to update the repository metadata after copying packages using `yume` in a way similar to this:

```
# yume --prepare --repo /tftpboot/distro/rhel6.0
```

Note: You can always make a managed service node provide the function of staging the updates.

Installing Updates on Running System Admin Controller (SAC), Rack Leader Controller (RLC), and Service Nodes

This section explains how to update existing nodes and images to the latest packages in the repositories.

To install updates on the SAC, perform the following command from the SAC:

```
admin:~ # cinstallman --update-node --node admin
```

To install updates on all online RLCs, perform the following command from the SAC:

```
admin:~ # cinstallman --update-node --node r\*lead
```

To install updates on all managed and online service nodes, perform the following from the SAC:

```
admin:~ # cinstallman --update-node --node service\*
```

To install updates on the SAC, all online RLCs, and all online and managed service nodes with one command, perform the following command from the SAC:

```
admin:~ # cinstallman --update-node --node \*
```

Please note the following:

- The `cinstallman` command does not operate on running compute nodes. For compute nodes, it is an image management tool only. You can use it to create and update compute images and use the `cimage` command to push those images out to RLCs (see "`cimage` Command" on page 169).

  For managed service nodes and RLCs, you can use the `cinstallman` command to update a running system, as well as, images on that system.

- When using a node aggregation, for example, the asterisk (*), as shown in the examples above, if a node happens to be unreachable, it is skipped. Therefore, you should ensure that all expected nodes get their updated packages.

- For more information on the `crepo` and `cinstallman` commands, see "`crepo` Command" on page 150 and "`cinstallman` Command" on page 154, respectively.

Updating Packages Within System Imager Images

You can also use the `cinstallman` command to update `systemimager` images with the latest software packages.

Note: Changes to the kernel package inside the compute image require some additional steps before the new kernel can be used on compute nodes (see "Additional Steps for Compute Image Kernel Updates" on page 131 for more details). This note does **not** apply to rack leader controllers (RLCs) or managed service nodes.

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

```
admin:~ # cinstallman --update-image --image lead-sles11
admin:~ # cinstallman --update-image --image service-sles11
admin:~ # cinstallman --update-image --image compute-sles11
```

Note: Changes to the compute image on the system admin controller (SAC) are not seen by the compute nodes until the updates have been pushed to the RLCs with the `cimage` command. Updating RLC and managed service node images ensure that the next time you add or re-discover or re-image an RLC or service node, it will already contain the updated packages.

Before pushing the compute image to the RLC using the `cimage` command, it is good idea to clean the `yum` cache.

Note: The `yum` cache can grow and is in the writable portion of the compute blade image. This means it is replicated 64 times per compute blade image per rack and the space that may be used by compute blades is limited by design to minimize network and load issues on RLCs.

To clean the `yum` cache, from the SAC, perform the following:

```
admin:~ # cinstallman --yum-image --image compute-sles11 clean all
```

Additional Steps for Compute Image Kernel Updates

Any time a compute image is updated with a new kernel, you will need to run some additional steps in order to make the new kernel available. The following example assumes that the compute node image name is `compute-sles11` and that you have already updated the compute node image in the image directory per the instructions in "Creating Compute and Service Node Images Using the `cinstallman` Command" on page 174. If you have named your compute node image something other than `compute-sles11`, replace this in the example that follows:

1. Shut down any compute nodes that are running the `compute-sles11` image (see "Power Management Commands" on page 183).
2. Push out the changes with the `cimage --push-rack` command, as follows:

   ```
   admin:~ # cimage --push-rack compute-sles11 r\*
   ```

3. Update the database to reflect the new kernel in the `compute-sles11`, as follows:

   ```
   admin:~ # cimage --update-db compute-sles11
   ```

4. Verify the available kernel versions and select one to associate with the `compute-sles11` image, as follows:

   ```
   admin:~ # cimage --list-images
   ```

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

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

   Note: Replace `2.6.16.46-0.12-smp` with the actual kernel version.

6. Reboot the compute nodes with the new kernel/image.

Installing MPI on a Running SGI ICE X System

This section describes how to install MPI on an SGI ICE X system that has already been installed. The instructions in this section update existing images instead of creating new ones. It should be noted that integrating MPI before cluster deployment is easier.

SGI supplied media, such as SGI® MPI and SGI® Accelerate™ CDs, have embedded in them suggested package lists for each node type. The `crepo` command, used in the following example, makes use of these lists and indeed recomputes the lists when new media is added and then selected.

File names in this example are just illustrations.

Register SGI MPI and SGI Accelerate with SMC, as follows:

```
# crepo --add accelerate-1.0-cd1-media-rhel6-x86_64.iso
# crepo --add mpi-1.0-cd1-media-rhel6-x86_64.iso
```

Update the `crepo` selected repositories so that all repositories associated with the software distribution (distro) you are installing for are present. For example, if you want MPI to work on RHEL 6, you might do something like this:

Show what is currently selected (the asterisks to the left):

```
# crepo --show
* SGI-Management-Center-1.5-rhel6 : /tftpboot/sgi/SGI-Management-Center-1.5-rhel6
* SGI-Foundation-Software-2.5-rhel6 : /tftpboot/sgi/SGI-Foundation-Software-2.5-rhel6
* SGI-XFS-XVM-2.5-for-RHEL-rhel6 : /tftpboot/sgi/SGI-XFS-XVM-2.5-for-RHEL-rhel6
* SGI-Accelerate-1.3-rhel6 : /tftpboot/sgi/SGI-Accelerate-1.3-rhel6
* SGI-Tempo-2.5-rhel6 : /tftpboot/sgi/SGI-Tempo-2.5-rhel6
* SGI-MPI-1.3-rhel6 : /tftpboot/sgi/SGI-MPI-1.3-rhel6
* Red-Hat-Enterprise-Linux-6.2 : /tftpboot/distro/rhel6.2
```

Unselect unrelated repositories:

```
# crepo --unselect SGI-Tempo-2.5-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel.6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
# crepo --unselect SGI-Foundation-Software-2.5-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
# crepo --unselect Red-Hat-Enterprise-Linux-Server-6.2
Removing: /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmlist
Removing: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
```

Select RHEL 6 related repositories:

```
# crepo --select Red-Hat-Enterprise-Linux-6.2
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
# crepo --select SGI-Foundation-Software-2.5-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
```

```
# crepo --select SGI-XFS-XVM-2.5-for-RHEL-rhel6
Updating: /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-lead-rhel6.2.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
```

After performing the steps, above, the proper repositories are registered and selected so you can operate on them by default. Since you are using an already deployed system, you need to update existing images and potentially existing service nodes themselves. This example uses SGI suggested/ generated rpmlists. If you have custom rpmlists, you need to manually reconcile the two lists for each node type. The list fragments in /var/opt/sgi/sgi-repodata/ may help you.

For a service node image, perform the following:

```
# cinstallman --refresh-image --image service-rhel6.2 --rpmlist /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmli
```

For a compute node image, perform the following:

```
# cinstallman --refresh-image --image compute-rhel6.2 --rpmlist /etc/opt/sgi/rpmlists/generated-compute-rhel6.2.rpmli
```

Finally, you need to push the updated compute image to the rack leader controllers (RLCs).

Note: If the compute nodes are booted on the image and are using NFS for roots, you need to shut the compute nodes down before being able to run this command.

```
# cimage --push-rack compute-rhel6.2 r"*"
```

To make sure the compute nodes your are operating on have the associated compute image you just updated, perform a command similar to the following”

```
# cimage --set compute-rhel6.2 2.6.32-71.el6.x86_64 "*"
```

You can find the available images and kernels using the cimage --list-images command.

If you have booted service/login nodes, you likely want to refresh those running nodes also. (You could also reinstall them, as well). Here is a refresh example:

```
# cinstallman --refresh-node --node service0 --rpmlist /etc/opt/sgi/rpmlists/generated-service-rhel6.2.rpmlist
```

Now reset or bring up the nodes (depends on the state you left them). If you want to bring up all nodes, this command will not disrupt nodes already operating:

```
# cpower --system --up
```

System Operation

This chapter describes how to operate your SGI ICE system and covers the following topics:

- "Software Image Management" on page 147
- "Power Management Commands" on page 183
- "C3 Commands" on page 190
- "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195
- "Console Management" on page 201
- "Keeping System Time Synchronized" on page 203
- "Changing the Size of Per-node Swap Space" on page 208
- "Switching Compute Nodes to a `tmpfs` Root" on page 210
- "Setting up Local Storage Space for Swap and Scratch Disk Space" on page 211
- "Changing the Size of `/tmp` on Compute Nodes" on page 205
- "RAID Utility" on page 217
- "Restoring the `grub` Boot-loader on a Node" on page 220
- "Backing up and Restoring the System Database" on page 221
- "Enabling EDNS" on page 222
- "Firmware Management" on page 223

Enabling or Disabling the Redundant Management Network (RMN)

The cluster configuration tool contains menus that let you enable or disable the backup RMN. You can also enable or disable the RMN with the `discover` command's `mgmt_bonding=TYPE` parameter and with the `cadmin` command's `--set-mgmt-bonding` parameter. If you use the `cadmin` command to change a service node or a leader node, reboot the node to make your changes take effect. The following examples show how to use commands to configure the RMN.

Example 1. The following `discover` command disables the RMN on node `service0`:

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

Example 2. The following `cadmin` command enables the RMN on node `service0`:

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

Example 3. The following `cadmin` command enables the RMN on RLC `r1lead` and shows the required subsequent reboot:

To turn on the redundant managment network on an RLC, perform the following command:

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

Enabling or Disabling the Backup Domain Name Service (DNS) on an SGI ICE X Cluster

The cluster configuration tool contains menus that let you enable or disable the backup DNS. You can also enable or disable this feature from the command line. The `/opt/sgi/sbin/backup-dns-setup` command enables or disables the backup DNS.

Example 1. To retrieve current DNS backup information, type the following:

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

Example 2. To disable the backup DNS, type the following:

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

Example 3. To enable a backup DNS on `service0`, type the following:

```
# backup-dns-setup --set-backup service0
 Shutting down name server BIND waiting for named to shut down (29s)
done
sys-admin: update-configs: updating SMC for SGI ICE X configuration files
sys-admin: update-configs: -> dns
  ...
```

Changing Network Settings

After your system is installed, you might need to change some network settings. For example, you might need to change information if your house network information changes. Use your operating system software to make the changes, run the `configure_cluster` command to update the system database, and then run the `discover` command.

Boot Parameters, Disk Partitioning, and Managing a Multiboot System

On a multiple-slot SGI ICE X system, the system admin controller (SAC), the rack leader controller (RLC), and the service nodes all have the same disk layout. If you insert an operating system installation disk, power-on the SAC, and type **`install`** at the `boot:` prompt, the default behavior is for the system to create two slots to write the initial installation to slot 1. After the system is installed, you cannot increase the number of slots without destroying the data on the disks. You can configure up to five slots, for a total of five root/boot directory pairs. As you increase the number of slots, however, you decrease the amout of disk space per slot. SGI recommends a minimum of 100 Gb per slot.

One feature of a multi-boot system is that you can install a different operating systems, or different operating system versions, into different slots. You can boot the system with the operating system of your choice. This configuration might be useful if you want to test an operating system or other software. The following are some other characteristics of single-boot systems and multi-boot systems:

Multi-boot	**Single-boot**
RLCs and service nodes boot from their own disk. Data is retained in the master boot record (MBR).	RLCs and service nodes boot from the boot partition in the slot that is currently configured as the boot

	slot. Only the SAC retains data in the MBR.
RLCs and service nodes software is reinstalled from the SAC.	Software on the RLCs and service nodes is reinstalled over the network.

The following topics describe the boot parameters you can specify, disk partitioning, and operations such as cloning:

- .

Specifying Boot Parameters

When you use the SGI system admin controller (SAC) installation DVD to boot an SGI ICE X system, you can specify parameters at the `boot:` prompt. By default, if you type **`install`** at the boot prompt and the installer detects a SAC with exactly one blank disk, the installer partitions the SAC with two slots, and the installer writes the initial installation to slot 1.

If you specify more than one boot parameter, use a space to separate each parameter.

The following list shows the result of pressing Enter at the boot prompt and shows the result of the different boot options:

Parameter **Effect**

Press `Enter`

If you press `Enter` at the `boot:` prompt, without specifying any boot parameters, the installer creates two slots, but it installs a root directory on only one slot. The software repository is not configured, so you when you run the `configure-cluster` command, you need to use the tool to manually enter the distribution ISO images in the `/tftpboot` directory on the SAC.

You can use the second slot for a clone or preproduction testing.

`console=`*specs*

Specify console characteristics. Use when you connected to the compute node through a serial console. By default, this is set to `ttyS1,38400n8`.

The BIOS console and the IPMI console settings are configured in the SAC BIOS. If you configure the SAC BIOS to be different from the default, use this boot option to specify your console's characteristics.

`destructive=1`

Permits partitioning operations that are potentially destructive. Use this parameter only if you want to repartition a disk that contains data. Use in conjunction with the `re_partition_with_slots` parameter.

If the system encounters data in a partition, nothing destructive happens unless you also specify the `destructive=1` parameter.

`install`

Creates two slots on each compute node and installs the operating system from the DVD to slot 1. Both slots contain a root directory, and the installer writes the software distribution ISO images to each slot.

This is the recommended boot option for an initial confiiguration.

`install_slot=`*slot*

Specifies the slot into which the operating system is installed first. Specify `1` (default), `2`, `3`, `4`, or `5` for *slot*. By default, slot 1 is installed first. To direct the initial installation to a slot other than slot 1, specify this parameter.

If you specify a *slot* that appears to have data on it, no repartitioning is performed unless you also specify `destructive=1`.

Example: You can specify `install_slot=2` to direct the first installation into slot 2. In this case, after the boot completes, type the following `cadmin` command to set the default slot to slot 2:

```
# cadmin --set-default-root --slot 2
```

In this example, if you do not specify the default slot (slot 2) during the first boot of the SAC, then the SAC attempts to boot from an empty slot 1. The boot fails. If this happens, restart the boot, and select the install slot during the boot.

`netinst=`*path*

Specifies the NFS path to an ISO image for a network installation.

`re_partition_with_slots=`*slots*

Partitions the SAC system drive with between one and five slots. Specify `1`, `2` (default), `3`, `4`, or `5` for *slots*.

The SAC system drive must be blank in order for this parameter to have an effect. If the SAC system drive contains data and you want to reconfigure the system, also specify the `destructive=1` parameter.

For example, the following parameter creates five slots:

```
re_partition_with_slots=5
```

The installer creates partitions on the rack leader controllers (RLCs) and service nodes to mimic the SAC. If an RLC or service node is discovered to have a slot count that does not match the SAC, the system reinitializes the partitions on the RLC and service nodes. Likewise, if you change the number of slots on the SAC, the system updates the disk partitioning on the RLC and service nodes, too.

`rescue=1`

.

`serial`

Configures the system for serial console operations for the installation and later operations. If you do not specify `serial`, the system sends output to the VGA.

Also see `vga` in this list.

`vga`

Configures the system so that you can use the VGA screen for the installation and later operations.

If you press `Enter` at the boot prompt, the result is the same as if you had typed `vga`.

Also see `serial` in this list.

Partiton Layout for a Two-slot SGI ICE X System (Default)

Table 4-1 on page 143 shows the layout for a default two-slot SGI ICE X system.

Table 4-1 Example Partition Layout for a Two–slot SGI ICE X System (Default Layout)

Partition	File System Type	File System Label	Notes
1	swap	`sgiswap`	Partition layout for multiple slots.
2	Ext3	`sgidata`	SGI data partition.
3	-	N/A	Extended partition. Makes logicals out of the rest of the disk.
5	Ext3	`sgiboot`	Slot 1 `/boot` partition.
6	Ext3 or XFS	`sgiroot`	Slot 1 `/` partition.
7	Ext3	`sgiboot`	Slot 2 `/boot` partition.
8	Ext3 or XFS	`sgiroot`	Slot 2 `/` partition.

Partition Layout for a Five—slot SGI ICE X System

Table 4-2 on page 143 shows a sample partition layout for a five-slot SGI ICE X system. This layout yields five boot partitions.

Table 4-2 Example Partition Layout for a Five–slot SGI ICE X System

Partition	File System Type	File System Label	Notes
1	swap	`sgiswap`	Partition layout for multiple slots.
2	Ext3	`sgidata`	SGI data partition.
3	-	N/A	Extended partition. Makes logicals out of the rest of the disk.
5	Ext3	`sgiboot`	Slot 1 `/boot` partition.
6	Ext3 or XFS	`sgiroot`	Slot 1 `/` partition.
7	Ext3	`sgiboot`	Slot 2 `/boot` partition.
8	Ext3 or XFS	`sgiroot`	Slot 2 `/` partition.

Partition	File System Type	File System Label	Notes
9	Ext3	sgiboot	Slot 3 /boot partition.
10	Ext3 or XFS	sgiroot	Slot 3 / partition.
11	Ext3	sgiboot	Slot 4 /boot partition.
12	Ext3 or XFS	sgiroot	Slot 4 / partition.
13	Ext3	sgiboot	Slot 5 /boot partition.
14	Ext3 or XFS	sgiroot	Slot 5 / partition.

Partition Layout for a One-slot (Single—boot) SGI ICE X System

Table 4-3 on page 144 shows a sample partition layout for a single-boot SGI ICE X system. This layout shows one slot, which yields one boot partition. If you configure a single-slot system and later decide to add another partition, the addition process destroys all the data on your system.

Table 4-3 Example Partition Layout for a Single-boot SGI ICE X System

Partition	File System Type	File System Label	Notes
1	Ext3	sgiboot	/boot partition.
2	-	N/A	Extended partition. Makes logicals out of the rest of the disk.
5	swap	sgiswap	Swap partition.
6	Ext3 or XFS	sgiroot	/ partition.

If you upgrade an existing SGI ICE X system to a more current release of the SGI Management Center software, your SAC might have a partition layout that differs from the layouts shown in the preceding tables.

Managing a Multiboot System

Managing the Boot Slot and Changing the Boot Slot

If you configured more than one slot, you can boot from the boot partition in any of the slots. The following procedure explains how to change the system to boot from a different slot.

Procedure 4-1 To change the boot partition and enable the system to boot from a different slot

1. Log in as the root user to the system admin controller (SAC).
2. Type the following command to verify the current boot slot:

   ```
   # cadmin --show_root-labels
   admin node currently booted on slot: 1
   ```

3. (Optional) Change the default slot.

   Perform this step if you know the slot from which you want to boot.

   You can specfy the new slot now, or you can specify the new slot during the reboot.

   ```
   # cadmin --set-default-root --slot num
   ```

   For *num*, specify the new boot slot number. *num* can be `1`, `2`, `3`, `4`, or `5`.

   For example, to specify a boot from slot 2, type the following:

   ```
   admin:~ # cadmin --set-default-root --slot 2
   ```

4. Type the following command to shut down the entire system:

   ```
   # cpower --shutdown --system
   ```

5. Type the following command to reboot the SAC:

   ```
   # reboot
   ```

6. Monitor the reboot and, optionally, select the slot from which you want to boot.

   During the reboot, the system displays a screen that shows all the available slots and highlights the current boot slot. If you need to select a different boot slot, use the arrow keys to select a new slot from which to boot and press `Enter`.

If you do not select a new slot, the system boots from the highlighted slot after approximately 10 seconds.

7. Log in as the root user again.
8. Type the following command to reboot all the rack leader controllers (RLCs) and service nodes:

   ```
   # cpower --reboot --system
   ```

   If the IP addresses are configured differently within different slots, the `cpower` command might not be able communicate with the baseboard management controllers (BMC)s immediately after you reboot the SAC. If you have trouble connecting to the RLC and service node BMCs after you change slots, wait up to 15 minutes and issue the `cpower` command again. The wait enables the nodes to obtain new IP addresses.

Cloning a Slot

A script named `/opt/sgi/sbin/clone-slot` is available. This script allows you to clone a source slot to a destination slot. It then handles synchronizing the data and fixing up `grub` and `fstabs` to make the cloned slot a viable booting choice.

The script sanitizes the input values, then calls a worker script in parallel on all managed nodes and the system admin controller (SAC) that does the actual work. The `clone-slot` script waits for all children to complete before exiting.

Important: If the slot you are using as a source is the mounted / active slot, the script will shut down `mysql` on the SAC prior to starting the backup operation and start it when the backup is complete. This ensures there is no data loss.

Customizing the Slot Labels on a Multiboot System

You can use the `cadmin` command to label the slots on a multiboot SGI ICE X system. After an installation, the slot label is `(none)`.

Procedure 4-2 To customize the slot labels

1. Log into the system admin controller (SAC) as the root user.
2. Type the following command to retrieve the current labels:

   ```
   admin:~ # cadmin --show-root-labels
   ```

3. Type the command again, in the following format, to specify the slot and the label:

   `cadmin --show-root-labels --slot` *num* `--label "`*mylabel*`"`

   For *num*, type `1`, `2`, `3`, `4`, or `5` to specify the slot you want to label.

   For *mylabel*, type the label you want to apply to the slot.

   For example:

```
# cadmin --set-root-label --slot 1 --label "SLES"
# cadmin --show-root-labels
slot 1: SMC for ICE x.x / sles11: SLES
slot 2: SMC for ICE x.x / sles11: RHEL
slot 3: SMC for ICE x.x / sles11: SLES-2
```

Software Image Management

This section describes image management operations.

This section describes Linux services turned off on compute nodes by default, how you can customize the software running on compute nodes or service nodes, create a simple clone image of compute node or service node software, how to use the `cimage` command to push images to compute nodes, how to use `crepo` command to manage software image reposistories, and how to use the `cinstallman` command to create compute and service node images. It covers these topics:

- "Finding Which Distributions (distros) Are Supported" on page 148
- "Operating Systems Supported per Node Type" on page 149
- "Compute Node Services Turned Off by Default" on page 150
- "`crepo` Command" on page 150
- "`cinstallman` Command" on page 154
- "Customizing Software On Your SGI ICE X System" on page 161
- "`cimage` Command" on page 169
- "Using `cinstallman` to Install Packages into Software Images" on page 173

Finding Which Distributions (distros) Are Supported

To find a list of distributions supported on SGI ICE X nodes, perform the following commands from the system admin controller (SAC), as follows:

```
sys-admin:~ # cd /opt/sgi/share/rpmlists/distro/

sys-admin:/opt/sgi/share/rpmlists/distro # ls
compute-distro-centos5.4.rpmlist  lead-distro-rhel6.1.rpmlist
compute-distro-centos5.5.rpmlist  lead-distro-rhel6.2.rpmlist
compute-distro-centos6.0.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist    service-distro-centos5.5.rpmlist
compute-distro-rhel5.5.rpmlist    service-distro-centos6.0.rpmlist
compute-distro-rhel5.6.rpmlist    service-distro-rhel5.4.rpmlist
compute-distro-rhel5.7.rpmlist    service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist    service-distro-rhel5.6.rpmlist
compute-distro-rhel6.1.rpmlist    service-distro-rhel5.7.rpmlist
compute-distro-rhel6.2.rpmlist    service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist  service-distro-rhel6.1.rpmlist
compute-distro-sles10sp4.rpmlist  service-distro-rhel6.2.rpmlist
compute-distro-sles11sp1.rpmlist  service-distro-sles10sp3.rpmlist
lead-distro-centos6.0.rpmlist     service-distro-sles10sp4.rpmlist
lead-distro-rhel6.0.rpmlist       service-distro-sles11sp1.rpmlist
```

Operating Systems Supported per Node Type

This section describes what operating systems are supported for various ICE X nodes.

System Admin Controller (SAC)

The SAC supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Rack Leader Controller (RLC)

The RLC supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Service Nodes

Service nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5

Compute Nodes

Compute nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4

Compute Node Services Turned Off by Default

To improve the performance of applications running MPI jobs on compute nodes, most services are disabled by default in compute node images. To see what adjustments are being made, view the `/etc/opt/sgi/conf.d/80-compute-distro-services` script.

If you wish to change anything in this script, SGI suggests that you copy the existing script to `.local` and adjust it there. Perform the following commands:

```
# cd /var/lib/systemimager/images/compute-image-name
# cp etc/opt/sgi/conf.d/80-compute-distro-services 80-compute-distro-services.local
# vi etc/opt/sgi/conf.d/80-compute-distro-services.local
```

At this point, the configuration framework will execute the `.local` version, and skip the other. For more information on making adjustments to configuration framework files, see "SGI ICE X System Configuration Framework" on page 179.

Use the `cimage` command to push the changed image out to the rack leader controllers (RLCs).

`crepo` Command

You can use the `crepo` command to manage software repositories such as SGI Foundation, SMC, SGI Performance Suite, and the Linux distribution(s) you are using on your system. You also use the `crepo` command to manage any custom repositories you create yourself.

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

Each repository has associated with it a name, directory, update URL, selection status, and suggested package lists. The update URL is used by the `sync-repo-updates` command. The directory is where the actual `yum` repository exists, and is located in one of these locations, as follows:

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

The repository information is determined from the media itself when adding media supplied by SGI, Linux distribution media (SLES, RHEL, and so on.), and any other YaST-compatible media. For customer-supplied repositories, the information must be provided to the `crepo` command when adding the repository.

Repositories can be selected and unselected. Usually, SMC commands ignore unselected repositories. One notable exception is that `sync-repo-updates` always operates on all repositories.

The `crepo` command constructs default RPM lists based on the suggested package lists. The RPM lists can be used by the `cinstallman` command when creating a new image. These RPM lists are only generated if a single distribution is selected and can be found in `/etc/opt/sgi/rpmlists`; they match the form generated-`*.rpmlist`. The `crepo` command will tell you when it updates or removes generated rpmlists. For example:

```
# crepo --select SUSE-Linux-Enterprise-Server-10-SP3
Updating: /etc/opt/sgi/rpmlists/generated-compute-sles10sp3.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-sles10sp3.rpmlist
```

When generating the RPM lists, the crepo command combines the a list of distribution RPMs with suggested RPMs from every other selected repository. The distribution RPM lists are usually read from the /opt/sgi/share/rpmlists/distro directory. For example, the compute node RPM list for sles11sp1 is /opt/sgi/share/rpmlists/distro/compute-distro-sles11sp1.rpmlist. The suggested RPMs for non-distribution repositories are read from the /var/opt/sgi/sgi-repodata directory. For example, the rpmlist for SLES 11 SP1 compute nodes is read from /var/opt/sgi/sgi-repodata/SMC-for-ICE 1.5-for-Linux-sles11/smc-ice-compute.rpmlist.

The suggested rpmlists can be overridden by creating an override rpmlist in the /etc/opt/sgi/rpmlists/override/ directory. For example, to change the default SMC for ICE 1.5 suggested rpmlist, a file /etc/opt/sgi/rpmlists/override/SMC-for-ICE-1.5 -for-Linux-sles11/smc-ice-compute.rpmlist can be created.

The following example shows the contents of the /etc/opt/sgi/rpmlists directory after the crepo command has created the suggested RPM lists. Change directory (cd) to the /etc/opt/sgi/rpmlists directory. Use the ls command to see a list of rpms, as follows:

```
admin distro]# ls
compute-distro-centos5.4.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist    service-distro-rhel5.4.rpmlist
compute-distro-rhel5.5.rpmlist    service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist    service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist  service-distro-sles10sp3.rpmlist
compute-distro-sles11sp1.rpmlist  service-distro-sles11sp1.rpmlist
lead-distro-rhel6.0.rpmlist
```

Specifically, SMC software looks for /etc/opt/sgi/rpmlists/generate-*.rpmlist and creates an image for each rpmlist that matches.

It also determines the default image to use for each node type by hard-coding $nodeType-$distro as the type, where distro is the system admin controller's (SAC's) distro and nodeType is compute, service, rack leader controller (RLC), and so on. The default image can be overridden by specifying a global cattr attribute named image_default_$nodeType; for example, image_default_service. Use cattr --h, for information about the cattr command.

The following example shows the contents of the /etc/opt/sgi/rpmlists directory after the crepo command has created the suggested RPM lists. The files with -distro- in the name are the base Linux distro RPMs that SGI recommends.

Use the cd(1) to change to the /etc/opt/sgi/rpmlists directory. Use the ls command to see a list of RPMs, as follows:

```
admin:/etc/opt/sgi/rpmlists # ls
compute-minimal-sles11sp1.rpmlist  generated-lead-rhel6.2.rpmlist
generated-compute-rhel6.2.rpmlist  generated-service-rhel6.2.rpmlist
```

For more information on rpmlist customization information, see "Creating Compute and Service Node Images Using the cinstallman Command" on page 174.

You can use the crepo --show command to show the available repositories on the SAC, as follows:

```
sys-admin:~ # crepo --show
* SGI-Management-Center-1.5-rhel6 : /tftpboot/sgi/SGI-Management-Center-1.5-rhel6
* SGI-Foundation-Software-2.5-rhel6 : /tftpboot/sgi/SGI-Foundation-Software-2.5-rhel6
* SGI-XFS-XVM-2.5-for-RHEL-rhel6 : /tftpboot/sgi/SGI-XFS-XVM-2.5-for-RHEL-rhel6
* SGI-Accelerate-1.3-rhel6 : /tftpboot/sgi/SGI-Accelerate-1.3-rhel6
* SGI-Tempo-2.5-rhel6 : /tftpboot/sgi/SGI-Tempo-2.5-rhel6
* SGI-MPI-1.3-rhel6 : /tftpboot/sgi/SGI-MPI-1.3-rhel6
* Red-Hat-Enterprise-Linux-6.2 : /tftpboot/distro/rhel6.2
```

For a crepo command usage statement, perform the following:

```
admin:~ # crepo --h
crepo Usage:
Operations:
--help                : print his usage message

--add {path/URL}      : add SGI/SMC media to the system repositories
       --custom {name}: Optional.Use with -add to add custom repo under
                        /tftpboot Repo must pre-exist for this case.

--del {product}       : delete an add-on product and associated /tftpboot repo

--select {product}    : mark the product as selected

--show                : show available add-on products
```

```
--show-distro         : like show, but only reports distro media like sles10sp2

--show-updateurls     : Show the update sources associated add-on products

--reexport            : re-export all repositories with yume.  Use if there
                        was a yume export problem previously.

--unselect {product}  : mark the product as not selected


Flags:

Note for --add: If the pathname is local to the machine, it can be an
ISO file or mounted media.  If a network path is used -- such as an nfs
path or a URL -- the path must point to an ISO file.  The argument to
--add may be a comma delimited list to specify multiple source media.

Use --add for SGI/SMC media, to make the repos and rpms available. If the
supplied SGI/SMC media has suggested rpms from SMC node types, those
suggested rpms will be integrated with the default rpmlists for leader,
service, and compute nodes.  You can use create-default-sgi-images to
re-create the default images including new suggested packages or you can
just browse the updated versions in /etc/opt/sgi/rpmlists.

Use --add with --custom to register your own custom repository.  This will
ensure that, by default, the custom repository is available to yume and
mksiimage commands.  It is assumed you will maintain your own default package
lists, perhaps using the sgi default package lists in /etc/opt/sgi/rpmlists
or /opt/sgi/share/rpmlists as a starting point.  The directory and rpms within
must pre-exist.  This script will create the yum metadata for it.
```

Example:

```
crepo --add /tftpboot/myrepo --custom my-custom-name
```

`cinstallman` Command

The `cinstallman` command is a wrapper tool for several SMC operations that previously ran separately. You can use the `cinstallman` command to perform the following:

- Create an image from scratch

- Clone an existing image
- Recreate an image (so that any nodes associated with said image prior to the command are also associated after)
- Use existing images that may have been created by some other means
- Delete images
- Show available images
- Update or manage images (via `yume`)
- Update or manage nodes (via `yume`)
- Assign images to nodes
- Choose what a node should do next time it reboots (image itself or boot from its disk)
- Refresh the `bittorrent` tarball and `torrent` file for a compute node image after making changes to the expanded image

When compute images are created for the first time, a `bittorrent` tarball is also created. When images are pushed to rack leader controllers (RLCs) for the first time, `bittorrent` is used to transport the tarball snapshot of the image. However, as you make adjustments to your compute image, those changes do not automatically generate a new `bittorrent` tarball. We handle that situation by always doing a follow-up `rsync` of the compute image after transporting the tarball. However, as your compute image begins to diverge from the `bittorrent` tarball snapshot, it becomes less and less efficient to transport a given compute node image that is new to a given RLC.

You no longer need to use `yum`, `yume`, or `mksiimage` commands directly for most common operations. Compute images are automatically configured in such a way as to make them available to the `cimage` command.

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

```
admin:~ # cinstallman --help
Usage: blademond [OPTION] ...

Discover CMCs and blades managed by CMCs.

Note: This daemon normally takes no arguments.
```

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

[root@r1lead ~]# exit
logout
Connection to r1lead closed.
[root@river-admin ~]# clear
[root@river-admin ~]# pwd
/root
[root@river-admin ~]# cinstallman --h
cinstallman Usage:

cinstallman is a tool that manages:
 - image creation (as a wrapper to mksiimage)
 - node package updates (as a wrapper to yume)
 - image package updates (yume within a chroot to the image)

This is a convenience tool and not all operations for the commands that are
wrapped are provided.  The most common operations are collected here for
ease of use.

For operations that take the --node parameter, the node can be an aggregation
of nodes like cimage and cpower can take.  Depending on the situation,
non-managed or offline nodes are skipped.

The tool retrieves the registered repositories from crepo so that they
need not be specified on the command line.

Operations:
--help                  : print his usage message
--create-image          : create a new systemimager image
                          By default, requires --rpmlist and --image
                          Optional flags  below:
       --clone          : Clone existing image, requires --source, --image.
                          Doesn’t require --rpmlist.
       --recreate       : Like --del-image then --add-image, but preserves any
                          node associations.
                          Requires --image and --rpmlist
```

```
       --repos {list}   : A comma-seperated list of repositories to use.
       --use-existing   : register an already existing image, doesn't
                          require --rpmlist
       --image {image}  : Specify the image to operate on
       --rpmlist {path} : Provide the rpmlist to use when creating images
       --source {image} : Specify a source image to operate on (for clone)

--del-image             : delete the image, may use with --del-nodes
       --image {image}  : Specify the image to operate on

--show-images           : List images, BT 1 if root tarballs are desired

--show-nodes            : Show non-compute nodes (similar to mksimachine -L)

--update-image          : update packages in image to latest packages available
                          in repos, Requires --image
       --image {image}  : Specify the image to operate on

--refresh-image         : Refresh the given image to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --image {image}  : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packes to be sure are included

--yum-image             : Perform yum operations to supplied image, via yume
                          Requires --image, trailing arguments passed to yume
       --image {image}  : Specify the image to operate on

--update-node           : Update supplied node to latest pkgs avail in
                          repos, requires --node
       --node {node}    : Specify the node or nodes to operate on

--refresh-node          : Refresh the given node to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --node {node}    : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packes to be sure are included

--yum-node              : Perform yum operations to nodes, via yume.  Requires
                          --node.  Trailing arguments passed to yume
       --node {node}    : Specify the node or nodes to operate on
```

```
--assign-image          : Assign image to node.  Requires --node, --image
       --node {node}    : Specify the node or nodes to operate on
       --image {image}  : Specify the image to operate on

--next-boot             : Action to perform when the service node or leader
                          node next boots.
        {image|bt|disk}:  disk: The node should boot from disk
                          image: re-install the node the standard way
                          bt: re-install the node, make use of bt, requires
                          assgined image to be set up with bittorrent, see
                          --add-to-bt.
                          Requires --node
       --node {node}    : Specify the node or nodes to operate on

--add-to-bt             : Start creating root BT tarballs for this image
                          Note: Compute nodes images are added by default
       --image {image}  : Specify the image to operate on

--del-from-bt           : No longer update BT root tarballs for this image.
       --image {image}  : Specify the image to operate on

--refresh-bt            : Refresh the bittorrent tarball and torrent file
                          Requires --image
       --image {image}  : Specify the image to operate on

[root@river-admin ~]# clear
[root@river-admin ~]# cinstallman --help
cinstallman Usage:

cinstallman is a tool that manages:
 - image creation (as a wrapper to mksiimage)
 - node package updates (as a wrapper to yume)
 - image package updates (yume within a chroot to the image)

This is a convenience tool and not all operations for the commands that are
wrapped are provided.  The most common operations are collected here for
ease of use.

For operations that take the --node parameter, the node can be an aggregation
of nodes like cimage and cpower can take.  Depending on the situation,
```

```
non-managed or offline nodes are skipped.

The tool retrieves the registered repositories from crepo so that they
need not be specified on the command line.

Operations:
--help                  : print his usage message
--create-image          : create a new systemimager image
                          By default, requires --rpmlist and --image
                          Optional flags  below:
       --clone          : Clone existing image, requires --source, --image.
                          Doesn't require --rpmlist.
       --recreate       : Like --del-image then --add-image, but preserves any
                          node associations.
                          Requires --image and --rpmlist
       --repos {list}   : A comma-seperated list of repositories to use.
       --use-existing   : register an already existing image, doesn't
                          require --rpmlist
       --image {image}  : Specify the image to operate on
       --rpmlist {path} : Provide the rpmlist to use when creating images
       --source {image} : Specify a source image to operate on (for clone)

--del-image             : delete the image, may use with --del-nodes
       --image {image}  : Specify the image to operate on

--show-images           : List images, BT 1 if root tarballs are desired

--show-nodes            : Show non-compute nodes (similar to mksimachine -L)

--update-image          : update packages in image to latest packages available
                          in repos, Requires --image
       --image {image}  : Specify the image to operate on

--refresh-image         : Refresh the given image to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --image {image}  : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packes to be sure are included

--yum-image             : Perform yum operations to supplied image, via yume
                          Requires --image, trailing arguments passed to yume
```

```
       --image {image}  : Specify the image to operate on

--update-node           : Update supplied node to latest pkgs avail in
                          repos, requires --node
       --node {node}    : Specify the node or nodes to operate on

--refresh-node          : Refresh the given node to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --node {node}    : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packes to be sure are included

--yum-node              : Perform yum operations to nodes, via yume.  Requires
                          --node.  Trailing arguments passed to yume
       --node {node}    : Specify the node or nodes to operate on

--assign-image          : Assign image to node.  Requires --node, --image
       --node {node}    : Specify the node or nodes to operate on
       --image {image}  : Specify the image to operate on

--next-boot             : Action to perform when the service node or leader
                          node next boots.
        {image|bt|disk}:  disk: The node should boot from disk
                          image: re-install the node the standard way
                          bt: re-install the node, make use of bt, requires
                          assgined image to be set up with bittorrent, see
                          --add-to-bt.
                          Requires --node
       --node {node}    : Specify the node or nodes to operate on

--add-to-bt             : Start creating root BT tarballs for this image
                          Note: Compute nodes images are added by default
       --image {image}  : Specify the image to operate on

--del-from-bt           : No longer update BT root tarballs for this image.
       --image {image}  : Specify the image to operate on

--refresh-bt            : Refresh the bittorrent tarball and torrent file
                          Requires --image
       --image {image}  : Specify the image to operate on
```

In the following example, the `--refresh-node` operation is used to ensure the online managed service nodes include all the packages in the list. You could use this if you updated your `rpmlist` to include new packages or if you recently added new media with the `crepo` command and want running nodes to have the newly updated packages. A similar `--refresh-image` operation exists for images.

```
# cinstallman --refresh-node --node service\* --rpmlist
/etc/opt/sgi/rpmlists/service-sles11.rpmlist
```

Customizing Software On Your SGI ICE X System

This section discusses how to manage various nodes on your SGI ICE X system. It describes how to configure the various nodes, including the compute and service nodes. It describes how to augment software packages. Many tasks having to do with package management have multiple valid methods to use.

For information on installing patches and updates, see "Installing SMC for SGI ICE Patches and Updating SGI ICE Systems " on page 122.

Creating Compute Node Custom Images

You can add per-host compute node customization to the compute node images. You do this by adding scripts either to the `/opt/sgi/share/per-host-customization/global/` directory or the `/opt/sgi/share/per-host-customization/mynewimage/` directory on the system admin controller (SAC).

Note: When creating custom images for compute nodes, make sure you clone the original SGI images. This provides the original images intact that you can fall back to if necessary. The following example is based on SLES.

Scripts in the global directory apply to all compute nodes images. Scripts under the image name apply only to the image in question. The scripts are cycled through once per host when being installed on the rack leader controllers (RLCs). They receive one input argument, which is the full path (on the RLC) 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 SAC, as follows:

```
This directory contains compute node image customization scripts which are
executed as part of the install-image operations on the leader nodes when
```

```
pulling over a new compute node image.

After the image has been pulled over, and the per-host-customization dir has
been rsynced, the per-host /etc and /var directories are populated, then the
scripts in this directory are cycled through once per-host.  This allows the
scripts to source the node specific network and cluster management settings,
and set node specific settings.

Scripts in the global directory are iterated through first, then if a
directory exists that matches the image name, those scripts are iterated
through next.

You can use the scripts in the global directory as examples.
```

An example global script, `/opt/sgi/share/per-host-customization/global/sgi-fstab` is, as follows:

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation; either version 2 of the License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307 USA
#
# Set up the compute node's /etc/fstab file.
#
# Modify per your sites requirements.
#
# This script is excecuted once per-host as part of the install-image
operation
```

```
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

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

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n '/^compat
/s/^compat.*: //p') )
compatdistro=${os[0]}${os[1]}

if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then

 #
 # SLES 10 compatible
 #
 cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
EOF

elif [ ${compatdistro} = "rhel5"  ]; then

 #
 # RHEL 5 compatible
 #

 #
 # RHEL expects several subsys directories to be present under
/var/run
 # and /var/lock, hence no tmpfs mounts for them
 #
 cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
devpts          /dev/pts        devpts  gid=5,mode=620  0       0
EOF

else
```

```
 echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi
```

Modify Compute Image Kernel Boot Options

You can use the `cattr` command to set extra kernel boot parameters for compute nodes on a per-image basis. For example to append "`cgroup_disable=memory`" to kernel boot parameters for any node booting the "`compute-sles11sp1`" image, perform a command similar to the following:

```
% cattr set kernel_extra_params-compute-sles11sp1 cgroup_disable=memory
```

Push the image, as follows:

```
# cimage --push-rack mynewimage r1
```

Compute Node Per-Host Customization for Additional Network Interfaces

---

Note: The following example is only for systems running SLES.

Per compute-node customization may be useful for configuring additional network interfaces that are on some, but not all, compute nodes. An example of how to configure network interfaces on individual compute nodes is the `/opt/sgi/share/per-host-customization/mynewimage/mycustomization` script, that follows:

```
#Copyright (c) 2008 Silicon Graphics, Inc.
# All rights reserved.
#
# do node specific setup
#
# This script is excecuted once per-host as part of the install-image operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $ARGV[0],
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#
```

```
use lib "/usr/lib/systemconfig","/opt/sgi/share/per-host-customization/global";
use sanity;

sanity_checks();

$blade_path = $node = $ARGV[0];
$node =~ s/.*\///;

sub i0n4 {
        my $ifcfg="etc/sysconfig/network/ifcfg-eth2";
        open(IFCFG, ">$blade_path/$ifcfg") or
                die "$0: can't open $blade_path/$ifcfg";
        print IFCFG<<EOF
BOOTPROTO='static'
IPADDR='10.20.0.1'
NETMASK='255.255.0.0'
STARTMODE='onboot'
WIRELESS='no'
EOF
        ;
        close(IFCFG);
}

@nodes = ("i0n4");

foreach $n (@nodes) {
        if ( $n eq $node ) {
                eval $n;
        }
}
```

Pushing `mynewimage` to rack 1 causes the eth2 interface of compute node `r1i0n4` to be configured with IP address 10.20.0.1 when the node is brought up with `mynewimage`. Push the image, as follows:

```
# cimage --push-rack mynewimage r1
```

Customizing Software Images

Note: Procedures in this section describe how to work with service node and compute node images. Always use a cloned image. If you are adjusting an RPM list, use your own copy of the RPM list.

The service and compute node images are created during the `configure-cluster` operation (or during your upgrade from a prior release). This process uses an RPM list to generate a root on the fly.

You can clone a compute node image, or create a new one based on an RPM list. For service nodes, SGI does not support a clone operation. For compute images, you can either clone the image and work on a copy or you can always make a new compute node image from the SGI supplied default RPM list.

Procedure 4-3 Creating a Simple Compute Node Image Clone

Note: Always work from a clone image, see "Customizing Software Images" on page 166.

To create a simple compute node image clone from the system admin controller (SAC), perform the following steps:

1. To clone the compute node image, perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image compute-sles11-new
```

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

```
# cimage --list-images
image: compute-sles11
       kernel: 2.6.27.19-5-smp

image: compute-sles11-new
       kernel: 2.6.27.19-5-smp
```

3. To push the compute node image out to the rack, perform the following:

```
# cimage --push-rack compute-sles11-new r\*
```

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

   # **cimage --set compute-sles11-new 2.6.27.19-5-smp "r*i*n*"**

Procedure 4-4 Manually Adding a Package to a Compute Node Image

To manually add a package to a compute node image, perform the steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "`cinstallman` Command" on page 154.

---

1. Make a clone of the compute node image, as described in "Customizing Software Images" on page 166.

2.

   Note: This example shows SLES11.

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

   ```
   # cimage --list-images
     image: compute-sles11
            kernel: 2.6.27.19-5-smp

     image: compute-sles11-new
            kernel: 2.6.27.19-5-smp
   ```

3. From the system admin controller (SAC), change directory to the images directory, as follows:

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

4. From the SAC, copy the RPMs you wish to add, as follows, where `compute-sles11-new` is your own compute node image, as follows:

   # **cp /tmp/newrpm.rpm compute-sles11-new/tmp**

5. The new RPMs now reside in `/tmp` directory in the image named `compute-sles11-new`. To install them into your new compute node image, perform the following commands:

   ```
   # chroot compute-sles11-new bash
   ```

   And then perform the following:

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

   At this point, the image has been updated with the RPM.

6. The image on the SAC is updated. However, you still need to push the changes out. Ensure there are no nodes currently using the image and then run this command:

   ```
   # cimage --push-rack compute-sles11-new r\*
   ```

   This will push the updates to the rack lead controllers and the changes will be seen by the compute nodes the next time they start up. For information on how to ensure the image is associated with a given node, see the `cimage --set` command and the example in Procedure 4-3, page 166.

Procedure 4-5 Manually Adding a Package to the Service Node Image

To manually add a package to the service node image, perform the following steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "`cinstallman` Command" on page 154.

---

1. Use the `cinstallman` command to create your own version of the service node image. See "`cinstallman` Command" on page 154.

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

   ```
   # cd /var/lib/systemimager/images/
   ```

3. From the SAC, copy the RPMs you wish to add, as follows, where `my-service-image` is your own service node image:

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

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

   ```
   # chroot my-service-image bash
   ```

   And then perform the following:

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

   At this point, the image has been updated with the RPM. Please note, that unlike compute node images, changes made to a service node image will not be seen by service nodes until they are reinstalled with the image. If you wish to install the package on running systems, you can copy the RPM to the running system and use the RPM from there.

cimage Command

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

For a help statement, perform the following command:

```
admin:~ # cimage --help
cimage is a program for managing compute node root images in SMC for ICE.

Usage: cimage OPTION ...

Options
 --help                              Usage and help text.
 --debug                             Output additional debug information.
 --list-images                       List images and their kernels.
 --list-nodes NODE                   List node(s) and what they are set to.
 --set [OPTION] IMAGE KERNEL NODE    Set node(s) to image and kernel.
    --nfs                            Use NFS roots (default).
    --tmpfs                          Use tmpfs roots.
 --set-default [OPTION] IMAGE KERNEL    Set default image, kernel, rootfs type.
    --nfs                            Use NFS roots (default).
    --tmpfs                          Use tmpfs roots.
 --show-default                      Show default image, kernel, rootfs type.
 --add-db IMAGE                      Add image and its kernels to the db.
 --del-db IMAGE                      Delete image and its kernels from db.
```

```
 --update-db IMAGE                      Short-cut for --del-db, then --add-db.
 --push-rack [OPTIONS] IMAGE RACK       Push or update image on rack(s).
    --force                             Bypass the booted nodes check, deletes.
    --update-only                       Skip files newer in dest, no delete.
    --quiet                             Turn off diagnostic information.
 --del-rack IMAGE RACK                  Delete an image from rack(s).
 --clone-image OIMAGE NIMAGE            Clone an existing image to a new image.
 --del-image [OPTIONS] IMAGE            Delete an existing image entirely.
    --quiet                             Turn off diagnostic information.

RACK arguments take the format ’rX’
NODE arguments take the format ’rXiYnZ’
ROOTFS argument can be either ’nfs’ or ’tmpfs’

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

EXAMPLES

Example 4-1 `cimage` Command Examples

The following examples walk you through some typical `cimage` command operations.

To list the available images and their associated kernels, perform the following:

```
# cimage --list-images
image: compute-sles11
        kernel: 2.6.27.19-5-carlsbad
        kernel: 2.6.27.19-5-default
image: compute-sles11-1_7
        kernel: 2.6.27.19-5-default
```

To list the compute nodes in rack 1 and the image and kernel they are set to boot, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-default nfs
r1i0n8: compute-sles11 2.6.27.19-5-default nfs
```

The `cimage` command also shows the root filesystem type (nfs or tmpfs)

To set the r1i0n0 compute node to boot the 2.6.27.19-5-smp kernel from the compute-sles11 image, perform the following: :

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i0n0
```

To list the nodes in rack 1 to see the changes set in the example above, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-smp
r1i0n1: compute-sles11 2.6.27.19-5-smp
r1i0n2: compute-sles11 2.6.27.19-5-smp
[...snip...]
```

To set all nodes in all racks to boot the 2.6.27.19-5-smp kernel from the compute-sles11 image, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r*i*n*
```

To set two ranges of nodes to boot the 2.6.27.19-5-smp kernel, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the compute-sles11 image to a new image (so that you can modify it) , perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image mynewimage
Cloning compute-sles11 to mynewimage ... done
```

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

Note: If you have made changes to the compute node image and are pushing that image out to rack leader controllers (RLCs), it is a good practice to use the cinstallman --refresh-bt --image {*image*} command to refresh the bittorrent tarball and torrent file for a compute node image. This avoids duplication by rsync when the image is pushed out to the RLCs. For more information, see the cinstallman --h usage statement or "cinstallman Command" on page 154.

To change to the cloned image created in the example, above, copy the needed rpms into the /var/lib/systemimager/images/mynewimage/tmp directory, use the chroot command to enter the directory and then install the rpms, perform the following:

```
# cp *.rpm /var/lib/systemimager/images/mynewimage/tmp
# chroot /var/lib/systemimager/images/mynewimage/ bash
# rpm -Uvh /tmp/*.rpm
```

If you make changes to the kernels in the image, you need to refresh the kernel database entries for your image, To do this, perform the following:

```
# cimage --update-db mynewimage
```

If you did not make changes to the kernels in the cloned image created in the example above, you can omit this step.

To push new software images out to the compute blades in a rack or set of racks, perform the following:

```
# cimage --push-rack mynewimage r*
r1lead: install-image: mynewimage
r1lead: install-image: mynewimage done.
```

To list images in the database the kernels they contain, perform the following:

```
# cimage --list-images

image: compute-sles11
        kernel: 2.6.16.60-0.7-carlsbad
        kernel: 2.6.16.60-0.7-smp

image: mynewimage
        kernel: 2.6.16.60-0.7-carlsbad
        kernel: 2.6.16.60-0.7-smp
```

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

```
# cimage --set mynewimage 2.6.16.60-0.7-smp r1i3n*
```

You need to reboot the compute nodes to run the new images.

Completely remove an image you no longer use, both from system admin controller (SAC) and all compute nodes in all racks, perform the following:

```
# cimage --del-image mynewimage
r1lead: delete-image: mynewimage
```

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

Using `cinstallman` to Install Packages into Software Images

The packages that make up SMC for SGI ICE X, SGI Foundation, and the Linux distribution media, and any other media or custom repositories you have added reside in repositories. The `cinstallman` command looks up the list of all repositories and provides that list to the commands it calls out for its operation such as `yume`.

Note: Always work with copies of software images.

The `cinstallman` command can update packages within `systemimager` images. You may also use `cinstallman` to install a single package within an image.

However, `cinstallman` and the commands it calls only works with the configured repositories. So if you are installing your own RPM, you will need that package to be part of an existing repository. You may use the `crepo` command to create a custom repository into which you can collect custom packages.

Note: The `yum` command maintains a cache of the package metadata. If you just recently changed the repositories, `yum` caches for the nodes or images you are working with may be out of date. In that case, you can issue the `yum` command "`clean all`" with `--yum-node` and `--yum-image`. The `cinstallman` command `--update-node` and `--update-image` options do this for you.

The following example shows how to install the `zlib-devel` package in to the service node image so that the next time you image or install a service node, it will have this new package.

```
# cinstallman --yum-image --image my-service-sles11 install zlib-devel
```

You can perform a similar operation for compute node images. Note the following:

- If you update a compute node image on the system admin controller (SAC), you have to use the `cimage` command to push the changes. For more information on the `cimage` command, see "`cimage` Command" on page 169.
- If you update a service node image on the SAC, that service node needs to be reinstalled and/or reimaged to get the change. The `discover` command can be given an alternate image or you may use the `cinstallman --assign-image`

command followed by the `cinstallman --next-boot` command to direct the service node to reimage itself with a specified image the next time it boots.

Using `yum` to Install Packages on Running Service or Rack Leader Controllers (RLCs)

Note: These instructions only apply to managed service nodes and RLCs. They do not apply to compute nodes.

You can use the `yum` command to install a package on a service node. From the system admin controller (SAC), you can issue a command similar to the following:

```
# cinstallman --yum-node --node service0 install zlib-devel
```

Note: To get all service nodes, replace `service0` with `service\*`.

For more information on the `cinstallman` command, see "`cinstallman` Command" on page 154.

Creating Compute and Service Node Images Using the `cinstallman` Command

You can create service node and compute node images using the `cinstallman` command. This generates a root directory for images, automatically.

Fresh installations of SMC for SGI ICE X create these images during the `configure-cluster` installation step (see "Installing SMC for ICE X System Admin Controller (SAC) Software").

The RPM lists that drive which packages get installed in the images are listed in files located in `/etc/opt/sgi/rpmlists`. For example, `/etc/opt/sgi/rpmlists/compute-sles11.rpmlist` (see "`crepo` Command" on page 150). You should **NOT** edit the default lists. These default files are recreated by the `crepo` command when repositories are added or removed. Therefore, you should only use the default RPM lists as a model for your own.

Note: The procedure below uses SLES.

Procedure 4-6 Using the `cinstallman` Command to Create a Service Node Image:

To create a service node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the example service node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/service-sles11.rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.

3. Use the `cinstallman` command with the `--create-image` option to create the images root directory, as follows:

```
# cinstallman --create-image --image my-service-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

   This example uses `my-service-node-image` as the home/name of the image.

   Output is logged to `/var/log/cinstallman` on the system admin controller (SAC).

4. After the `cinstallman` comand finishes, the image is ready to be used with service nodes. You can supply this image as an optional image name to the `discover` command, or you may assign an existing service node to this image using the `cinstallman --assign-image` command. You can tell a service node to image itself next reboot by using `the cinstallman --next-boot` option.

Procedure 4-7 Use the `cinstallman` Command to Create a Compute Node Image

To create a compute node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the compute node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/compute-sles11.rpmlist
/etc/opt/sgi/rpmlists/my-compute-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.

3. Run the `cinstallman` command to create the root, as follows:

```
# cinstallman --create-image --image my-compute-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-compute-node.rpmlist
```

   This example uses the name name `my-compute-node-image` as the name.

   Output is logged to `/var/log/cinstallman` on the SAC.

   The `cinstallman` command makes the new image available to the `cimage` command.

4. For information on how to use the `cimage` command to push this new image to rack leader controllers (RLCs), see "cimage Command" on page 169.

Installing a Service Node with a Non-default Image

If you have a non-default service node image you wish to install on a service node, you have two choices, as follows:

- Specify the image name when you first discover the node with the `discover` command.
- Use the `cinstallman` command to associate an image with a service node, then set up the node to reinstall itself the next time it boots.

The following example shows how to associate a custom image at discover time:

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

The next example shows how to reinstall an already discovered service node with a new image:

```
# cinstallman --assign-image --node service2 --image my-service-node-image
# cinstallman --next-boot image --node service2
```

When you reboot the node, it will reinstall itself.

For more information on the `discover` command, see "discover Command" on page 84. For more information on the `cinstallman` command, see "cinstallman Command" on page 154.

Retrieving a Service Node Image from a Running Service Node

To retrieve a service node image from a running service node, perform the following steps:

1. As **root user**, log into the service node from which you wish to retrieve an image. You can use the `si_prepareclient`(8) program to extract an image. Start the program, as follows:

```
service0:~ # si_prepareclient --server admin
Welcome to the SystemImager si_prepareclient command. This command may modify
the following files to prepare your golden client for having its image
retrieved by the imageserver.  It will also create the /etc/systemimager
directory and fill it with information about your golden client. All modified
files will be backed up with the .before_systemimager-3.8.0 extension.

 /etc/services:
   This file defines the port numbers used by certain software on your system.
   Entries for rsync will be added if necessary.

 /tmp/filetlOeP5:
   This is a temporary configuration file that rsync needs on your golden client
   in order to make your filesystem available to your SystemImager server.

 inetd configuration:
   SystemImager needs to run rsync as a standalone daemon on your golden client
   until its image is retrieved by your SystemImager server.  If rsyncd is
   configured to run as a service started by inetd, it will be temporarily
   disabled, and any running rsync daemons or commands will be stopped.  Then,
   an rsync daemon will be started using the temporary configuration file
   mentioned above.

See "si_prepareclient --help" for command line options.

Continue? (y/[n]):
```

Enter **y** to continue. After a few moments, you are returned to the command prompt. You are now ready to retrieve the image from the system admin controller (SAC).

2. Exit the **service0** node, and as **root user** on the SAC, perform the following command: (Replace the image name and service node name, as needed.)

   ```
   admin # mksiimage --Get --client service0 --name myimage
   ```

   It now retrieves the image. No progress information is provided. It takes several minutes depending on the size of the image on the service node.

3. Use the `cinstallman` command to register the newly collected image:

   ```
   admin # cinstallman --create --use-existing --image myimage
   ```

4. If you want to discover a node using this image directly, you can use the `discover` command, as follows:

   ```
   admin # discover --service 0,image=myimage
   ```

5. If you want to re-image an already discovered node with your new image, run the following commands:

   ```
   # cinstallman --assign-image --node service0 --image myimag
   # cinstallman --next-boot image --node service0
   ```

6. Reboot the service node.

Using a Custom Repository for Site Packages

This section describes how to maintain packages specific to your site and have them available to the `crepo` command (see "`crepo` Command" on page 150).

SGI suggests putting site-specific packages in a separate location. They should not reside in the same location as SGI or Novell supplied packages.

Procedure 4-8 Setting Up a Custom Repository for Site Packages

To set up a custom repository for your custom packages, perform the following steps:

1. Create directory for your site-specific packages on the system admin controller (SAC), as follows:

   ```
   # mkdir -p /tftpboot/site-local/sles-10-x86_64
   ```

2. Copy your site packages in to the new directory, as follows:

   ```
   # cp my-package-1.5.x86_64.rpm /tftpboot/site-local/sles-10-x86_64
   ```

3. Register your custom repository using the `crepo` command. This command will ensure your repository is consulted when the `cinstallman` command performs its operations. This command also creates the necessary `yum/repomd` metadata.

   ```
   # crepo --add /tftpboot/site-local/sles-10-x86_64 --custom my-repo
   ```

   Your new repository may be consulted by `cinstallman` command operations going forward including updating images, nodes, and creating images.

4. If you wish this repository to be used by `cinstallman` by default, you need to select it. Use the following command:

   ```
   # crepo --select my-repo
   ```

5. If you use `cinstallman` to create an image, you will want to add your custom package to the rpmlist you use with the cinstallman command (see "Using `cinstallman` to Install Packages into Software Images" on page 173).

SGI ICE X System Configuration Framework

All node types that are part of an SGI ICE X system can have configuration settings adjusted by the configuration framework. There is some overlap between the per-host customization instructions and the configuration framework instructions. Each approach plays a role in configuring your system. The major differences between the two methods are, as follows:

- Per-host customization runs at the time an image is pushed to the rack leader controllers (RLCs).
- Per-host customization only applies to compute node images.
- The SGI ICE system configuration framework can be used with all node types.
- The system configuration framework is run when a new root is created, when `SuSEconfig` command is run for some other reason, as part of a `yum` operation, or when new compute images are pushed with the `cimage` command.

This framework exists to make it easy to adjust configuration items. There are SGI-supplied scripts already present. You can add more scripts as you wish. You can also exclude scripts from running without purging the script if you decide a certain script should not be run. The following set of questions in bold and bulleted answers describes how to use the system configuration framework.

How does the system configuration framwork operate?

These files could be added, for example, to a running service node, or to an already created service or compute image. Remember that images destined for compute nodes need to be pushed with the `cimage` command after being altered. For more information, see "`cimage` Command" on page 169.

- A `/opt/sgi/lib/cluster-configuration` script is called, from where it is called is described below.
- That script iterates through scripts residing in `/etc/opt/sgi/conf.d`.
- Any scripts listed in **`/etc/opt/sgi/conf.d/exclude`** are skipped, as are scripts, that are not executable.
- Scripts in system configruation framework **must** be tolerant of files that do not exist yet, as described below. For example, check that a `syslog` configuration file exists before trying to adjust it.
- Scripts ending in a `distro` name, or a distro name with a specific distro version are only run if the node in question is running that `distro`. For example, `/etc/opt/sgi/conf.d/99-foo.sles` would only run if the node was running `sles`. This example shows the precedence of operations:If you had `88-myscript.sles10`, `88-myscript.sles`, and `88-myscript`
  - On a `sles10` system, `88-myscript.sles10` would execute
  - On a `sles` system that is not `sles10`, `88-myscript.sles` would execute
  - On all other distros, `88-myscript` would would execute
- If you wish to make a custom version of an script supplied by SGI, you may simply name it with "`.local`" and the local version will run in place of the one supplied by SGI. This allows for customization without modifying scripts supplied by SGI. Scripts ending in `.local` have the highest precedence. In other words, if you had `88-myscript.sles`, and `88-myscript.local`, then `88-myscript.local` would execute in all cases and the other `88-myscript` scripts would never execute.

From where is the framework called?

- The callout for `/opt/sgi/lib/cluster-configuration` is implemented as a `yum` plugin that executes after packages have been installed and cleaned.
- On SLES only, there is also a SUSE configuration script in the `/sbin/conf.d` directory, called `SuSEconfig.00cluster-configuration`, that calls the framework. This is in case of you are using YaST to install or upgrade packages.

- On SLES only, one of the scripts called by the framework calls `SuSEconfig`. A check is made to avoid a callout loop.
- The framework is also called when the system admin conroller (SAC), RLC, or service nodes start up. The call is made just after networking is configured. As a site administrator, you could create custom scripts here that check on or perform certain configuration operations.
- When using the `cimage` command to push a compute node root image to RLCs, the configuration framework executes within the `chroot` of the compute node image after it is pulled from the SAC to the RLC.

How do I adjust my system configuration?

- Create a small script in `/etc/opt/sgi/conf.d` to do the adjustment.

  Be sure that you test for existence of files and do not assume they are there (see "Why do scripts need to tolerate files that do not exist but should?" below).

Why do scripts need to tolerate files that do not exist but should?

- This is because the `mksiimage` command runs `yume` and `yum` in two steps. The first step only installs 40 or so RPMs but our framework is called then too. The second pass installs the other "hundreds" of RPMs. So the framework is called once before many packages are installed, and again after everything is in place. So not all files you expect might be available when your small script is called.

How does the yum plugin work?

- In order for the `yum` plugin to work, the `/etc/yum.conf` file has to have `plugins=1` set in its configuration file. SMC for SGI ICE X software ensures that is in place by way of a trigger in the `sgi-cluster` package. Any time `yum` is installed or updated, it verify `plugins=1` is set.

How does yume work?

- `yume`, an oscar wrapper for `yum`, works by creating a temporary `yum` configuration file in `/tmp` and then points `yum` at it. This temporary configuration file needs to have plugins enabled. A tiny patch to `yume` makes this happen. This fixes it for `yume` and also `mksiimage`, which calls `yume` as part of its operation.

Cluster Configuration Repository: Updates on Demand

SMC for ICE X contains a cluster configuration repository/update framework. This framework generates and distributes configuration updates to system admin controller (SAC), rack leader controller (RLC), and service nodes in the cluster. Some of the configuration files managed by this framework include C3 conserver, DNS, Ganglia, hosts files, and NTP.

When an event occurs that requires these files to be updated, the framework executes on the SAC. The SAC stores the updated configuration framework in a special cached location and updates the appropriate nodes with their new configuration files.

In addition to the updates happening as required, the configuration file repository is consulted when a SAC, RLC, or service node boots. This happens shortly after networking is started. Any configuration files that are new or updated are transferred at this early stage so that the node is fully configured by the time the node is fully operational.

There are no hooks for customer configuration in the configuration repository at this time.

This update framework is tied in with the `/etc/opt/sgi/conf.d` configuration framework to provide a full configuration solution. As mentioned earlier, customers are encouraged to create `/etc/opt/sgi/conf.d` scripts to do cluster configuration.

cnodes Command

The `cnodes` command provides information about the types of nodes in your system. For help information, perform the following:

```
[admin ~]# cnodes --help

Usage: cnodes [OPTIONS]

Options:
 --all                  all compute, leader and service nodes, and switches
 --compute              all compute nodes
 --leader               all leader nodes
 --service              all service nodes
 --ibswitch             all ib switch nodes
 --mgmtswitch           all cluster management switches
 --switch-blade         all switch blade nodes
```

```
 --cmc                  all CMCs
 --online               modifier: nodes marked online
 --offline              modifier: nodes marked offline
 --managed              modifier: managed nodes
 --unmanaged            modifier: unmanaged nodes
 --temponames           modifier: return Tempo node names instead of hostnames
 --rack=RACK            modifier: only match nodes related to RACK

Note: default modifiers are 'online' and 'managed' unless otherwise specified.
```

EXAMPLES

Example 4-2 `cnodes` Example

The following examples walk you through some typical `cnodes` command operations.

To see a list of all nodes in your system, perform the following:

```
[admin ~]# cnodes --all
r1i0n0
r1i0n1
r1lead
service0
```

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

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

To see a list of service nodes, perform the following:

```
[admin ~]#  cnodes --service
service0
```

Power Management Commands

The `cpower` command allows you to power up, power down, reset, and show the power status of system components.

`cpower` Command

The `cpower` command is, as follows:

`cpower [<`*option*`> ...] [<`*target_type*`>] [<`*action*`>] <`*target*`>`

The <*option*> argument can be one or more of the following:

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

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

`--node`	Applies the action to nodes. Nodes are compute nodes, RLCs, system admin controllers (SACs), and service nodes. [default]
`--iru`	Applies the action at the IRU level (now referred to as a blade enclosure pair).
`--rack`	Applies the action at the rack level.
`--system`	Applies the action to the system. You must **not** specify a target with this type.

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

`--status`	Show the power status of the target, including whether it is booted or not. [default]
`--up | --on`	Powers up the target.
`--down | --off`	Powers down the target.
`--reset`	Performs a hard reset on the target.

`--cycle`	Power cycles the target.
`--boot`	Boots up the target, unless it is already booted. Waits for all targets to boot.
`--reboot`	Reboots the target, even if already booted. Wait for all targets to boot.
`--halt`	Halts and then powers off the target.
`--shutdown`	Shuts down the target, but does not power it off. Waits for targets to shut down.
`--identify` <*interval*>	Turns on the identifying LED for the specified interval in seconds. Uses an interval of 0 to turn off immediately.
`-h, --help`	Shows help usage statement.

The target must always be specified except when the `--system` option is used. Wildcards may be used, but be careful **not** to accidentally power off or reboot the RLCs. If wildcard use affects any RLC, 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, rack leader controllers (RLCs), or service nodes. If you do **not** specify the `--iru`, `--rack`, or `--system` option, the command defaults to operating as if you had specified `--node`. Individual rack units are now called blade enclosure pairs but the command syntax is the same.

Here are examples of node target names:

- `r1i3n10`

  Compute node at rack 1, IRU 3, slot 10

- `service0`

  Service node 0

- `r3lead`

  RLC for rack 3

- `r1i*n*`

Wildcards let you specify ranges of nodes, for example, `r1i*n*` all compute nodes in all IRUs on rack 1

IPMI-style Commands

The default operation for the `cpower` command is to operate on nodes and to provide you the status of these nodes, as follows:

```
# cpower r1i*n*
```

This command is equivalent to the following:

```
# cpower --node --status r1i*n*
```

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

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

The default is to apply to a node.

The following commands behave exactly as you would expect as if you were using `ipmitool`, and have no special extra logic for ordering:

```
# cpower --up r1i*n*
```

```
# cpower --reset r1i*n*
```

```
# cpower --cycle r1i*n*
```

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

Note: `--up` is a synonym for `--on` and `--down` is a synonym for `--off`.

IRU, Rack, and System Domains

The `cpower` command contains more logic when you go up to higher levels of abstraction, for example, when using the `--iru`, `--rack`, and `--system` options. 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.

Note: Individual rack units (IRUs) are now called blade enclosure pairs. The command syntax works the same, as in previous releases.

The `--iru` option tells the command to use correct ordering with IRU power commands. In this case, it firsts connect to the CMC on each IRU in rack 1 to issue the `power on` command, which turns on power to the IRU chassis (this is not the equivalent `ipmitool` command). Then it powers up the compute nodes in the IRU. Powering things down is the opposite, with the power to the IRU being turned off after power to the blades. IRU targets are specified as follows:`r3i2` for rack 3, IRU 2.

```
# cpower --iru --up r1i*
```

The `--rack` option ensures power commands to the rack leader controller (RLC) are down in the correct order relative to compute nodes within a rack. First, it powers up the RLC 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 RLC (after doing a shutdown) after all the IRUs are powered down. To avoid including RLCs 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 RLCs, then to IRUs and compute blades, in just the same way. Likewise, compute blades are powered down before IRUs, RLCs, and service nodes, in that order. To avoid including service nodes in a system-domain command, use the **`--noservice`** option. Note that you must not specify a target with `--system` option, since it applies to the SGI ICE system.

Shutting Down and Booting

Note: The `--shutdown --off` combination of actions were deprecated in a previous release. Use the `--halt` option in its place.

It is useful to be able to shutdown a machine before turning off the power, in most cases. The following `cpower` options to enable you to do this: `--halt`, `--boot`, and `--reboot`. The `--halt` option allows you to shut down a node. The `--reboot` option ensures that a system is always rebooted, whereas --boot will only boot up a system if it is not already booted. Thus, --boot is useful for booting up compute blades that have failed to start.

You need to configure the order in which service nodes are booted up and shut down as part of the overall system power management process. This is done by setting a boot_order for each service node. Use the cadmin command to set the boot order for a service node, for example:

```
# cadmin --set-boot-order --node service0 2
```

The cpower --system --boot command boots up service nodes with a lower boot order, first. It then boots up service nodes with a higher boot order. The reverse is true when shutting down the system with cpower. For example, if service1 has a boot order of 3 and service2 has a boot order of 5, service1 is booted completely, and then service2 is booted, afterwards. During shutdown, service2 is shut down completely before service1 is shutdown.

There is a special meaning to a service node having a boot order of zero. This value causes the cpower --system command to skip that service node completely for both start up and shutdown (although not for status queries). Negative values for the service node boot order setting are not permitted.

Note: The IPMI power commands necessary to enable a system to boot (either with a power reset, or a power on) may be sent to a node. The --halt option, halts the target node and then powers it off.

The --halt options works on node, IRU, or rack domain levels. It will shut down nodes (in the correct order if you use the --iru or --rack options), and then just leave them as they are, power still applied. Using both these actions results in nodes being halted, then powered off. This is particularly useful when powering off a rack, since otherwise, the rack leader controllers (RLCs) may be shut down before there is a chance to power off the compute blades. Here is an example:

```
# cpower --halt --rack r1
```

To boot up systems that have not already been booted, perform the following:

```
# cpower --boot  r1i2n*
```

Again, the command boots up nodes in the right orders if you specify the --iru or --rack options and the appropriate target. Otherwise, there is no guarantee that, for example, the command will attempt to power on the RLC 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 4-3 `cpower` Command Examples

To boot compute blade `r1i0n8`, perform the following:

```
# cpower --boot r1i0n8
```

To boot a number of compute blades at the same time, perform the following:

```
# cpower --boot --rack r1
```

Note: The `--boot` option will only boot those nodes that have not already booted.

To shut down service node 0, perform the following:

```
# cpower --halt service0
```

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

```
# cpower --halt --rack r3
```

Note: This command will shutdown and then power off all of the computer nodes in parallel, then shutdown and power off the RLC. Use the `--noleader` option if you want the RLC to remain booted up.

To shutdown the entire system, including all service nodes and all RLCs, but not the system admin controller (SAC), and not turn the power off to anything, perform the following:

```
# cpower --halt --system
```

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

```
# cpower --halt --system --noleader --noservice
```

Note: The only way to shut down the SAC is to perform the operation manually.

C3 Commands

Note: For legacy SGI ICE systems, this section remains intact. However, SGI recommends you use the `pdsh` and `pdcp` utilities described in "`pdsh` and `pdcp` Utilities" on page 194.

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

Note: The SMC for SGI ICE X version of C3 does not include the `cshutdown` and `cpushimage` commands.

The C3 commands used on the the SGI Alitx ICE 8200 system are, as follows:

C3 Utilities	Description
`cexec`(s)	Executes a given command string on each node of a cluster
`cget`	Retrieves a specified file from each node of a cluster and places it into the specified target directory
`ckill`	Runs `kill` on each node of a cluster for a specified process name
`clist`	Lists the names and types of clusters in the cluster configuration file
`cnum`	Returns the node names specified by the range specified on the command line
`cname`	Returns the node positions specified by the node name given on the command line

`cpush` Pushes files from the local machine to the nodes in your cluster

`cexec` is the most useful C3 utility. Use the `cpower` command rather than `cshutdown` (see "Power Management Commands" on page 183).

EXAMPLES

Example 4-4 C3 Command General Examples

The following examples walk you through some typical C3 command operations.

You can use the `cname` and `cnum` commands to map names to locations and vice versa, as follows:

```
# cname rack_1:0-2
local name for cluster:  rack_1
nodes from cluster:  rack_1
cluster:  rack_1 ; node name:  r1i0n0
cluster:  rack_1 ; node name:  r1i0n1
cluster:  rack_1 ; node name:  r1i0n10

# cnum rack_1: r1i0n0
local name for cluster:  rack_1
nodes from cluster:  rack_1
r1i0n0 is at index 0 in cluster rack_1

# cnum rack_1: r1i0n1
local name for cluster:  rack_1
nodes from cluster:  rack_1
```

You can use the `clist` command to retrieve the number of racks, as follows:

```
# clist
cluster  rack_1  is an indirect remote cluster
cluster  rack_2  is an indirect remote cluster
cluster  rack_3  is an indirect remote cluster
cluster  rack_4  is an indirect remote cluster
```

You can use the `cexec` command to view the addressing scheme of the C3 utility, as follows:

```
# cexec rack_1:1 hostname
************************* rack_1 *************************
```

```
************************ rack_1 ************************
--------- r1i0n1---------
r1i0n1

# cexec rack_1:2-3 rack_4:0-3,10 hostname
************************ rack_1 ************************
************************ rack_1 ************************
--------- r1i0n10---------
r1i0n10
--------- r1i0n11---------
r1i0n11
************************ rack_4 ************************
************************ rack_4 ************************
--------- r4i0n0---------
r4i0n0
--------- r4i0n1---------
r4i0n1
--------- r4i0n10---------
r4i0n10
--------- r4i0n11---------
r4i0n11
--------- r4i0n4---------
r4i0n4
```

The following set of command shows how to use the C3 commands to transverse the different levels of hierarchy in your SGI ICE X system (for information on the hierarchical design of your SGI ICE X system see "Basic System Building Blocks" on page 1).

To execute a C3 command on all blades within the default SGI ICE X system, for example, rack 1, perform the following:

```
# cexec hostname
************************ rack_1 ************************
************************ rack_1 ************************
--------- r1i0n0---------
r1i0n0
--------- r1i0n1---------
r1i0n1
--------- r1i0n10---------
r1i0n10
```

```
--------- r1i0n11---------
r1i0n11

...
```

To run a C3 command on all compute nodes across an SGI ICE X 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 (RLC), in the first rack, perform the following:

```
# cexec --head hostname
************************ rack_1 ************************
--------- rack_1---------
r1lead
```

To run a C3 command against all RLCs, across all racks, perform the following:

```
# cexec --head --all hostname
************************ rack_1 ************************
--------- rack_1---------
r1lead
************************ rack_2 ************************
--------- rack_2---------
r2lead
************************ rack_3 ************************
```

```
--------- rack_3---------
r3lead
************************ rack_4 ************************
--------- rack_4---------
r4lead
```

The following set of examples shows some specific case uses for the C3 commands that you are likely to employ.

Example 4-5 C3 Command Specific Use Examples

From the system admin controller (SAC), run command on rack 1 without including the RLC, 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 RLCs, as follows:

cexec --all --head <*cmd*>

Run a command on blade 42 (compute node 42) in rack 2, as follows:

cexec rack_2:42 <*cmd*>

From a **service node** over the InfiniBand Fabric, run a command on all blades (compute nodes) in the system, as follows:

cexec --all <*cmd*>

Run a command on blade 42 (compute node 42), as follows:

cexec blades:42 <*cmd*>

pdsh and pdcp Utilities

The pdsh(1) command is the parallel shell utility. The pdcp(1) command is the parallel copy/fetch utility. The SMC for SGI ICE X software populates some dshgroups files for the various node types. On the system admin controller (SAC), SMC for SGI ICE X software populates the `leader` and `service` groups files, which contain the list of online nodes in each of those groups.

On the rack leader controller (RLC), software populates the `compute` group for all the online compute nodes in that group.

On the service node, software populates the `compute` group which contains all the online compute nodes in the whole system.

For more information, see the `pdsh`(1) and `pdcp`(1) man pages.

EXAMPLES

From the SAC, to run the `hostname` command on all the RLCs, perform the following:

```
# pdsh -g leader hostname
```

To run the `hostname` command on all the compute nodes in the system, via the RLCs, perform the following:

```
# pdsh -g leader pdsh -g compute hostname
```

To run the `hostname` command on just `r1lead` and `r2lead`, perform the following:

```
# pdsh -w r1lead,r2lead hostname
```

cadmin: SMC for SGI ICE X Administrative Interface

The `cadmin` command allows you to change certain administrative parameters in the cluster such as the boot order of service nodes, the administrative status of nodes, and the adding, changing, and removal of IP addresses associated with service nodes.

To get the `cadmin` usage statement, perform the following:

```
[sys-admin ~]# cadmin --h
cadmin: SGI Tempo Administrative Interface
Help:

In general, these commands operate on {node}.  {node} is the Tempo style
node name.  For example, service0, r1lead, r1i0n0.  Even when the host name
for a service node is changed, the Tempo name for that node may still be used
for {node} below.  The node name can either be the tempo unique node name
or a customer-supplied host name associated with a tempo unique node name.
```

```
--version : Display current release information
--set-admin-status --node {node} {value}  : Set Administrative Status
--show-admin-status --node {node} : Show Administrative Status
--set-boot-order --node {node} [value] : Set boot order [1]
--show-boot-order --node {node} : Show boot order [1]
--set-ip --node {node} --net {net} {hostname}={ip} : Change an allocated ip [1]
--del-ip --node {node} --net {net} {hostname}={ip} : Delete an ip [1]
--add-ip --node {node} --net {net} {hostname}={ip} : allocate a new ip [1]
--show-ips --node {node} : Show all allocated IPs associated with node
--set-hostname --node {node} {new-hostname} : change the host name [5]
--show-hostname --node {node} : show the current host name for ice node {node}
--set-subdomain {domain} : Set the cluster subdomain [3]
--show-subdomain : Show the cluster subdomain
--set-admin-domain {domain} : Set the admin node house network domain
--show-admin-domain : Show the admin node house network domain
--db-purge --node {node} : Purge service or lead node (incl entire rack) from DB
--set-external-dns --ip {ip} : Set IP addr(s) of external DNS master(s) [4]
--show-external-dns : Show the IP addr(s) of the external DNS master(s)
--del-external-dns : Delete the configuration of external DNS master(s)
--show-root-labels : Show grub root labels if multiple roots are in use
--set-root-label --slot {#} --label {label} : Set changeable part of root label
--show-default-root : Show default root if multiple roots are in use
--set-default-root --slot {#} : Set the default slot if multiple roots in use
--show-current-root : Show current root slot
--enable-auto-recovery : Enable ability for nodes to recover themselves [6]
--disable-auto-recovery : Disable auto recovery [6]
--show-auto-recovery : Show the current state of node auto recovery [6]
--enable-redundant-mgmt-network --node {node}: Enable network
    management redundancy
--disable-redundant-mgmt-network --node {node}: Disable management network
    redundancy
--show-redundant-mgmt-network --node {node}:  Show current value.
--show-dhcp-option: Show admin dhcp option code used to distinguish mgmt network
--set-dhcp-option {value}: Set admin dhcp option code
--enable-switch-mgmt-network --node {node}: Enable switch management network
    for a node that is connected to managed top level switches. Not for ICE
    8200/8400 nodes. [7]
--disable-switch-mgmt-network --node {node}: Disable switch management network
    for the specified node.
--show-switch-mgmt-network --node {node}:  Show current value.
```

```
--enable-replication --node {node}: Enable MySQL Replication on the specified
  admin, leader, or service node. [8]
--disable-replication --node {node}: Disable MySQL Replication on the specified
  admin, leader, or service node.
--show-replication-status --node {node}:  Show current value.
--set-mgmt-bonding --node {node} {value}: "802.3ad" or "active-backup"
   Must be "active-backup" ICE 8200/8400 systems. Depends on mgmt switch.
--show-mgmt-bonding --node {node}:  Show current value.
--set-max-rack-irus --node {node} {value}: Max IRUs/CMCs for this leader node.
   Saves push-rack time by not doing expanisons for nodes that do not exist.
--show-max-rack-irus --node {node}:  Show current value
--set-blademond-rescan --node {node} {value}: rescan interval for this leader.
   Configures how many seconds blademond waits between checks for blade changes.
--show-blademond-rescan --node {node}:  Show current value

Node-attribute options:
--add-attribute [--string-data "{string}"] [--int-data {int}] {attribute-name}
--is-attribute {attribute-name}
--delete-attribute {attribute-name}
--set-attribute-data [--string-data "{string}"] [--int-data {int}]
  {attribute-name}
--get-attribute-data {attribute-name}
--search-attributes [--string-data "{string|regex}"] [--int-data {int}]
--add-node-attribute [--string-data "{string}"] [--int-data {int}]
  --node {node} --attribute {attribute-name}
--is-node-attribute --node {node} --attribute {attribute-name}
--delete-node-attribute --node {node} --attribute {attribute-name}
--set-node-attribute-data [--string-data "{string}"] [--int-data {int}]
  --node {node} --attribute {attribute-name}
--get-node-attribute-data --node {node} --attribute {attribute-name}
--search-node-attributes [--node {node}] [--attribute {attribute-name}]
  [--string-data "{string|regex}"] [--int-data {int}]

Descriptions of Selected Values:
{hostname}={ip} means specify the host name associated with the specified
ip address.
{net} is the tempo network to change such as ib-0, ib-1, head, gbe, bmc, etc
{node} is a tempo-style node name such as r1lead, service0, or r1i0n0.
[1] Only applies to service/ibswitch/mgmtswitch nodes
[2] This operation may require the cluster to be fully shut down and AC power
    to be removed.  IPs will have to be re-allocated to fit in the new range.
```

```
[3] All cluster nodes will have to be reset. Compute images need to be pushed
    again with the cimage command.
[4] Use quoted, semi-colon separated list if more than one master
[5] Only applies to admin and service/ibswitch/mgmtswitch nodes
[6] Auto recovery will allow service and leader nodes to boot in to a special
    recovery mode if the cluster doesn't recognize them.  This is enabled by
    default and would be used, for example, if a node's main board was replaced
    but the original system disks were imported from the original system.
[7] The global value for this is automatically detected and adjustable using
    configure-cluster. This per-node value is used for systems that mix the
    older cascaded CMC management network found in ICE 8200/8400 with the newer
    top level switch management network. In a mixed system, all leader and
    managed service nodes in the 8200/8400 part of the system should have this
    set to no. This can also be specified using the "discover" command.
[8] Node should be admin, leader, or a service node. Use configure-cluster to
    globally configure MySQL Replication on the whole system. If MySQL
    Replication is disabled on the admin node, replication will be off for the
    whole system, even if some nodes may have the replication attribute set to
    "yes". Use this feature with caution, if replication is enabled on a leader
    or service node, but disabled on the admin node, the leader or service node
    will not be able to get the information from the admin node, and the
    database calls will fail or return out of date information. To enable MySQL
    Replication on a limited number of node, disable it globally from within
    configure-cluster, then use cadmin to enable it on the admin node, and then
    on the other nodes.
```

EXAMPLES

Example 4-6 SMC for SGI ICE X Administrative Interface (`cadmin`) Command

Set a node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

Set a node online, as follows:

```
# cadmin --set-admin-status --node r1i0n0 online
```

Set the boot order for a service node, as follows:

```
# cadmin --set-boot-order --node service0 2
```

Add an IP to an existing service node, as follows:

```
# cadmin --add-ip --node service0 --net ib-0 my-new-ib0-ip=10.148.0.200
```

Change the SMC for SGI ICE X needed `service0-ib0` IP address, as follows:

```
# cadmin --set-ip --node service0 --net head service0=172.23.0.199
```

Show currently allocated IP addresses for `service0`, as follows:

```
# cadmin --show-ips --node service0
IP Address Information for SMC for SGI ICE X node: service0

ifname        ip               Network

myservice-bmc 172.24.0.3       head-bmc
myservice     172.23.0.3       head
myservice-ib0 10.148.0.254     ib-0
myservice-ib1 10.149.0.67      ib-1
myhost        172.24.0.55      head-bmc
myhost2       172.24.0.56      head-bmc
myhost3       172.24.0.57      head-bmc
```

Delete a site-added IP address (you cannot delete SMC for SGI ICE X needed IP addresses), as follows:

```
admin:~ # cadmin --del-ip --node service0 --net ib-0 my-new-ib0-2-ip=10.148.0.201
```

Change the hostname associated with `service0` to be `myservice`, as follows:

```
admin:~ # cadmin --set-hostname --node service0 myservice
```

Change the hostname associated with `admin` to be `newname`, as follows:

```
admin:~ # cadmin --set-hostname --node admin newname
```

Set and show the cluster subdomain, as follows:

```
admin:~ # cadmin --set-subdomain mysubdomain.domain.mycompany.com
admin:~ # cadmin --show-subdomain
The cluster subdomain is: mysubdomain
```

Show the system admin controller (SAC) house network domain, as follows:

```
admin:~ # cadmin --show-admin-domain
The admin node house network domain is: domain.mycompany.com
```

Show the SMC for SGI ICE X systems DHCP option identifier, as follows:

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

Show the current switch management value for a specified node, as follows:

```
admin:~ # cadmin --show-switch-mgmt-network --node admin
no
```

Enable the switch management network for a specificed node that is connected to managed top level switches, as follows:

```
admin:~ # cadmin --enable-switch-mgmt-network --node admin
```

Note: This option cannot be used for SGI ICE 8200 or SGI ICE 8400 system nodes.

Disable the switch management network for a specificed node that is connected to managed top level switches, as follows:

```
admin:~ # cadmin --disable-switch-mgmt-network --node admin
```

Note: This option cannot be used for SGI ICE 8200 or SGI ICE 8400 system nodes.

Show MySQL replication status for a specified system admin controller (SAC), rack leader controller (RLC), or service node, as follows:

```
admin:~ # cadmin --show-replication --node r2lead
yes
```

Note: MySQL replication is disabled by default.

Enable MySQL replication on a specified SAC, RLC, or service node, as follows:

```
admin:~ # cadmin --enable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...
mysql                     0:off  1:off  2:on   3:on   4:off  5:on   6:off
Restarting service MySQL
Shutting down service MySQL ..done
Starting service MySQL ..done
```

Disable MySQL replication on a specified SAC, RLC, or service node, as follows:

```
admin:~ # cadmin --disable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...
Shutting down service MySQL ..done
mysql                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Console Management

SMC for SGI ICE X 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 SGI ICE X system
- A `conserver` daemon runs on the system admin controller (SAC) and the rack leader controllers (RLCs). The SAC manages RLC and service node consoles. The RLCs 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 SAC.
- All consoles are logged. These logs can be found at `/var/log/consoles` on the SAC and RLCs. An `autofs` configuration file is created to allow you to access RLC-managed console logs from the SAC, as follows:

  ```
  admin #  cd /net/r1lead/var/log/consoles/
  ```

The `/etc/conserver.cf` file is the configuration file for the conserver daemon. This file is generated for both the SAC and the RLCs from the `/opt/sgi/sbin/generate-conserver-files` script on the SAC. 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 SAC.

Note: The conserver package replaces cconsole for access to all consoles (blades, RLCs, 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 4-9 Using conserver Console Manager

To use the conserver console manager, perform the following steps:

1. To see the list of available consoles, perform the following:

```
admin:~ #console -x
 service0                 on /dev/pts/2                    at  Local
 r2lead                   on /dev/pts/1                    at  Local
 r1lead                   on /dev/pts/0                    at  Local
 r1i0n8                   on /dev/pts/0                    at  Local
 r1i0n0                   on /dev/pts/1                    at  Local
```

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

```
admin:~ # console service0
[Enter '^Ec?' for help]


Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64) - Kernel 2.6.16.60-0.12-smp (ttyS1).


service0 login:
```

3. To connect to the RLC console, perform the following:

```
admin:~ # console r1lead
[Enter '^Ec?' for help]


Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64)
- Kernel 2.6.16.60-0.12-smp (ttyS1).


r1lead login:
```

4. To trigger system request commands `sysrq` (once connected to a console), perform the following:

```
Ctrl-e c l 1 8               # set log level to 8
Ctrl-e c l 1 <sysrq cmd>             # send sysrq command
```

5. To see the list of `conserver` escape keys, perform the following:

```
Ctrl-e c ?
```

Keeping System Time Synchronized

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

- "System Admin Controller (SAC) NTP" on page 204
- "Rack Leader Controller (RLC) NTP" on page 204
- "Managed Service, Compute, and Rack Leader Controller (RLC) BMC Setup with NTP" on page 204
- "Service Node NTP" on page 204
- "Compute Node NTP" on page 204
- "NTP Work Arounds" on page 205

System Admin Controller (SAC) NTP

When you used the `configure-cluster` command, it guided you through setting up NTP on the SAC. The NTP client on the SAC 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 SAC sends NTP broadcasts to some networks to keep the nodes in sync after they have booted.

Rack Leader Controller (RLC) NTP

NTP client on the RLC gets time from the system admin controller (SAC) when it is booted and then stays in sync by connecting to the SAC for time. The NTP server on the leader node provides NTP service to SGI ICE components so that compute nodes can sync their time when they are booted. The RLC sends NTP broadcasts to some networks to keep the compute nodes in sync after they have booted.

Managed Service, Compute, and Rack Leader Controller (RLC) BMC Setup with NTP

The BMC controllers on managed service nodes, compute nodes, and RLCs are also kept in sync with NTP. Note that you may need the latest BMC firmware for the BMCs to sync with NTP properly. The NTP server information for BMCs is provided by special options stored in the DHCP server configuration file.

Service Node NTP

The NTP client on *managed* service nodes (for a definition of managed, see "`discover` Command" on page 84) sets its time at initial booting from the system admin controller (SAC). It listens to NTP broadcasts from the SAC 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 (RLC). It listens to NTP broadcasts from the RLC to stay in sync.

NTP Work Arounds

Sometime, especially during initial deployment of an SGI ICE X 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 rack leader controllers (RLCs) and service nodes may fail to get time from the system admin controller (SAC) as they come on-line. Compute nodes may also fail to get time from the RLC 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 SAC and RLCs have the `time` service enabled (`xinetd`).
- All system node types have the `netdate` command.
- A special startup script is on RLC, service, and compute nodes that runs before the NTP startup script.

  This script attempts to get the time using the `ntpdate` command. If the `ntpdate` command fails because the NTP server it is using is not ready yet to offer time service, it uses the `netdate` command to get the clock "close".

  The `ntp` startup script starts the NTP service as normal. Since the clock is known to be "close", NTP will fix the time when the NTP servers start offering time service.

Changing the Size of `/tmp` on Compute Nodes

This section describes how to change the size of `/tmp` on SGI ICE X compute nodes.

Procedure 4-10 Increasing the `/tmp` Size

To change the size of `/tmp` on your system compute nodes, perform the following steps:

1. From the system admin controller (SAC), use the `cd`(1) command to change to the following directory:

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

2. Open the `sgi-fstab` file and change the `size=` parameter for the `/tmp` mount in both locations that it appears.

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation; either version 2 of the License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307 USA
#
# Set up the compute node’s /etc/fstab file.
#
# Modify per your sites requirements.
#
# This script is excecuted once per-host as part of the install-image operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

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

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n ’/^compat /s/^compat.*: //p’) )

compatdistro=${os[0]}${os[1]}
```

```
if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then

        #
        # SLES 10 compatible
        #
        cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
EOF

elif [ ${compatdistro} = "rhel5"  ]; then

        #
        # RHEL 5 compatible
        #

        #
        # RHEL expects several subsys directories to be present under /var/run
        # and /var/lock, hence no tmpfs mounts for them
        #
        cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
devpts          /dev/pts        devpts  gid=5,mode=620  0       0
EOF

else

        echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi
```

3. Push the image out to the racks to pick up the change, as follows:

   # **cimage --push-rack mynewimage r***

   For more information on using the `cimage` command, see "`cimage` Command" on page 169.

Enabling or Disabling the Compute Node iSCSI Swap Device

This section describes how to enable or disable the Internet small computer system interface (iSCSI) compute node swap device. The iSCSI compute node swap device is turned off by default for new installations. It can cause problems during rack-wide out of memory (OOM) conditions, with both compute nodes and the rack leader controller (RLC) becoming unresponsive during the heavy write-out to the per-node iSCSI swap devices.

Procedure 4-11 Enabling the iSCSI Swap Device

If you wish to enable the iSCSI swap device in a given compute node image, perform the following steps:

1. Change root (`chroot`) into the compute node image on the system admin controller (SAC) and enable the `iscsiswap` service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap on
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Procedure 4-12 Disabling the iSCSI Swap Device

To disable the iSCSI swap device in a compute node image where it is currently enabled, perform the following steps:

1. Disable the service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap off
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Changing the Size of Per-node Swap Space

This section describes how to change per-node swap space on your SGI ICE system.

Procedure 4-13 Increasing Per-node Swap Space

To increase the default size of the per-blade swap space on your system, perform the following:

1. Shutdown all blades in the affected rack (see "Shutting Down and Booting" on page 187).

2. Log into the rack leader controller (RLC) for the rack in question. (Note that you need to do this on each RLC).

3. Change directory (`cd`) to the `/var/lib/sgi/swapfiles` directory.

4. To adjust the swap space size appropriate for your site, run a script similar to the following:

```
#!/bin/bash

size=262144     # size in KB

for i in $(seq 0 3); do
        for n in $(seq 0 15); do
                dd if=/dev/zero of=i${i}n${n} bs=1k count=${size}
                mkswap i${i}n${n}
        done
done
```

5. Reboot the all blades in the affected rack (see "Shutting Down and Booting" on page 187).

6. From the RLC, use the `cexec --all` command to run the `free`(1) command on the compute blades to view the new swap sizes, as follows:

```
r1lead:~ # cexec --all free
************************* rack_1 *************************
--------- r1i0n0---------
             total       used       free     shared    buffers     cached
Mem:       2060140     206768    1853372          0          4      46256
-/+ buffers/cache:     160508    1899632
Swap:        49144          0      49144
--------- r1i0n1---------
             total       used       free     shared    buffers     cached
Mem:       2060140     137848    1922292          0          4      44200
-/+ buffers/cache:      93644    1966496
Swap:        49144          0      49144
--------- r1i0n8---------
             total       used       free     shared    buffers     cached
Mem:       2060140     138076    1922064          0          4      43172
-/+ buffers/cache:      94900    1965240
Swap:        49144          0      49144
```

If you want change per-node swap space across your entire system, all (new) RLCs as part of discovery, you can edit the `/etc/opt/sgi/conf.d/35-compute-swapfiles` “inside” the `lead-sles11` image on the system admin controller (SAC). The images are in the `/var/lib/systemimager/images` directory. For more information on customizing these images, see "Customizing Software Images" on page 166.

Switching Compute Nodes to a `tmpfs` Root

This section describes how to switch your system compute nodes to a `tmpfs` root.

Procedure 4-14 Switching Compute Nodes to a `tmpfs` Root

To switch your compute nodes to a `tmpfs` root, from the system admin controller (SAC) perform the following steps:

1. To switch compute nodes to a `tmpfs` root, use the optional `--tmpfs` flag to the `cimage --set` command, for example:

```
adminadmin:~ # cimage --set --tmpfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

> **Note:** To use a `/tmpfs` root with the standard compute node image, the compute node needs to have 4GB of memory or above. A standard `/tmpfs` mount has access to half the system memory, and the standard compute node image is just over 1 GB in size.

2. You can view the current setting of a compute node, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp tmpfs
```

3. To set it back to an NFS root, use the `--nfs` flag to the `cimage --set` command, as follows:

```
admin:~ # cimage --set --nfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

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

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp nfs
```

For help information, use the `cimage --h` option.

Setting up Local Storage Space for Swap and Scratch Disk Space

The SGI ICE X system has the option to support local storage space on compute nodes (also known as blades). Solid-state drive (SSD) devices and 2.5" disks are available for this purpose. You can define the size and status for both swap and scratch partitions. The values can be set globally or per node or group of nodes. By default, the disks are partitioned only if blank, the swap is off, and the scratch is set to occupy the whole disk space and be mounted in `/tmp/scratch`.

The `/etc/init.d/set-swap-scratch` script is responsible for auto-configuring the swap and scratch space based on the settings retrieved via the `cattr` command. You can use the `cadmin` to configure settings globally or you can use the `cattr` command to set custom values for specific nodes.

The `/etc/opt/sgi/conf.d/30-set-swap-scratch` script makes sure `/etc/init.d/swapscratch` service is on so that swap/scratch partitions are configured directly after booting. The `swapscratch` service calls the `/opt/sgi/lib/set-swap-scratch` script when the service is started and then it exits.

You can customize the following settings:

- `blade_disk_allow_partitioning`

  The default value is "`on`" which means that the `set-swap-scratch` script will repartition and format the local storage disk if needed.

  Note: To protect user data, the script will not re-partition the disk if it is already partitioned. In this case, you need a blank disk before it can be used for swap/scratch.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_allow_partitioning` value for the node on which it is running:

  ```
  # cattr get blade_disk_allow_partitioning -N $compute_node_name --default on
  ```

  You can globally set the value on, as follows:

  ```
  # cadmin --add-attribute --string-data on blade_disk_allow_partitioning
  ```

  You can globally turn if off, as follows:

  ```
  # cadmin --add-attribute --string-data off blade_disk_allow_partitioning
  ```

- `blade_disk_swap_status`

  The default value is "`off`" which means that the `set-swap-scratch` script will not enable a swap partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_swap_status` value for the node on which it is running:

  ```
  # cattr get blade_disk_swap_status -N $compute_node_name --default off
  ```

  You can globally set the value on, as follows:

  ```
  # cadmin --add-attribute --string-data on blade_disk_swap_status
  ```

  You can globally turn if off, as follows:

  ```
  # cadmin --add-attribute --string-data off blade_disk_swap_status
  ```

  The `set-swap-scratch` script uses `SGI_SWAP` label when partitioning the disk. It enables the swap only if it finds a partition labeled `SGI_SWAP`.

- `blade_disk_swap_size`

  The default value is 0 which means that the `set-swap-scratch` script will not create a swap partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_swap_size` value for the node on which it is running:

  ```
  attr get blade_disk_swap_size -N $compute_node_name --default 0
  ```

  You can globally set the value, as follows:

  ```
  # cadmin --add-attribute --string-data 1024 blade_disk_swap_size
  ```

  The size is specified in megabytes. Allowed values are, as follows: 0, -0 (use all free space when partitioning), 1, 2, ...

- `blade_disk_scratch_status`

  The default value is "`off`" which means that the `set-swap-scratch` script will not enable the scratch partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_status` value for the node on which it is running:

  ```
  cattr get blade_disk_scratch_status -N $compute_node_name --default off
  ```

You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_scratch_status
```

You can globally turn if off, as follows:

```
cadmin --add-attribute --string-data off blade_disk_scratch_status
```

Note: The `set-swap-scratch` script uses the `SGI_SCRATCH` label when partitioning the disk. It mounts the scratch only on the partition labeled as `SGI_SCRATCH`.

- `blade_disk_scratch_size`

  The default value is `-0` which means that the `set-swap-scratch` script will use all remaining free space when creating the scratch partition.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

  ```
  cattr get blade_disk_scratch_size -N $compute_node_name --default -0
  ```

  You can globally set the value, as follows:

  ```
  cadmin --add-attribute --string-data 10240 blade_disk_scratch_size
  ```

  The size is specified in megabytes. Allowed values are, as follows: 0, -0 (use all free space when partitioning), 1, 2, ...

- `blade_disk_scratch_mount_point`

  The default value is `/tmp/scratch` which means that the `set-swap-scratch` script will mount the scratch partition in `/tmp/scratch`.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

```
# cattr get blade_disk_scratch_mount_point -N $compute_node_name --default /tmp/scratch
```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data /tmp/scratch blade_disk_scratch_mount_point
```

You can mount the disk to any mount point you desire. The `set-swap-scratch` script will create that folder if it does not exists (as long as the script has the permission to create it at that path). The root mount point (/) is not writable on the compute nodes. You need to create that folder as part of the compute node image if you want to mount something like `/scratch`.

- `blade_disk_raid_level`

  The default value is `off`. When set to `0`, it allows you to setup RAID0, if you have two disk for swap/scratch. Values of `1, ...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_raid_level` value for the node on which it is running:

  ```
  attr get blade_disk_raid_level -N $compute_node_name --default 0
  ```

  You can globally set the value, as follows:

  ```
  # cadmin --add-attribute --string-data blade_disk_raid_level
  ```

- `blade_disk_reformat_swap_at_boot`

  The default value is `off`. When set to `0`, it allows you to format the swap every time the node reboots. Values of `1, ...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_swap_at_boot` value for the node on which it is running:

  ```
  attr get blade_disk_reformat_swap_at_boot -N $compute_node_name --default 0
  ```

  You can globally set the value, as follows:

  ```
  # cadmin --add-attribute --string-data blade_disk_reformat_swap_at_boot
  ```

- `blade_disk_reformat_scratch_at_boot`

  The default value is `off`. When set to `0`, it allows you to format the swap every time the node reboots. Values of `1, ...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_scratch_at_boot` value for the node on which it is running:

  ```
  attr get blade_disk_reformat_scratch_at_boot -N $compute_node_name --default 0
  ```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data blade_disk_reformat_scratch_at_boot
```

For a `cattr` command help statement, perform the following command:

```
# cattr -h
Usage:
  cattr [--help] COMMAND [ARG]...

Commands:
  exists   check for the existence of an attribute
  get      print the value of an attribute
  list     print a list of attribute values
  set      set the value of an attribute
  unset    delete the value of an attribute

For more detailed help, use 'cattr COMMAND --help'.
```

Viewing the Compute Node Read-Write Quotas

This section describes how to view the per compute node read and write quota.

Procedure 4-15 Viewing the Compute Node Read-Write Quotas

To view the per compute node read and write quota, log onto the rack leader controller (RLC) and perform the following:

```
r1lead:~ # xfs_quota -x -c 'quota -ph 1'
Disk quotas for Project #1 (1)
Filesystem   Blocks  Quota  Limit Warn/Time    Mounted on
/dev/disk/by-label/sgiroot
              64.6M      0     1G  00 [------] /
```

Map the XFS project ID to the quota you are interested in by looking it up in `/etc/projects` file.

If you decided to change the `xfs_quota` values, log back onto the system admin controller (SAC) and edit the `/etc/opt/sgi/cminfo` file **inside** the compute image where you want to change the value, for example,

/var/lib/systemimager/images/*image_name*. Change the value of the PER_BLADE_QUOTA variable and then repush the image with the following command:

cimage --push-rack *image_name racks*

For help information, perform the following:

```
xfs_quota> help
df [-bir] [-hn] [-f file] -- show free and used counts for blocks and inodes
help [command] -- help for one or all commands
print -- list known mount points and projects
quit -- exit the program
quota [-bir] [-gpu] [-hnv] [-f file] [id|name]... -- show usage and limits

Use 'help commandname' for extended help
```

Use help *commandname* for extended help, such as the following:

```
xfs_quota> help quota

quota [-bir] [-gpu] [-hnv] [-f file] [id|name]... -- show usage and limits

 display usage and quota information

 -g -- display group quota information
 -p -- display project quota information
 -u -- display user quota information
 -b -- display number of blocks used
 -i -- display number of inodes used
 -r -- display number of realtime blocks used
 -h -- report in a human-readable format
 -n -- skip identifier-to-name translations, just report IDs
 -N -- suppress the initial header
 -v -- increase verbosity in reporting (also dumps zero values)
 -f -- send output to a file
 The (optional) user/group/project can be specified either by name or by
 number (i.e. uid/gid/projid).

xfs_quota>
```

RAID Utility

The infrastructure nodes on your SGI ICE system have LSI Logic RAID enabled by default from the factory. Prior SGI ICE systems shipped with the `lsiutil` command-line utility. SGI ICE X ships with the LSI Logic MegaRAID command line tool (see "LSI Logic MegaRAID Command-line Utility" on page 220).

LSI Logic `lsiutil` Command-line Utility

The `lsiutil` command-line utility is included with the installation for the system admin controller (SAC), the rack leader controller (RLC), and the service node (when installed from the SGI service node image). This tool allows you to look at the devices connected to the RAID controller and manage them. Some functions, such as, setting up mirrored or striped volumes, can be handled either by the LSI BIOS configuration tool or the `lsiutil` utility.

Note: These instructions only apply to Altix XE250 or Altix XE270 systems with the 1068-based controller. They do not apply to Altix XE250 or Altix XE270 systems that have the LSI Megaraid controller.

Example 4-7 Using the `lsiutil` Utility

The following `lsiutil` command-line utility example shows a sample session, as follows:

Start the `lsiutil` tool, as follows:

```
admin:~ # lsiutil

LSI Logic MPT Configuration Utility, Version 1.54, January 22, 2008

1 MPT Port found

     Port Name         Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
 1.  /proc/mpt/ioc0    LSI Logic SAS1068E B2     105      01140100     0

Select a device:  [1-1 or 0 to quit]
```

Select **1** to show the **MPT Port**, as follows:

```
1 MPT Port found
```

```
    Port Name         Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
 1.  /proc/mpt/ioc0    LSI Logic SAS1068E B2     105      01140100     0

Select a device:  [1-1 or 0 to quit] 1

 1.  Identify firmware, BIOS, and/or FCode
 2.  Download firmware (update the FLASH)
 4.  Download/erase BIOS and/or FCode (update the FLASH)
 8.  Scan for devices
10.  Change IOC settings (interrupt coalescing)
13.  Change SAS IO Unit settings
16.  Display attached devices
20.  Diagnostics
21.  RAID actions
22.  Reset bus
23.  Reset target
42.  Display operating system names for devices
45.  Concatenate SAS firmware and NVDATA files
60.  Show non-default settings
61.  Restore default settings
69.  Show board manufacturing information
97.  Reset SAS link, HARD RESET
98.  Reset SAS link
99.  Reset port
 e   Enable expert mode in menus
 p   Enable paged mode in menus
 w   Enable logging

Main menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **21. RAID** actions, as follows:

```
Main menu, select an option:  [1-99 or e/p/w or 0 to quit] 21

 1.  Show volumes
 2.  Show physical disks
 3.  Get volume state
 4.  Wait for volume resync to complete
23.  Replace physical disk
26.  Disable drive firmware update mode
27.  Enable drive firmware update mode
30.  Create volume
```

```
31.  Delete volume
32.  Change volume settings
50.  Create hot spare
99.  Reset port
 e   Enable expert mode in menus
 p   Enable paged mode in menus
 w   Enable logging

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **2. Show physical disks**, to show the status of the disks making up the volume, as follows:

```
RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit] 2

1 volume is active, 2 physical disks are active

PhysDisk 0 is Bus 0 Target 1
  PhysDisk State:  online
  PhysDisk Size 238475 MB, Inquiry Data:  ATA      Hitachi HDT72502 A73A

PhysDisk 1 is Bus 0 Target 2
  PhysDisk State:  online
  PhysDisk Size 238475 MB, Inquiry Data:  ATA      Hitachi HDT72502 A73A

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **1. Show volumes**, to show information about the volume including its health, as follows:

```
RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit] 1

1 volume is active, 2 physical disks are active

Volume 0 is Bus 0 Target 0, Type IM (Integrated Mirroring)
  Volume Name:
  Volume WWID:  09195c6d31688623
  Volume State:  optimal, enabled
  Volume Settings:  write caching disabled, auto configure
  Volume draws from Hot Spare Pools:  0
  Volume Size 237464 MB, 2 Members
  Primary is PhysDisk 1 (Bus 0 Target 2)
```

```
  Secondary is PhysDisk 0 (Bus 0 Target 1)

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

LSI Logic MegaRAID Command-line Utility

This section provides a brief description of the LSI Logic MegaRAID command-line utility. There is also a graphical version available that you can download and install should you choose to.

For an MegaRAID help statement, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -h
```

To show physical disks, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -pdInfo -PhysDrv[252:0] -a0
```

To show logical disk information, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -LdPdInfo -a0
```

To show a MegaRAID summary, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -ShowSummary -a0
```

Restoring the grub Boot-loader on a Node

When grub(8) boot-loader is not written to the rack leader controllers (RLCs) or any of the system service nodes or is not functioning correctly, the grub boot-loader will have to be re-installed on the master boot record (MBR) of the root drive for the node.

To rewrite grub to the MBR of the root drive on a system that is booted, issue the following grub commands:

```
# grub
grub>    root (hd0,0)
grub>    setup (hd0)
grub>    quit
```

If you cannot boot your system (and it is hanging on `grub`), you need to boot the node in rescue mode and then issue the following commands:

```
# mount /dev/ /system
# mount -o bind /dev /system/dev
# mount -t proc proc /system/proc # optional
# mount -t sysfs sysfs /system/sys # optional
# chroot /system
# grub
grub>    root (hd0,0)
grub>    setup (hd0)
grub>    quit
# reboot
```

Backing up and Restoring the System Database

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

Managed objects on an SGI ICE X include the following

- SGI ICE X system

  One SGI ICE X system is modeled as a meta-cluster. This meta-cluster contains the racks each modeled as a sub-cluster.

- Nodes

  System admin controller (SAC), rack leader controllers (RLCs), 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 4-16 Backing up and Restoring the System Database

To back up and restore the system database, perform the following steps:

1.

   Note: A password is required to use the `mysqldump` command. The password file is located in the `/etc/odapw` file.

   From the SAC, 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 SAC, perform a command similar to the following:

   ```
   # mysql oscar < backup-file.sql
   ```

For more information, see the `mysqldump`(1) man page.

Enabling EDNS

Extension mechanisms for DNS (EDNS) can cause excessive logging activity when not working properly. SMC on SGI ICE X contains code to limit EDNS logging. This section describes how to delete this code and allow EDNS to work unrestricted and log messages.

Procedure 4-17 Enabling EDNS

To enable EDNS on your SGI ICE X system, perform the following steps:

1. Open the `/opt/sgi/lib/Tempo/Named.pm` file with your favorite editing tool.

2. To remove the limit on the `edns_udp_size` parameter, comment out or remove the following line:

   ```
   $limit_edns_udp_size = "edns-udp-size 512;";"
   ```

3. Remove the following lines so that EDNS logging is no longer disabled:

   ```
   logging {
   category lame-servers {null; };
   category edns-disabled { null; };  };
   ```

Firmware Management

The `fwmgr` tool and its associated libraries form a firmware update framework. This framework makes managing the various firmware types in a cluster easier.

A given cluster may have several types of firmware including mainboard BIOS, BMC, disk controllers, InfiniBand (`ib`) interfaces, /Ethernet NICs, network switches, and many other types.

The firmware management tools allow the firmware to be stored in a central location (firmware bundle library) to be accessed by command line or graphical tools. The tools allow you to add firmware to the library, remove firmware from the library, install firmware on a given set of nodes, and other related operations.

License Requirement

This framework is licensed. It cannot be used without the appropriate license.

Terminology

This section describes some terminology associated with the firmware management, as follows:

- Raw firmware file

  These are files that you download, likely from SGI, that include the firmware and option tools to flash said firmware. For example, a raw firmware file for an SGI ICE X compute node BIOS update might be downloaded as, `sgi-ice-blade-bios-2009.12.14-1.x86_64.rpm`.

- Firmware bundle

A firmware bundle is a file that contains the firmware to be flashed in a way that the integrated tools understand. Normally, firmware bundles are stored in the firmware bundle library (see below). However, these bundles can also be checked out of the library and accessed directly in some cases. In most situations, a firmware bundle is a sort of wrapper around the raw firmware file(s) and various attributes and tools. A firmware bundle can contain more than one type of firmware. This is the case when the underlying flash tool supports more than one firmware type. An example of this is the SGI ICE X compute node firmware, that contains several different BIOS files for different mainboards and multiple BMC firmware revisions. Another example might be a raw file that includes both the BIOS and BMC firmware for a given mainboard/server.

- Firmware bundle library

  This is a storage repository for firmware bundles. The management tools allow you to query the library for available bundles and associated attributes.

- Update environment

  Some raw firmware types, like the various SGI ICE X firmware released as RPMs, run "live" on the system admin controller (SAC) to facilitate flashing. The underlying tool may indeed set nodes up to network boot a low level flash tool, but there are many other methods used by the underlying tools. Some firmware types, like BIOS ROMs with associated flash executables, require an update environment to be constructed. One type of update environment is a DOS Update Environment. This update environment may be used, for example, to construct a DOS boot image for the BIOS ROM and associated flash tool. A firmware bundle calls for a specific update environment. In this way, a firmware bundle with an associated update environment form the necessary pieces to facilitate booting of a DOS update environment over the network that flashes the target nodes with the specified BIOS ROM (as an example).

Firmware Update High Level Example

This section describes the steps you need to take to update a set of nodes in your cluster with a new BIOS level, as follows:

- Download the raw firmware file for this system type. You might do this, for example, from SGI Supportfolio web site located at https://support.sgi.com/login.
- Add the raw firmware file to the firmware bundle library using a graphical or command line tool.

- The tool will convert the raw firmware file into a firmware bundle and store it in the firmware bundle library. In some cases, you will be required to provide additional information in order to convert the raw firmware file into a firmware bundle. This could be information necessary to facilitate flashing that the framework can not derive from the file on its own.
- Once the firmware bundle is available in the firmware library, you can use the graphical or command line tool to select a firmware bundle and a list of target nodes to which to push the firmware update.
- The underlying tool then creates the appropriate update environment (if required) and facilitates flashing of the nodes.

Firmwware Manager Command Line Interface (`fwmgr`)

The `fwmgr` command is the command line interface (CLI) to the firmware update infrastructure.

For a usage statement, enter `fwmgr --help`. The `fwmgr` command has several sub-commands, each of which can be called with the `--help` option for usage information.

You can use the `fwmgr` command to perform the following:

- List the available firmware bundles
- Add raw firmware files or firmware bundle files to the firmware bundle library. If it is a raw firmware type, it will be converted to a firmware bundle and placed in the library.
- Remove firmware bundles from the firmware bundle library
- Rename an existing firmware bundle in the firmware bundle library
- Install a given firmware bundle on to a list of nodes
- Checkout a firmware bundle which allows you to store the firmware bundle itself

---

Note: It is currently not necessary to run the `fwmgrd` command (firmware manager daemon) to use the CLI.

Firmwware Manager Daemon (`fwmgrd`)

This `fwmgrd` daemon is installed and enabled by default in SGIMC 1.3 on SGI ICE X systems, only. This daemon provides the services needed for the SGI Management Center graphical user interface to communicate with the firmware management infrastructure. This daemon needs to be running in order to access firmware management from the graphical user interface.

Even if you intend to only use the CLI, it is recommended that the `fwmgrd` daemon be left running and available.

By default, the `fwmgrd` log file is located at:

`/var/log/fwmgrd.log`

View this log for important messages during flashing operations from the SGI Management Center graphical interface.

Notes specific to Management Center 1.5

The first release of the Firmware Management framework only supports SGI ICE X firmware, released as RPMs. This includes: `sgi-ice-blade-bios`, `sgi-ice-blade-ib`, `sgi-ice-blade-zoar`, `sgi-ice-cmc`, and `sgi-ice-ib-switch`. This includes the SGI ICE X compute nodes but does not yet include other managed node types.

SGI intends to expand this firmware management framework to support additional node types in SGI ICE X and SGI Rackable cluster hardware in later releases.

Note: SGI ICE X integrated InfiniBand switches are supported but only on SGI ICE 8400 series systems or later. Some integrated InfiniBand switch parts in the SGI ICE 8200 series systems will **not** flash properly with this framework.

System Fabric Management

The InfiniBand network on SGI ICE X systems uses Open Fabrics Enterprise Distribution (OFED) software. This section describes the InfiniBand fabric and how to manage it. For background information on OFED, see http://www.openfabrics.org.

InfiniBand Fabric Management

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

- "InfiniBand Fabric Overview" on page 227
- "InfiniBand Management Tool Graphical User Interface" on page 228
- "Fabric Component `sgifmcli` Command" on page 232
- "InfiniBand Fabric Management Configuration and Operation Overview" on page 236
- "InfiniBand Fabric Failover Mechanism" on page 240
- "Configuring the InfiniBand Fat-tree Network Topology" on page 242
- "Configuring the Lightweight Fabric" on page 244
- "Useful Utilities and Diagnostics" on page 245

InfiniBand Fabric Overview

InfiniBand fabric management on SGI ICE X systems is done using the OFED OpenSM software package and the `sgifmcli` tool (see "Fabric Component `sgifmcli` Command" on page 232). The InfiniBand fabric connects the service nodes, rack leader controllers (RLCs), and the compute nodes. It does not connect to the system admin controller (SAC) or the chassis management control (CMC) blades. SGI ICE X systems usually have two separate InfiniBand fabrics, which are generally referred to as `ib0` and `ib1` within this manual.

On SGI ICE X systems, each InfiniBand fabric (also sometimes called an InfiniBand subnet) has its own subnet manager (SM), which runs on an RLC. For a system with two or more racks, the SM for each fabric is usually configured to run on different

RLCs. In a single rack system, both SMs will run on the single RLC. Each SM may also be paired with a standby SM which can take over in the event of the failure of the primary SM. For more information, see "InfiniBand Fabric Failover Mechanism" on page 240.

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

RLCs associate a SM instance with a particular port on the RLC. Usually, `ib0` is mapped to port 1 of the InfiniBand host channel adapter (HCA) on the SM node, and `ib1` is mapped to port 2 of the HCA on the SM node. SM for `ib0` and `ib1` is configured using the corresponding `/etc/ofa/opensm-ib[01].conf` file.

Note: After a system reboot, the `opensm` daemons start running automatically.

SGI supports the following topologies: hypercube, enhanced hypercube, and fat tree.

InfiniBand Management Tool Graphical User Interface

You can use the InfiniBand management tool graphical user interface (GUI) to configure, administer, or verify the InfiniBand fabric on your SGI ICE X system. You can use it to configure, start, stop, restart, cleanup, or get status for the InfiniBand fabric.

From the system admin controller (SAC), enter the following command:

```
admin:~ # smc-configure-fabric
```

The **InfiniBand Management Tool** GUI appears, as shown in Figure 5-1 on page 229.

You can also access this command from the `configure-cluster` GUI main menu **Configure Infiniband Fabric** option (see "Running the Cluster Configuration Tool" on page 61). For more information, see Figure 5-1.

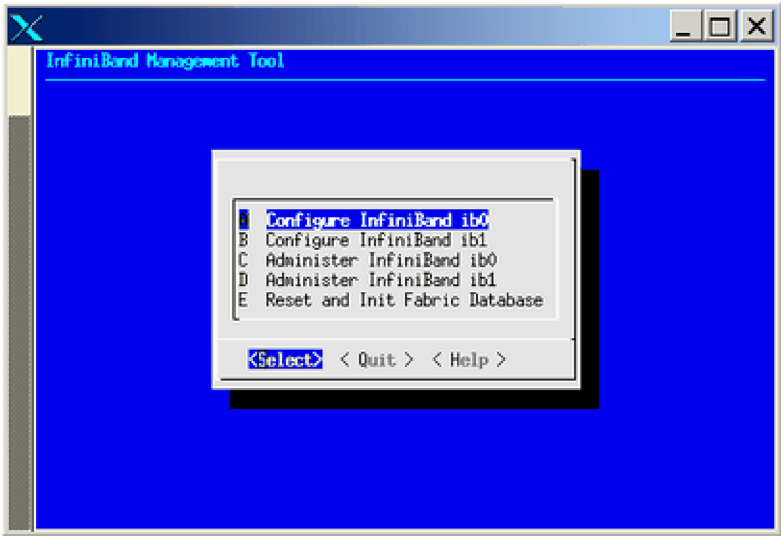

Figure 5-1 InfiniBand Management Tool Screen

Use the **Select** button to select the action you want to perform. A submenu will appear. Use the **Quit** button to return to the previous screen. Use the InfiniBand Management GUI to manage your InfiniBand fabric. You can use the **Help** button to get online help for each of the GUI actions.

If the `smc-configure-fabric` command fails in a configuration or administrative operation, it suggests that you use the `sgifmcli`(8) command (described in "Fabric Component `sgifmcli` Command" on page 232) to debug the problem. Alternatively, you can use the **Reset and Init Fabric Database** option from the **InfiniBand Management Tool** main menu (see Figure 5-1 on page 229) to start over and completely reconfigure the InfiniBand fabrics.

From the **Configure InfiniBand** screen, make sure you select the **Configure Topolgy** option to set the topology as shown in Figure 5-2 on page 230. For more information, see "Network Topology" on page 237.

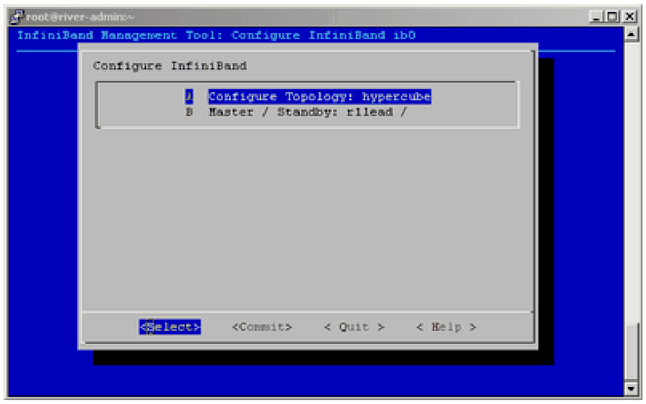

Figure 5-2 Configure Topology Screen

Use the the online help available with this tool to guide you through the InfiniBand configuration. After configuring and bringing up the InfiniBand network, select the **Administer InfiniBand ib0** option or the **Administer InfiniBand ib1** option, the **Administer InfiniBand** screen appears as shown in Figure 5-3. You can use this screen to start, stop, restart, or refresh a fabric.

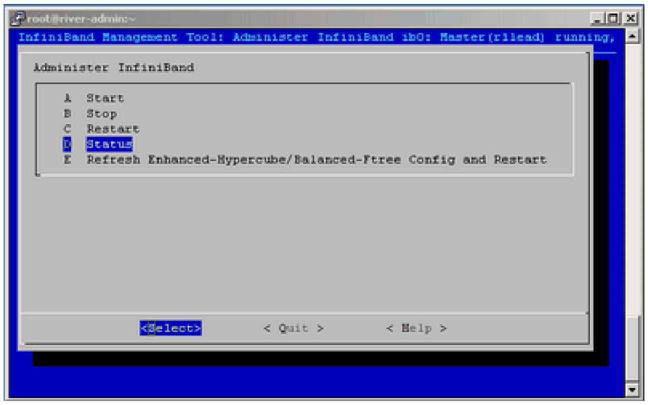

Figure 5-3 Administer InfiniBand Tool Screen

You can verify the status via the **Status** option, as shown in Figure 5-4 on page 231.

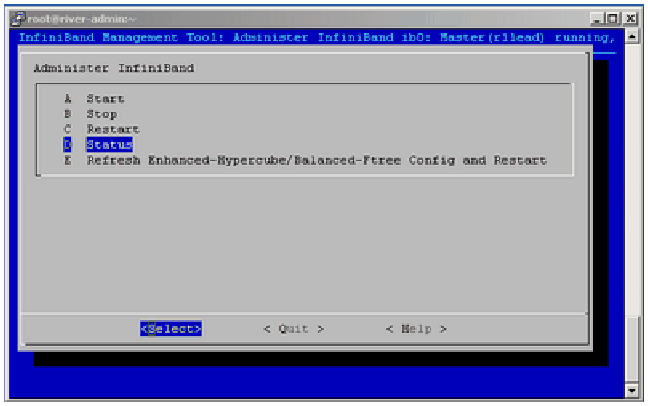

Figure 5-4 **Administer InfiniBand Status** Option

The **Status** option returns information similar to the following:

```
Master SM
Host = r1lead
Guid = 0x0002c9030006938b
Fabric = ib0
Topology = hypercube
Routing Engine = dor
OpenSM = running
```

Press the `Enter` key to return to the `configure-cluster` GUI.

The **Refresh Enhanced Hypercube Config and Restart** option applies only to the Enhanced Hypercube topology. You are required to refresh the fabric configuration when you either add, remove, or move one or more compute blades or service nodes. The refresh action updates the `guid` routing order file which is used to balance InfiniBand traffic for the Enhanced Hypercube topology. In addition, this action also automatically restarts the master subnet manaager (SM) and the optional standby SM for the specified fabric (see "InfiniBand Fabric Failover Mechanism" on page 240).

Ideally, the refresh action for a fabric should be taken when there are no jobs running in the system. Restarting the subnet manager can have an adverse impact on the running jobs in the system.

Fabric Component `sgifmcli` Command

For the most common fabric management operations, the `smc-configure-fabric` command (described in "InfiniBand Management Tool Graphical User Interface" on page 228) is entirely sufficient, and recommended. The `sgifmcli`(8) command can be used for more advanced fabric management tasks.

The most common operations that `sgifmcli` would be used for are, as follows:

- Initializing and configuring external InfiniBand switches
- Verifying the integrity of the InfiniBand fabric(s)

For more information, see the `sgifmcli`(8) man page.

Currently, the following switches are supported:

Switch Type	Description
`voltaire-isr-9024`	Voltaire ISR 9024
`voltaire-isr-2004`	Voltaire ISR 2004
`voltaire-isr-2012`	Voltaire ISR 2012
`voltaire-isr-9096`	Voltaire ISR 9096
`voltaire-isr-9288`	Voltaire ISR 9288
`voltaire4036`	Voltaire Grid Director 4036
`mellanox5030`	Mellanox IS5030

To configure an external InfiniBand switch, cluster-wide InfiniBand connectivity is not required. The only necessity is that the supplied switch host name is resolvable and a working networking connection to the external InfiniBand switch exists. See the `sgifmcli`(8) man page for more information about adding external InfiniBand switches to your cluster's fabric.

Verify the integrity of an InfiniBand fabric requires that the InfiniBand network is first configured properly. This is most easily done using `smc-configure-fabric` (see "InfiniBand Management Tool Graphical User Interface" on page 228). See the `sgifmcli`(8) man page for details about the fabric verification operation.

`sgifmcli` SGI Fabric Component Command

The `sgifmcli`(8) command is, as follows:

```
sgifmcli [type action [options]] | [options]
```

Note: You can use shortened versions of the following `sgifmcli` options as long as the option is unambiguous. For example, **`sgifmcli --vers`** for **`sgifmcli --version`**.

It accepts the following general options:

General Option	Description
`-h, --help`	Displays a help message and the exits
`-V, --version`	Shows the version number of the program
`-v, --verbose [DEBUG	INFO	ERROR]`	Select verbosity level (default: ERROR). Most the messages from `sgmifmcli` are written to a log file named `/var/log/sgifmcli.log`. The default level reports error messages only. `INFO` provides the user with details about the operation of `sgifmcli` in addition to error messages. The `DEBUG` level produces output that is tailored toward the developer to help with bug fixing. In addition, the `DEBUG` level also produces `INFO` and `ERROR` messages.

It accepts the following detailed options:

Detailed Option	Description
`type`	The `type` option is one of the following: • `--mastersm` – Master subnet manager • `--standby` – Standby subnet manager • `--ibswitch` - InfiniBand switch • `--ibfabric` - InfiniBand fabric
`action`	The `action` option is one of the following: • `--init` - Initializes the switch or fabric

- `--start` - Starts a subnet manager
- `--stop` - Stops a subnet manager
- `--status` - Prints the status of a subnet manager
- `--verify` - Verifies the fabric
- `--refresh` - Update a InfiniBand fabric (for Enhanced Hypercube)
- `--set` - Sets specific SM configuration parameter (see `arglist`)
- `--add` - Adds a subcomponent to its container, for example, add a switch to a fabric
- `--delete` - Deletes a subcomponent from its container, for example, delete a switch from a fabric Removes the switch or fabric
- `--remove` - Removes an entity
- `--showconfig` - Prints fabric configuration
- `--switchlist` - Lists switches in a fabric
- `--create-node-name-map` – Creates a node name map for internal SGI ICE X switches

`options`

The `options` option is one or more of the following with no duplicates, for example, the `--fabric` option must be either `ib0` or `ib1`, not both:

- `--id` - Unique identifier, for example, host name
- `--hostname` - Name of the node on which to run OpenSM
- `--switchtype` - Type of switch (leaf or spine)
- `--model` - Switch model (`voltaire-isr-9024`, `voltaire-isr-2004`, `voltaire-isr-2012`, `voltaire-isr-9096`, or `voltaire-isr-9288`)
- `--fabric` – Fabric, either `ib0` or `ib1`

- `--topology` – InfiniBand topology, either `hypercube`, `enhanced-hypercube`, or `ftree`
- `--arglist` - List of Subnet Manager configuration parameters: `param_1=val_1`, `param_2=val_2`, `...`

EXIT CODES

To facilitate the use of the `sgifmcli`(8) command in shell scripts, an exit code is returned to give an indication of what occurred during a given connection.

The exit codes returned by `sgifmcli` are, as follows:

`0`	Successful termination.
`255`	Abnormal termination.

For a detailed man page, perform the following command from the system admin controller (SAC):

```
admin:~ # man sgifmcli
```

The `sgifmcli`(8) fabric administration utilities man page appears.

sgifmdb Fabric Management Database Command

The fabric component maintains a database (DB) of the objects it manages (managed objects). The database version is automatically set during cluster install. You do not need to set it. Most likely, this database will change over time. To manage multiple database versions and also to aid in field support, SGI has added another command line tool that currently reports the managed objects database version.

The `sgifmdb` command is, as follows:

```
sgifmdb [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]
```

It accepts the following general options:

General Option	Description
`-g, --get`	Reads the database version object from the database
`-d, --dump`	Dumps the database. This option allows the you to see what fabric objects are currently stored in the fabric database.

`-v, --version`	Prints version
`-r, --reset`	Resets the database and starts clean
`-h, --help`	-h, –help

Example 5-1 Getting `sgifmdb`(8) Command Help

For a `sgifmdb` command usage statement, perform the following from the system admin controller (SAC):

```
admin:~ # sgifmdb -h
SGI Fabric Component DB tool
Usage: db_version [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]

        -g, --get       Read DB version object from DB
        -d, --dump      Dump the DB
        -v, --version   Print version
        -r, --reset     Reset the database and start clean
        -h, --help      Show this text
```

InfiniBand Fabric Management Configuration and Operation Overview

Each subnet manager (SM) performs a light sweep of the fabric it is managing, every 10 seconds by default. The time interval is set by setting the `sweep_interval` variable in the `/opt/sgi/var/sgifmcli/opensm-ib0.conf.templ` file and then doing a **Commit** operation in the `smc-configure-fabric` GUI. Alternately, the `sgifmcli` command has a `--arglist` option to set various subnet manager configuration parameters including the sweep interval.

Note: If your cluster is larger than 256 nodes, SGI highly recommends increasing this variable to 90 seconds or even larger value.

If an SM detects a change in the fabric during a light sweep, such as, the addition or deletion of a node, it performs a *heavy* sweep. The heavy sweep actually changes the fabric configuration to reflect the current state of the system. For more information, see the `opensm`(8) man page on the rack leader controller (RLC).

The `opensm-ib`*x*`.conf` configuration files are located in the `/opt/sgi/var/sgifmcli` directory on the system admin controller (SAC).

Each opensm instance (one for each fabric) associates itself with a particular globally unique identifier (GUID) for a port on the node where opensm runs (see). This association is configured with the "guid" entry in the corresponding opensm-ib[01].conf file.

Network Topology

For SGI ICE X systems with a hypercube topology, SGI uses the dimension order routing (DOR) algorithm.

The dimension order routing algorithm is based on the min hop algorithm and so uses shortest paths. Instead of spreading traffic out across different paths with the same shortest distance, it chooses among the available shortest paths based on an ordering of dimensions.

For SGI ICE X systems with a fat-tree topology, SGI uses updn as the default routing algorithm. Unicast routing algorithm (UPDN) is also based on the minimum hops to each node, but it is constrained to ranking rules.

For more information on routing variables, see the opensm(8) man page.

Hypercube network topology is well suited for smaller node count MPI jobs or jobs that have communication patterns that are not sensitive to bisection bandwidth. Fat-tree network topology is well suited for large node count MPI jobs that are sensitive to bi-section bandwidth.

As stated above, there are two opensm daemons, one for each fabric, opensmd-ib0 and opensmd-ib1, respectively. They are controlled by the init.d scripts. Each init.d script has a separate configuration file for each fabric, opensm-ib0 and opensm-ib1, respectively.

You can use the sminfo command to show the GUID of the SM master.

Configuring the InfiniBand Fabric

This section describes how to configure and administer the InfiniBand fabric using the sgifmcli(8) command.

Note: SGI highly recommends that you use the smc-configure-fabric GUI to configure and administer the fabric (see "InfiniBand Management Tool Graphical User Interface" on page 228).

Procedure 5-1 Configure the Master Subnet Manager

When configuring the SM master, the following rules apply:

- Each InfiniBand fabric needs to have a subnet manager (SM) master.
- There can be at most one SM master per InfiniBand fabric.
- Fabric configuration and administration can only be done via the SM master.
- Fabric configuration becomes active after (re)starting the SM master.
- Deleting an SM master automatically deletes its standby, if it exists.

The syntax to configure an SM master is, as follows:

```
sgifmcli --mastersm --init --id identifier --hostname hostname --fabric fabric --topology topology
```

This command creates a master with the name provided by the `--id` option. The identifier can be any arbitrary string. The `hostname` determines the host on which the SM master manager is launched. The `fabric` option associates the SM master manager with either `ib0` or `ib1`. The `topology` option refers to the InfiniBand topology, which can be either hypercube, enhanced hypercube, or fat tree.

To configure a master for the fabric `ib0` on a hypercube cluster, perform the following steps:

1. From the system admin controller (SAC) to configure an SM master, perform the following:

```
# sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube
```

   This creates an SM master for `ib0`. The underlying topology is a hypercube and thus the routing algorithm `dor` will be used. This SM master, named `master_ib0`, is configured to run on the host `r1lead`.

2. The syntax to start an SM master is, as follows:

   ```
   sgifmcli --start --id identifier
   ```

   To start the `master_ib0` SM master, perform the following:

   ```
   # sgifmcli --start --id master_ib0
   ```

At this point a master for the fabric `ib0` is running on the `r1lead` and thus the fabric `ib0` is available for compute jobs. If a standby has been defined, it will be launched automatically, in addition, to the master.

3. The syntax to stop an SM master is, as follows:

   ```
   sgifmcli --stop --id identifier
   ```

   To stop the `master_ib0` SM master, perform the following:

   ```
   # sgifmcli --stop --id master_ib0
   ```

   The SM master `master_ib0` running on host `r1lead` is stopped. If a standby has been defined then it will be stopped automatically, in addition to the master.

4. The syntax to check the status of an SM master is, as follows:

   ```
   sgifmcli --status --id identifier
   ```

   To check the status of the `master_ib0` SM master, perform the following:

   ```
   # sgifmcli --status --id master_ib0
   Master SM
   Host = rlead
   Guid = 0x0002c902002838f5
   Fabric = ib0
   Topology = hypercube
   Routing Engine = dor
   OpenSM = running
   ```

   The status of the master SM master `master_ib0` running on host `r1lead` is reported. If a standby has been defined, its status will be reported in addition to the master.

5. The syntax to remove an SM master is, as follows:

   ```
   sgifmcli --remove --id identifier
   ```

   To remove the `master_ib0` SM master, first stop it and then perform the **`-remove`** option, as follows:

   ```
   # sgifmcli --stop --id master_ib0
   
   # sgifmcli --remove --id master_ib0
   ```

The SM master is removed from the entity list. If a standby has been defined, it is removed, in addition to the master.

6. To find the ID of the master SM in the database, perform the following:

   ```
   # sgifmcli --dump --id ib0 | grep MASTER
   ```

7. To print the fabric configuration, run the following:

   ```
   # sgifmcli --showconfig

   --------------
   NAME = ib1
   TYPE = ibfabric
   MASTER =
   STANDBY =
   SWITCH_LIST =
   --------------
   NAME = ib0
   TYPE = ibfabric
   MASTER =
   STANDBY =
   SWITCH_LIST =
   ```

InfiniBand Fabric Failover Mechanism

Each subnet manager (SM) has a failover mechanism. If the master SM fails, the standby SM takes over operation of the fabric. This failover operation is performed automatically by the `opensm` software.Typically, `rack1` is the `MASTER` for the `ib0` fabric and `rack2` has the `MASTER` for the `ib1` fabric.

The following procedure describes how to setup the failover mechanism.

Procedure 5-2 Enabling the InfiniBand Failover Mechanism

When enabling the InfiniBand failover mechanism, the following rules apply:

- Each InfiniBand fabric can optionally have exactly one standby.
- A standby SM can only be created for a particular fabric when a master already exists.
- When adding a standby after a master has already been defined and started, the master needs to be stopped before the standby is defined via the `--init` option. After defining the standby via `--init`, restart the master.

- A SM master and SM standby for a particular fabric can not coexist on the same node.

SGI highly recommends that you use the `smc-configure-fabric` GUI to configure the failover mechanism. If it is necessary to use `sgifmcli`(8) to enable the InfiniBand failover mechanism, perform the following steps:

1. If an SM master is defined and running, stop it, as follows:

```
# sgifmcli --stop --id master_ib0
```

   If the SM master has not been defined, define it, as follows:

```
# sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube
```

2. Define the SM standby, as follows:

```
# sgifmcli --standbysm --init --id standby_ib0 --hostname r2lead --fabric ib0
```

3. Start the SM master, as follows:

```
# sgifmcli --start --id master_ib0
```

   This automatically starts the SM master and the SM standby for `ib0`.

4. Now check the status for the subnet manager of ib0, as follows:

```
sgifmcli --status --id master_ib0

Master SM
Host = r1lead
Guid = 0x0008f10403987da9
Fabric = ib0
Toplogy = hypercube
Routing Engine = dor
OpenSM = running
Standby SM
Host = r2lead
Guid = 0x0008f10403987d25
Fabric = ib0
OpenSM = running
```

5.

To remove the standby_ib0 SM standby, first stop its master and then perform the **remove** option, as follows:

```
# sgifmcli --stop --id master_ib0
# sgifmcli --remove --id standby_ib0
```

The SM standby is removed from the entity list. If a standby has been defined, it is removed, in addition to the master.

Configuring the InfiniBand Fat-tree Network Topology

This section describes how to configure InfiniBand fat-tree network topology. The fat-tree topology involves external InfiniBand switches. For the list of supported external switches, see "Fabric Component sgifmcli Command" on page 232.

InfiniBand switches are generally classified as being of two types: edge switches and core or spine switches. Edge switches are used to connect to compute nodes. Core or spine switches are used to connect edge switches together. The integrated InfiniBand switches in SGI ICE X systems are considered to be edge switches and external InfiniBand switches used to connect these edge switches together in a fat-tree topology are considered to be spine switches.

The sgifmcli command allows two types of fat-tree topologies to be configured: FTREE and BFTREE. BFTREE is a balanced fat-tree. If the fat-tree topology is not balanced, choose FTREE; otherwise, choose BFTREE for a balanced fat-tree.

SGI recommends that you use the SMC for ICE X discover command (see "discover Command" on page 84) to discover external IB switches. After discovery is completed, an external switch can also be initialized and added to the InfiniBand system using the sgifmcli command.

The --init and --add options below are completed by the SMC for ICE Xdiscover command when the external switch is discovered with the --switch option. If the external switch is discovered not to be an external switch but as a general node, then the --init and --add options below, need to done.

Procedure 5-3 Configuring InfiniBand Fat-tree Network Topology

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

1. Make sure that your switch is properly connected to the InfiniBand network. Also, make sure that the admin port of the switch is properly connected to the Ethernet network.
2. Power on the switch. See the switch manual for operation information.
3. From the system admin controller (SAC), initialize the switch. The syntax to initialize the switch is, as follows:

```
sgifmcli --init --ibswitch --model   --id  --switchtype [leaf | spine]
```

   An example command is, as follows:

```
# sgifmcli --init --ibswitch --model voltaire-isr-2004  --id isr2004 --switchtype spine
```

   This configures a Voltaire switch ISR2004 with hostname `isr2004` as a spine switch. `isr2004` refers to the admin port of the switch and needs to be configured previously to allow for switch access. The switch is now initialized and the root GUID from the spine switches have been downloaded.

4. From the SAC, add the switch to the fabric. The syntax to add the switch is, as follows:

   ```
   sgifmcli --add --id <fabric> --switch <hostname>
   ```

   An example command is, as follows:

   ```
   # sgifmcli --add --id ib0 --switch isr2004
   ```

   In this example, ISR2004 is connected to the `ib0` fabric.

5. For the new switch to be activated, the SM master and the optional SM standby need to be (re)started.

   ```
   # sgifmcli --start --id master_ib0
   ```

   If the SM master was running while the switch was added, you first need to stop and then start the master, as follows:

   ```
   # sgifmcli --stop --id master_ib0
   # sgifmcli --start --id master_ib0
   ```

   If a standby has been defined, then in case of an SM master failure the SM standby subnet manager will automatically take over and assume control over the switch.

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

   ```
   # sgifmcli --switchlist --id <fabric>
   ```

Configuring the Lightweight Fabric

This section describes how to configure the lighweight fabric with fat-tree topology using external Mellanox switches.

Procedure 5-4 Configuring the Lightweight Fabric

To configure the Lightweight Fabric, perform the following steps:

1. The switch should be setup to use dynamic host configuration protocol (DHCP), as part of the initial setup. This is done by SGI in the factory. You only need to go through the process if a new switch is being installed. For configuration information, see the Mellanox Technologies *IS5025/5030/5031/5035 Installation Guide.* See the section called "Configuring the switch for the First Time". When asked about using DHCP answer "**Yes**". For IP configuration information, see Table 4 - "Configuration Wizard Session - IP Configuration by DHCP".

2. Use the `discover` command, to discover external switches. See "`discover` Command" on page 84. The switch model to be used is "mellanox5030". The `discover` command supports external switches in a manner similar to racks and service nodes, except that switches do not have BMCs and there is no software to install.

3. Discover all external switches.

4. Use `smc-configure-fabric` to configure the fabric, as described in "InfiniBand Management Tool Graphical User Interface" on page 228.

   In the **Configure Topology** option, use **BFTREE** as the topology. The **FAT TREE** topology option should **not** be used. Proceed with the steps, described in "InfiniBand Management Tool Graphical User Interface" on page 228, to configure and verify the fabric.

Verifying the InfiniBand Network

After your InfiniBand fabric has been configured and started, you can use the `sgifmcli`(8) command to verify the health of the fabric.

Procedure 5-5 Verifying the InfiniBand Network

The fabric can be either `ib0` or `ib1`. This version of the InfiniBand verifier runs the recommended OFED test suite. In addition, the SMC for ICE X cluster view is compared with the InfiniBand cluster view and potential differences are reported.

To verify the `ibo` fabric, perform the following command:

sgifmcli --verify --id <*fabric*>

For more information, see the `sgifmcli`(8) man page.

Useful Utilities and Diagnostics

The `infiniband-diags-pp` package contains useful tools and diagnostic software for Open Fabrics Enterprise Distribution (OFED). This section describes some of these tools. These tools reside on the rack leader controller (RLC) in the `/usr/sbin` directory. To see a full list of diagnostics, from the RLC, use the following command:

rpm -ql infiniband-diags-pp | grep bin

This section covers the following topics:

- "`ibstat` and `ibstatus` Commands" on page 245
- "`perfquery` Command" on page 247
- "`ibnetdiscover` Command" on page 249
- "`ibdiagnet` Command" on page 250

`ibstat` and `ibstatus` Commands

You can use the `ibstat` command to see the current status of the host channel adapters (HCA) in your InfiniBand fabric including the HCAs on rack leader controllers (RLCs). The following view is **prior** to starting the fabric management:

```
r1lead:/usr/bin # ibstat
CA 'mthca0'
        CA type: MT25208 (MT23108 compat mode)
        Number of ports: 2
        Firmware version: 4.7.600
        Hardware version: a0
```

```
        Node GUID: 0x0008f104039881a8
        System image GUID: 0x0008f104039881ab
        Port 1:
                State: Initializing
                Physical state: LinkUp
                Rate: 20
                Base lid: 0
                LMC: 0
                SM lid: 0
                Capability mask: 0x02510a68
                Port GUID: 0x0008f104039881a9
        Port 2:
                State: Initializing
                Physical state: LinkUp
                Rate: 20
                Base lid: 0
                LMC: 0
                SM lid: 0
                Capability mask: 0x02510a68
                Port GUID: 0x0008f104039881aa
```

The following shows output from the `ibstat` command **after** the fabric management software has been started:

```
r1lead:/opt/sgi/sbin # ibstat
CA 'mthca0'
        CA type: MT25208 (MT23108 compat mode)
        Number of ports: 2
        Firmware version: 4.7.600
        Hardware version: a0
        Node GUID: 0x0008f104039881a8
        System image GUID: 0x0008f104039881ab
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 20
                Base lid: 1
                LMC: 0
                SM lid: 1
                Capability mask: 0x02510a6a
                Port GUID: 0x0008f104039881a9
        Port 2:
```

```
State: Active
Physical state: LinkUp
Rate: 20
Base lid: 1
LMC: 0
SM lid: 1
Capability mask: 0x02510a6a
Port GUID: 0x0008f104039881aa
```

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

```
r1lead:/opt/sgi/sbin # ibstatus
Infiniband device 'mthca0' port 1 status:
        default gid:     fe80:0000:0000:0000:0008:f104:0398:81a9
        base lid:        0x1
        sm lid:          0x1
        state:           4: ACTIVE
        phys state:      5: LinkUp
        rate:            20 Gb/sec (4X DDR)

Infiniband device 'mthca0' port 2 status:
        default gid:     fe80:0000:0000:0000:0008:f104:0398:81aa
        base lid:        0x1
        sm lid:          0x1
        state:           4: ACTIVE
        phys state:      5: LinkUp
        rate:            20 Gb/sec (4X DDR)
```

Note: If link rate is not 20 Gb/sec 4xDDR, and you have a DDR capable HCA, there is a physical link problem with your system.

perfquery Command

The `perfquery` command is useful for finding errors on a particular HCA (or a number of them) and switch ports. You can also use `perfquery` to reset HCA and switch port counters.

To see a usage statement for the `perfquery` command, perform the following:

```
r1lead:/opt/sgi/sbin # perfquery --help
Usage: perfquery [-d(ebug) -G(uid) -a(ll_ports) -r(eset_after_read) -C ca_name -P ca_port -R(eset_only)
 -t(imeout) timeout_ms -V(ersion) -h(elp)] [<lid|guid> [[port] [reset_mask]]]
        Examples:
                perfquery               # read local port's performance counters
                perfquery 32 1          # read performance counters from lid 32, port 1
                perfquery -e 32 1       # read extended performance counters from lid 32, port 1
                perfquery -a 32         # read performance counters from lid 32, all ports
                perfquery -r 32 1       # read performance counters and reset
                perfquery -e -r 32 1    # read extended performance counters and reset
                perfquery -R 0x20 1     # reset performance counters of port 1 only
                perfquery -e -R 0x20 1  # reset extended performance counters of port 1 only
                perfquery -R -a 32      # reset performance counters of all ports
                perfquery -R 32 2 0x0fff        # reset only error counters of port 2
                perfquery -R 32 2 0xf000        # reset only non-error counters of port 2
```

Some sample output from the `perfquery` command is, as follows:

```
r1lead:/opt/sgi/sbin # perfquery
# Port counters: Lid 1 port 1
PortSelect:......................1
CounterSelect:...................0x0000
SymbolErrors:....................0
LinkRecovers:....................0
LinkDowned:......................0
RcvErrors:.......................0
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................0
XmtDiscards:.....................0
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................0
XmtData:.........................0
RcvData:.........................0
XmtPkts:.........................0
RcvPkts:.........................0
```

ibnetdiscover Command

The `ibnetdiscover` command allows you discover the IB fabric.

To see a usage statement for the `ibnetdiscover` command, perform the following:

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

Note: Only abbreviated output is shown in the this example.

Some sample output from the `ibnetdiscover` command is, as follows:

```
r1lead:/opt/sgi/sbin # ibnetdiscover
#
# Topology file: generated on Tue Jul 17 14:05:20 2007
#
# Max of 3 hops discovered
# Initiated from node 0008f104039881a8 port 0008f104039881a9

vendid=0x2c9
devid=0xb924
sysimgguid=0x8006900000000dd

...

Switch   : 0x08006900000000dc ports 24 devid 0xb924 vendid 0x2c9
"MT47396 Infiniscale-III Mellanox Technologies"
Switch   : 0x08006900000000a4 ports 24 devid 0xb924 vendid 0x2c9
"MT47396 Infiniscale-III Mellanox Technologies"

r1lead:/opt/sgi/sbin # ibnetdiscover -H (HCA's)
Ca       : 0x0030487aa7940000 ports 1 devid 0x6274 vendid 0x2c9 "MT25204 InfiniHostLx Mellanox Technologies"
Ca       : 0x0030487aa78c0000 ports 1 devid 0x6274 vendid 0x2c9 "r1i0n8-ib0 HCA-1"
Ca       : 0x0008f10403988198 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"
Ca       : 0x0030487aa7840000 ports 1 devid 0x6274 vendid 0x2c9 "r1i0n1-ib0 HCA-1"
Ca       : 0x0030487aa79c0000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n0-ib0 HCA-1"
Ca       : 0x0030487aa7900000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n8-ib0 HCA-1"
```

```
Ca       : 0x0030487aa7980000 ports 1 devid 0x6274 vendid 0x2c9 "r1i1n1-ib0 HCA-1"
Ca       : 0x0008f104039881a8 ports 2 devid 0x6278 vendid 0x8f1 " HCA-1"

=========================================================================================
```

ibdiagnet Command

The ibdiagnet command is a useful diagnostic tool.

To see a usage statement for the ibdiagnet command, perform the following:

```
r1lead:/opt/sgi/sbin # ibdiagnet --help
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
NAME
  ibdiagnet
SYNOPSYS
  ibdiagnet [-c ] [-v] [-r] [-o ]
     [-t ] [-s ] [-i ] [-p ]
     [-pm] [-pc] [-P <>]
     [-lw <1x|4x|12x>] [-ls <2.5|5|10>]


DESCRIPTION
  ibdiagnet scans the fabric using directed route packets and extracts all the
  available information regarding its connectivity and devices.
  It then produces the following files in the output directory defined by the
  -o option (see below):
    ibdiagnet.lst    - List of all the nodes, ports and links in the fabric
    ibdiagnet.fdbs   - A dump of the unicast forwarding tables of the fabric
                       switches
    ibdiagnet.mcfdbs - A dump of the multicast forwarding tables of the fabric
                       switches
    ibdiagnet.masks  - In case of duplicate port/node Guids, these file include
                       the map between masked Guid and real Guids
    ibdiagnet.sm     - A dump of all the SM (state and priority) in the fabric
    ibdiagnet.pm     - In case -pm option was provided, this file contain a dump
                       of all the nodes PM counters
  In addition to generating the files above, the discovery phase also checks for
  duplicate node/port GUIDs in the IB fabric. If such an error is detected, it
  is displayed on the standard output.
  After the discovery phase is completed, directed route packets are sent
```

```
  multiple times (according to the -c option) to detect possible problematic
  paths on which packets may be lost. Such paths are explored, and a report of
  the suspected bad links is displayed on the standard output.
  After scanning the fabric, if the -r option is provided, a full report of the
  fabric qualities is displayed.
  This report includes:
    SM report
    Number of nodes and systems
    Hop-count information:
         maximal hop-count, an example path, and a hop-count histogram
    All CA-to-CA paths traced
    Credit loop report
    mgid-mlid-HCAs matching table
  Note: In case the IB fabric includes only one CA, then CA-to-CA paths are not
  reported.
  Furthermore, if a topology file is provided, ibdiagnet uses the names defined
  in it for the output reports.

OPTIONS
  -c                      : The minimal number of packets to be sent
                                   across each link (default = 10)
  -v                             : Instructs the tool to run in verbose mode
  -r                             : Provides a report of the fabric qualities
  -o                    : Specifies the directory where the output
                                   files will be placed (default = /tmp)
  -t                  : Specifies the topology file name
  -s                   : Specifies the local system name. Meaningful
                                   only if a topology file is specified
  -i                  : Specifies the index of the device of the port
                                   used to connect to the IB fabric (in case of
                                   multiple devices on the local system)
  -p                   : Specifies the local device’s port number used
                                   to connect to the IB fabric
  -pm                            : Dumps all pmCounters values into ibdiagnet.pm
  -pc                            : reset all the fabric links pmCounters
  -P <>: If any of the provided pm is greater then its
                                   provided value, print it to screen
  -lw <1x|4x|12x>                : Specifies the expected link width
  -ls <2.5|5|10>                 : Specifies the expected link speed

  -h|--help                      : Prints this help information
```

```
  -V|--version                   : Prints the version of the tool
     --vars                      : Prints the tool’s environment variables and
                                   their values

ERROR CODES
  1 - Failed to fully discover the fabric
  2 - Failed to parse command line options
  3 - Failed to interact with IB fabric
  4 - Failed to use local device or local port
  5 - Failed to use Topology File
  6 - Failed to load required Package
```

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

```
r1lead:/opt/sgi/sbin # ibdiagnet
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdm1.2
-W- Topology file is not specified.
    Reports regarding cluster links will use direct routes.
-W- A few ports of local device are up.
    Since port-num was not specified (-p option), port 1 of device 1 will be
    used as the local port.
-I- Discovering the subnet ... 10 nodes (2 Switches & 8 CA-s) discovered.


-I---------------------------------------------------
-I- Bad Guids Info
-I---------------------------------------------------
-I- No bad Guids were found

-I---------------------------------------------------
-I- Links With Logical State = INIT
-I---------------------------------------------------
-I- No bad Links (with logical state = INIT) were found

-I---------------------------------------------------
-I- PM Counters Info
-I---------------------------------------------------
-I- No illegal PM counters values were found

-I---------------------------------------------------
```

```
-I- Bad Links Info
-I---------------------------------------------------
-I- No bad link were found

-I- Done. Run time was 0 seconds.
```

You can use `ibdiagnet` to load the fabric to test it, as follows:

```
r1lead:/opt/sgi/sbin # ibdiagnet -c 5000
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
Loading IBDM from: /usr/lib64/ibdm1.2
-W- Topology file is not specified.
    Reports regarding cluster links will use direct routes.
-W- A few ports of local device are up.
    Since port-num was not specified (-p option), port 1 of device 1 will be
    used as the local port.
-I- Discovering the subnet ... 10 nodes (2 Switches & 8 CA-s) discovered.


-I---------------------------------------------------
-I- Bad Guids Info
-I---------------------------------------------------
-I- No bad Guids were found

-I---------------------------------------------------
-I- Links With Logical State = INIT
-I---------------------------------------------------
-I- No bad Links (with logical state = INIT) were found

-I---------------------------------------------------
-I- PM Counters Info
-I---------------------------------------------------
-I- No illegal PM counters values were found

-I---------------------------------------------------
-I- Bad Links Info
-I---------------------------------------------------
-I- No bad link were found

-I- Done. Run time was 8 seconds.
```

System Maintenance, Monitoring, and Debugging

This chapter describes system monitoring and covers the following topics:

Maintenance Procedures

This section describes some common maintenance procedures, as follows:

Temporarily Take a Node Offline for Maintenance

This section describes how to temporarily take a node offline for maintenance.

Procedure 6-1 Temporarily Take a Node Offline for Maintenance

To temporarily Take a node offline for maintenance, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

```
# cpower --down r1i0n0
```

3. Mark the node offline, as follows:

```
# cadmin --set-admin-status --node r1i0n0 offline
```

4. Perform any maintenance to the blade that needs to be done.
5. Mark the node online, as follows:

```
# cadmin --set-admin-status --node r1i0n0 online
```

6. Power up the node, as follows:

```
# cpower --boot r1i0n0
```

7. Enable the node in the batch scheduler (depends on your batch scheduler).

Permanently Replace a Failed Blade

Note: See your SGI field support person for the physical removal and replacement of SGI ICE X compute nodes (blades).

This section describes how to permanently replace a failed blade.

Procedure 6-2 Permanently Replace a Failed Blade

To permanently replace a failed blade (compute node), perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

```
# cpower --down r1i0n0
```

3. Mark the node offline, as follows:

   ```
   # cadmin --set-admin-status --node r1i0n0 offline
   ```

4. Physically remove and replace the failed blade.

5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon. See "Discovering Compute Nodes" on page 96, for more information.

6. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 169 for your options), as follows:

   ```
   # cimage --set mycomputeimage mykernel r1i0n0
   ```

7. Power up the node, as follows:

   ```
   # cpower --boot r1i0n0
   ```

8. Enable the node in the batch scheduler (depends on your batch scheduler).

Permanently Remove a Blade

This section describes how to permanently remove a blade from your SGI ICE X system.

Procedure 6-3 Permanently Remove a Blade

To permanently remove a blade from your system, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).

2. Power off the node, as follows:

   ```
   # cpower --down r1i0n0
   ```

3. Mark the node offline, as follows:

   ```
   # cadmin --set-admin-status --node r1i0n0 offline
   ```

4. Physically remove the failed blade.

5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon. See "Discovering Compute Nodes" on page 96, for more information.

Add a New Blade

This section describes how to add a new blade to an SGI ICE X system.

Procedure 6-4 Add a New Blade

To add a new blade to your system, perform the following steps:

1. Physically insert the new blade
2. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon. See "Discovering Compute Nodes" on page 96, for more information.
3. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 169 for your options), as follows:

   ```
   # cimage --set mycomputeimage mykernel r1i0n0
   ```

4. Power up the node, as follows:

   ```
   # cpower --boot r1i0n0
   ```

5. Enable the node in the batch scheduler (depends on your batch scheduler).

Node Replacement Procedure for Cold Spare System Admin Controller (SAC), Rack Leader Controller (RLC), or Service Nodes

This section describe how to install and configure a spare SAC, RLC, or managed service node. The cold spare can be a shelf spare or a factory-installed cold spare that ships with your system. For more information on cold spare requirements and tools needed to do this procedure, see "Cold Spare System Admin Controller (SAC) or Rack Leader Controler (RLC) Availability" on page 259.

It covers the following topics:

- "Cold Spare System Admin Controller (SAC) or Rack Leader Controler (RLC) Availability" on page 259
- "Identify the Failed Unit and Unplug all Cables" on page 260
- "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264

- "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 266
- "Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode" on page 269

Note: When ordering shelf spare systems from SGI, it is important to order spare nodes appropriate to or in conjunction with your SGI ICE X system. This is because the SGI ICE serial number is programmed into the SAC itself. If you try to migrate the SAC to a shelf spare system that does not have the correct SGI ICE system serial number programmed into it, parts of Tempo software may not work correctly.

Depending on the system ordered, your SGI ICE X system should be mounted in an SGI rack or racks. The SAC and RLC are generally installed within (or in some cases on top of) the system rack. For an example, see Figure 1-2 on page 3. The replacement of a failed SAC or RLC is accomplished in four basic steps:

- Identify the failed unit and disconnect system and power cables.
- Transfer the disk drives from the failed server into the cold spare unit.
- Connect the applicable cables to the cold spare server.
- Power-up the new server and restart the ICE system.

For detailed procedures on installing a cold spare, see sections"Identify the Failed Unit and Unplug all Cables" on page 260, "Transfer Disks from Existing Server to the Cold Spare" on page 263, "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264 and "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 266.

Note: If you are using multiple root slots (making use of cascading dual-boot as described in "Booting the System" on page 52) the procedures described in this section will have to be repeated for each slot.

Cold Spare System Admin Controller (SAC) or Rack Leader Controler (RLC) Availability

A cold spare node is like an existing SAC or RLC, but it sits on a shelf or is a factory preinstalled node to be used in an emergency.

If the SAC or RLC node should fail, the cold spare can be swapped in to position to take over the duities of the failed node.

If you wish to make use of cold spare nodes, SGI suggests that you have both a SAC and an RLC on the shelf as available spares. Some of the reasons to have two separate nodes instead of one are (not an exhaustive list), as follows:

- The BIOS settings of a SAC and an RLC are different. For example, a SAC does not PXE boot by default. However, an RLC must PXE boot each boot. This means that the boot order is different for each type.
- The BMC of an RLC is set up to use DHCP by default. A SAC may not be set up this way.
- Given the first two items in this list, if you try to use a shelf-spare SAC as an RLC, the RLC is not discovered properly.

Shelf Spare Hardware Limitations

Currently, the hardware replacement procedure described in this section only supports SGI ICE X `ice-csn` nodes, that is, system admin controller (SAC), rack leader controller (RLC), and managed service nodes.

Tools Required

You will need a Video Graphics Array (VGA) screen and a keyboard to perform this procedure. This is because you need to interact with the LSI BIOS tool to import the root volumes. You cannot do this from an Intelligent Platform Management Interface (IPMI) serial console session because of the following:

- For rack leader controllers (RLCs), the cluster does not know the MAC addresses of the replacement BMC so there is no way for the cluster to connect to it until the migration script is run.
- The LSI BIOS tool requires the use of `Alt` characters which often do not transfer through the serial console properly.

Identify the Failed Unit and Unplug all Cables

If you identified the failed system admin controller (SAC) or rack leader controller (RLC), disconnect the cables from the failed unit. The front panel lights on the server can indicate if the unit has failed and give you information on why, see Figure 6-1 on page 261.

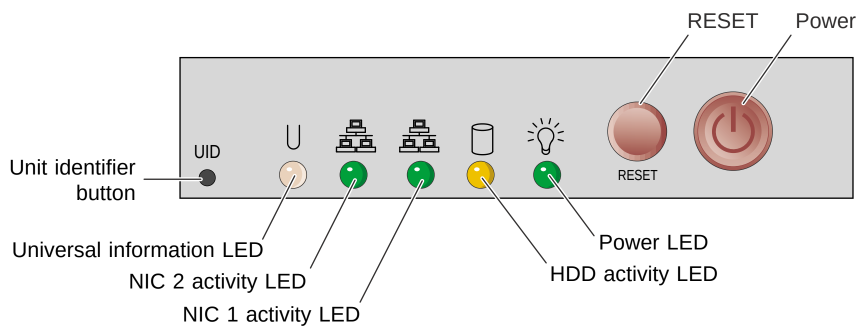

Figure 6-1 Admin/RLC Server Front Panel Controls and Indicator LEDs

The universal information LED (left side of the panel) shows two types of failure that can bring the server down. This multi-color LED blinks red quickly to indicate a fan failure and blinks red slowly for a power failure. A continuous solid red LED indicates a CPU is overheating.

If the unit's power supply has failed or been disconnected, the power LED (far right) will be dark. Check both ends of the power cable for a firm connection prior to switching over to the cold spare.

If you find that a SAC or RLC has failed and you need to replace it with a cold spare system, this section describes what to do in terms of the physical hardware.

The SAC stores the system-wide serial number. The SAC shelf spares must be ordered from the factory as SAC shelf spares so that the proper serial number can be stored within.

Procedure 6-5 Replacing a Node with a Cold Spare: Installing the Hardware

To replace a SAC or RLC that has failed, perform the following steps:

1. Power down the failed node (if possible).
2. Disconnect both power cables, see Figure 6-2 on page 263 for server connection locations.

3. Remove the two system disks from the failed node and set them aside for later reinstallation.
4. Unplug the Ethernet cable used for system management (be sure to note the plug number. Label the cables to avoid confusing them. It is important that they stay in the same jacks in the new node). See the example drawing in Figure 1-4 on page 6. This connection is vital to proper system management and communication. **The Ethernet cable must be connected to the same plug on the cold spare unit.**
5. If the unit has a system console attached, remove the keyboard, mouse, and video cables.
6. Remove the system from the rack.
7. Install the shelf spare system into the rack.
8. Install the system disks you set aside in step 3 (from the system you are replacing).
9. Connect the Ethernet cables in the same way they were connected to the replaced node.
10. Connect AC power.
11. Connect a keyboard and VGA monitor (and mouse if you like).
12. Do **NOT** power up the system just yet. Proceed to "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264.

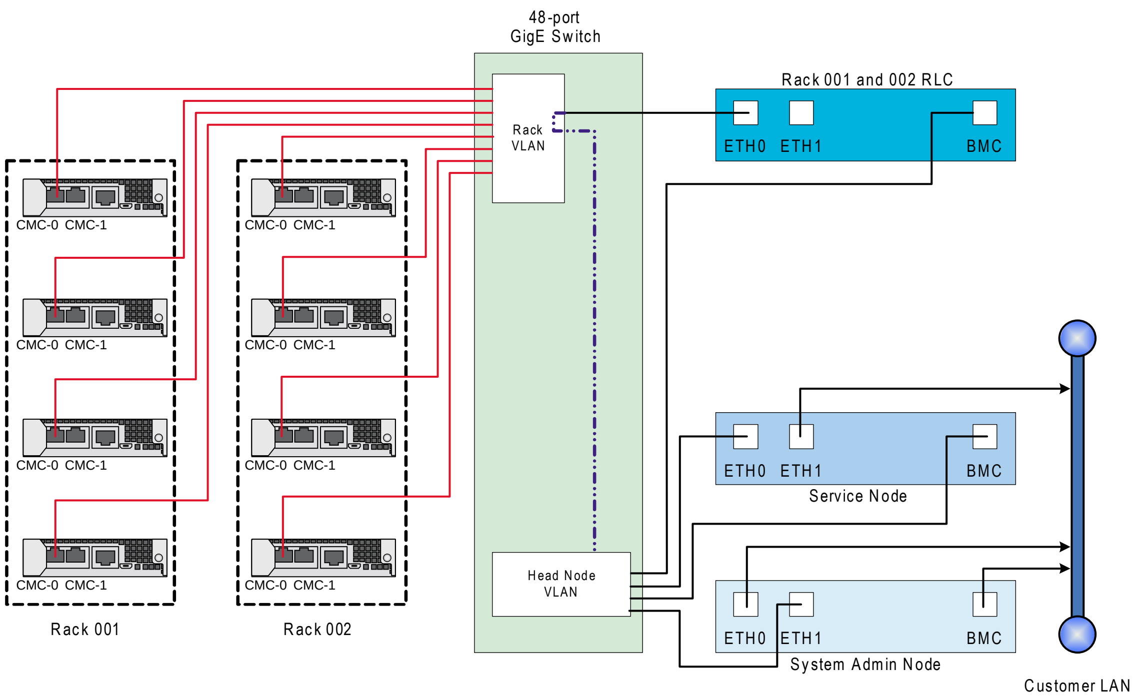

Figure 6-2 Simple CMC LAN (VLAN) Cable Examples

Transfer Disks from Existing Server to the Cold Spare

Note: The factory-installed cold spare does NOT ship with disks so you need to transfer existing disks and PCI cards from the existing server to the cold spare before mounting the spare rack.

Transfer disks from the existing server to the cold spare as shown in Figure 6-3 on page 264.

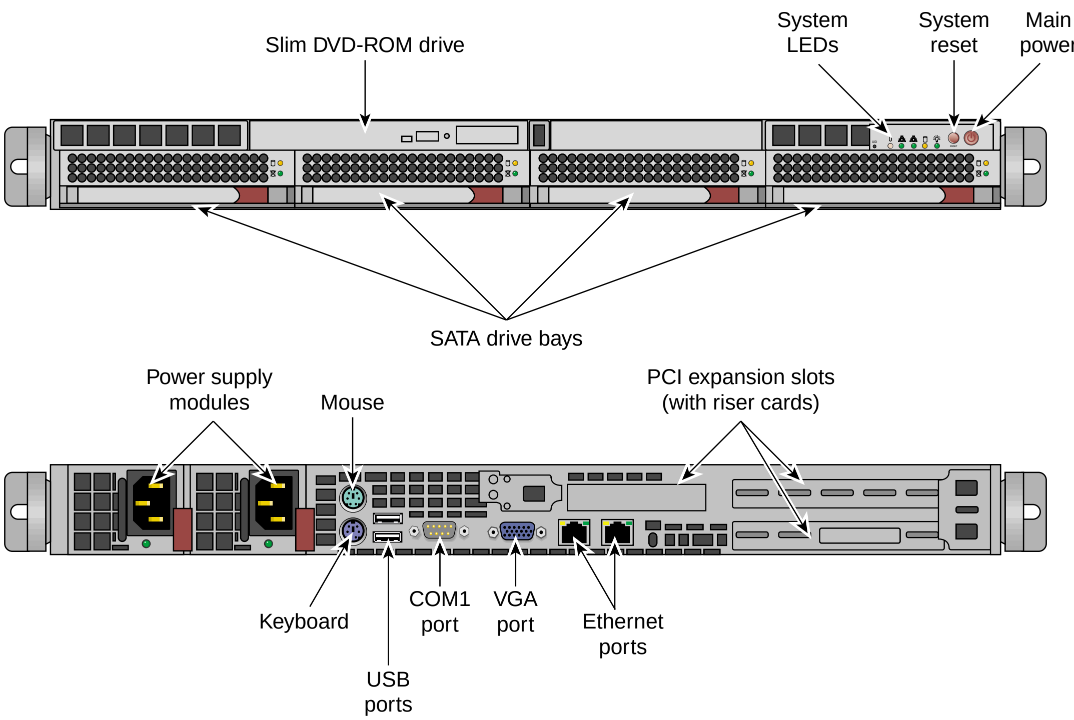

Figure 6-3 SAC and RLC Server Front Features and Rear Connector Locations

Migrating to a Cold Spare: Importing the Disk Volumes

This section describes how to import the disk volumes into the new node installed in "Identify the Failed Unit and Unplug all Cables" on page 260. For LSI 106x based systems, follow the procedure below. For LSI MegaRAID based systems, When you import the disk pair from the dead system to the new one, they will automatically be imported.

Warning: You must use the same class of system for the shelf spare. That is to say, you cannot move disks formatted with LSI 106x RAID to a megaRAID based system and import the volume.

Although not supported, going from megaRAID to LSI 106x may allow manual importing of the data without data loss.

Procedure 6-6 Migrating to a Shelf Spare: Importing the Disk Volumes

To import the disk volumes into the new node, perform the following steps:

1. At this time, you can power up the system using the power button.
2. Watch the VGA screen output.
3. When you see the LSI BIOS tool come up up, enter `Ctrl-C`. This will instruct the LSI BIOS tool to enter the configuration utility.
4. A screen appears listing the LSI controllers in the system. Normally, there is just one. Hit the `Enter` key to proceed.
5. Choose **RAID Properties**.
6. It is important to note that the controller supports only two RAIDs at a time. Therefore, if the system had two volumes at a time in the past, one or more volumes may appear empty now. It is important to use the utility to delete these empty volumes representing disks that are no longer installed before proceeding. Otherwise, if the tool sees more than one volume, activating volumes will not work.
7. Enter `Alt-N` to browse the list of volumes. Delete the empty ones as described in the step, above. Eventually, you will encounter an inactive volume. This inactive volume represents the disks you migrated from the failed node to this node.
8. With the inactive volume selected, choose **Manage Array**.
9. Choose **Activate** and answer **y** to the **activate and exit this menu** choice.
10. At this point, especially if the node has more than one volume, it is important to select the migrated system disk volume as the boot volume. To select the boot volume, choose **SAS Topology**.
11. In **SAS Topology**, you can expand the volumes to see the disks within them if you choose by hitting `Enter` on volumes.
12. Choose the volume that represents your newly imported volume. Highlight it, then enter `Alt-B`.
13. You should see that the volume now has a **Boot** flag associated with it.

Note: If, after you exit the tool, the system does not appear to boot from the disk. You may have selected the wrong volume from which to boot. In that case, reset, re-enter the LSI BIOS Tool, and choose a different volume to be the boot volume.

---

14. Escape out of the LSI tool and exit.
15. Keep watching the VGA screen! You will have to hit a key at the correct moment in the next section. Go to "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 266.

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

This section provides information on booting the system for the first time on a replacement node.

Note: Important: If your site is using cascading dual boot, only the currently used slot will be updated or repaired. Therefore, if the system admin controler (SAC) is booted to slot 2, the fix up operations documented in these sections only apply to slot 2. The instructions need to be done for each slot you wish to fix up.

In a prior release, automatic recovery was implemented for cascading dual boot clusters. This means, if cascading dual boot is in use, when a managed service node or rack leader controller (RLC) boots after having procedure 5-6 performed, it will go in to an automatic recovery boot, perform some fix up, then reboot again in to its normal operating mode. For the case of the SAC, a script is run by hand to integrate the repaired SAC with the cluster.

Note: Automatic Recovery is disabled by default because it can make certain `discovery` operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]#  cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, whend doing `discovery` operations, you may find it convenient to disable it.

For the case of the SAC, you will need to ensure your console output goes to the VGA screen and not serial-over-lan (SOL). For managed service nodes and RLCs in cascading dual boot clusters, the default output location during the auto recovery boot is VGA. It is best to leave it VGA since part of the repair procedure will affect the network configuration for the BMC.

How do I know which procedure to follow?

- SACs, all cases: Procedure 6-7, page 267.
- Managed Service nodes and RLCs in a non-cascading dual boot cluster: Procedure 6-7, page 267.
- Managed service, RLCs in a cascading dual boot cluster: Procedure 6-8, page 269.

Procedure 6-7 Migrating to a Cold Spare in a Non-cascading Dual Boot Cluster Node

This section describes how to boot the SAC or RLC or service node in non-cascading dual boot clusters.

Note: This section applies to SACs and sites that are **not** making use of cascading dual boot. Cascading dual boot is set up by default on newer SMC for SGI ICE X software releases. If you are using cascading dual boot, follow these instructions **only** for the SAC.

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

1. Ensure that the VGA console is powered on.
2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264).

   Note: After rebooting, drive 1 will resync with drive 0, automatically. Drive 1 will have the RED LED on during this time. This process takes from eight to 48 hours depending on the drive size. During that period, the RAID redundancy is not available but the system will function normally.

   When you see the GRUB boot menu come up, the first boot option will be highlighted by default. This should NOT be the choice starting with Failsafe. As an example, in SMC for SGI ICE 1.5 the highlighted choice should be : **SUSE**

Linux Enterprise Server 11 SP1. Enter **e** to edit the boot parameters for this boot only.

3. Enter **e** to edit the kernel parameters.
4. Arrow down once so that the line starting **kernel** is highlighted.
5. Look at the settings. If no serial console is defined, you do not need to change anything. If a serial console is defined, append "`console=tty0`" to the end of the parameter list. This will ensure that console output goes to the VGA screen for this boot.

   Note: By default, the SAC goes to the VGA screen. Therefore, this adjustment does not need to be made. RLCs and service nodes have serial consoles by default.

   ---

6. Press the `Enter` key.
7. Enter **b** to boot the system.

   The system will now boot with console output going to the VGA screen.

   Networking will fail to start and some error messages will appear.

   It is normal to see that the Ethernet devices were renumbered. This will be fixed below.

   Eventually the login prompt will appear.

8. Log in as root.
9. The following script fixes the network settings and update the SMC for ICE X database for the new network interfaces, as follows:

   \# **`migrate-to-shelf-spare-node`**

   Note: If you have additional Ethernet cards installed, you may need to check the settings of interfaces not controlled or managed by SMC for X ICE software.

   ---

10. Reboot the node and let it boot normally.

Procedure 6-8 Migrating to a Cold Spare: Service Node or or RLC Using Cascading Dual Boot

This section describes what to do for managed service nodes and RLCs in a cluster making use of cascading dual boot. It does **not** apply to SACs. For SACs, see Procedure 6-7, page 267.

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

1. Ensure that the VGA console is powered on.
2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264).

   Note: After rebooting, drive 1 will resync with drive 0, automatically. Drive 1 will have the RED LED on during this time. This process takes from eight to 48 hours depending on the drive size. During that period, the RAID redundancy is not available but the system will function normally.

   ---

3. At this time, you can plug the node in to AC power and press the power button on the front of the node.
4. Watch the VGA screen. The system should network boot in to recovery mode. It will do some repairs and reboot itself.
5. At this point, it will boot as a normal node. If, for some reason, it is unable to boot from the disk, the wrong volume may be selected as the boot disk in the LSI BIOS tool (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 264). It is true that the node network boots, but the network boot does a chainload to the first disk and it is still impacted by the BIOS and LSI firmware settings.

Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode

This section gives some advanced details on the Auto Recovery feature including how it is set up and how to control the feature.

Overview

The auto recovery feature allows managed service nodes and rack leader controllers (RLCs) to automatically make the necessary adjustments for both the node setup itself and the SMC for SGI ICE X cluster database. This feature is mainly useful for clusters making use of cascading dual boot. The automated recovery mode applies to managed service nodes and RLCs in cascading dual boot clusters. The goal is to provide an easy way for these nodes to perform any fix ups to themselves and the SMC for SGI ICE X cluster at large when faulty systems are replaced.

Enable or Disable Auto Recovery Mode

Note: Automatic Recovery is disabled by default because it can make certain `discovery` operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]#  cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, when performing `discovery` operations, you may find it convenient to disable it.

Use the `cadmin --show-auto-recovery` command to show the current state. Use the `cadmin --disable-auto-recovery` command to disable it.

IP Addresses Reserved for Auto Recovery Mode

Four IP addresses are reserved on the head network for auto recovery operations. For clusters being installed for the first time, these tend to be low numbers as they are reserved before any service nodes or rack leader controllers (RLCs) are discovered. For systems being upgraded from previous SMC for SGI ICE X releases, the allocated IP addresses are allocated the first boot after the upgrade and would tend to have higher numbers.

DHCP Set Up for Auto Recovery Mode

When the auto recovery feature is enabled, the `dhcpd.conf` file is configured with DHCP addresses available to unknown systems. That is, when this mode is enabled, any system attached to the head network that is performing DHCP requests will get a generic pool address and then boot in to the auto recovery mode. When the auto recovery mode is disabled, DHCP is configured to not offer these special IP addresses.

Auto Recovery and the discover Command

The auto recovery mode conflicts with the way that the discover command operates by default. Therefore, the discover command automatically and temporarily disables auto recovery (if it was enabled) for the duration of the run of the discover command. For more information on the discover command, see "discover Command" on page 84.

If you plan to discover a node, start discover before applying AC power. This is because auto recovery provides IP addresses to unknown nodes and because the discover command temporarily disables this, it is best to start the discover command before plugging in AC power to the node being discovered. Otherwise, it may get an unintended IP address.

How To Avoid Out of Memory Occurrences on SLES11 When Using the PBS Professional Batch Scheduler

SGI ICE X is a diskless blade server typically configured with nfs root and a small (50 MB) swap space that is served via iscsi. A maximum of 64 blades boot from a rack leader controller (RLC). The RLC typically has SATA disks in a mirrored pair for blade filesystems and blade swap space. Some users turn off swap entirely because a full rack of blades swapping has proven to be stressful to the RLCs. When a Linux system has more memory requests than it can provide the kernel takes steps to defend the system using the out of memory (OOM) killer. The following section describes strategies for avoiding the loss of ICE blades due to OOM occurrences when the operating system is SLES11 and the batch scheduler is PBS Professional.

Some general guidelines are, as follows:

- Make sure that your application requests the proper amount of memory.
- After you ensure that your application asks for memory correctly, configure the pbs_mom process in PBS Professional to enforce memory limits. See your PBS Professional documentation for a complete description of the pbs_mom process.

  This only works well when the SGI memacct function is installed to properly compute the amount of memory used. This requires that Linux kernel jobs and Comprehensive System Accounting (CSA) are installed. For more information, see the *Linux Resource Administration Guide*. CSA does not have to be configured to log. Modify /var/spool/PBS/mom_priv/config file by adding $enforce mem to the file. As an example, an application that just allocates memory one megabyte at a time will be killed once it goes over the limit. Applications that allocate in bigger chunks can still get above the limit before PBS can kill the job.

- The PBS Pro enforce `mem` variable has no configuration options. To avoid OOM occurrences you need your own daemon, such as the `policykill` daemon.

  The `policykill` daemon looks for swapping in cpusets and works well in both large single-system image (SSI) with multiple cpusets and cluster (single cpuset). On large SSI, use of PBSPro's cpuset mom is required. On SGI ICE X systems use of SGI Altix bundle (example `PBSPro_10.1.0-SGIAltix_pp6_x86_64.tar.gz`) from Altair Engineering, Inc. is suggested. `policykill` has an `init` script, configuration file and daemon process itself. It requires customization for limits and notification methods.

- The Linux kernel Out Of Memory killer (`mm/oom_kill.c`) is responsible for keeping the system alive when memory has been exhausted. A snippet from the code is, as follows:

```
* The formula used is relatively simple and documented inline in the
* function. The main rationale is that we want to select a good task
* to kill when we run out of memory.
*
* Good in this context means that:
* 1) we lose the minimum amount of work done
* 2) we recover a large amount of memory
* 3) we don't kill anything innocent of eating tons of memory
* 4) we want to kill the minimum amount of processes (one)
* 5) we try to kill the process the user expects us to kill, this
*    algorithm has been meticulously tuned to meet the principle
*    of least surprise ... (be careful when you change it)
```

  You can use `arrayd` to manage what processes gets killed. For more information on `arrayd`, see the `arrayd`(8) man page and Chapter 3. "Array Services" of the *Linux Resource Administration Guide.* `arrayd` has a configuration option to protect the daemon:

```
-oom oom_daemon,oom_child
Specify oom_adj ( OutOfMemory Adjustments ) respectively for the main arrayd da
arrayd  children.  The  default  is  "-17,0", hence resulting in the arrayd dae
selected as a candidate by the oom kernel killer thread and children selected a
dates. The value range from -17 to 15.
```

Each pid has an oom_adj (/proc//oom_adj) that you can independently protect. In general, you want root owned processes to be protected and user processes to be able to be killed.

A combination of PBS prologue and cron can set the values at job start and through the job's life span. On SMC for SGI ICE X systems, cron is configured off in 80-compute-distro-services which is in

```
/var/lib/systemimager/images/<your compute image>/etc/opt/sgi/conf.d/80-compute-distro-services
```

by commenting out the following line:

```
initDisableServiceIfExists cron
```

To just enable cron on a blade is **not** a good practice. Files in

```
/var/lib/systemimager/images/<your compute image>/etc/cron*
```

must be reviewed for correctness in mixed writeable and read-only environment. For example, sysstat, logrotate, suse.de-cron-local, are the only services available in /etc/cron* directories. For a list of sample scripts, see Appendix A, "Out of Memory Adjustment" on page 317.

- Virtual memory sysctl tuning tries to balance use of system resources for user jobs and for system threads. The default setup is skewed towards user jobs but in the face of OOM system threads need more resources. For more information on sysctl, see the sysctl(8) man page. For an SMC for SGI ICE X system, the sysctl parameters might be predefined similar to the following:

```
# Give the kernel a bit more breathing room by requiring more free space
vm.min_free_kbytes = 131072
# Push dirty pages out faster
vm.dirty_expire_centisecs = 1000          # Default is 3000
vm.dirty_writeback_centisecs = 500        # Default (unchanged)
vm.dirty_ratio = 20                       # Default is 40
vm.dirty_background_ratio = 5             # Default is 10
```

If blades are run without swap, set the following variable:

```
vm.swappiness = 0
```

Inventory Verification Tool

You can use the SMC for SGI ICE XI inventory verification tool to query, take snapshots, analyze and compare the node and network inventory of a cluster. Various hardware, network and operating system configuration properties are available and are presented in user-specified formats.

Note: If you are reinstalling the system admin controller (SAC), you may want to make a backup of the cluster configuration snapshot that comes with your system so that you can recover it later. You can find it in the `/opt/sgi/var/ivt` directory on the SAC; it is the earliest snapshot taken. You can use this information with the interconnect verification tool (IVT) to verify that the current system shows the same hardware configuration as when it was shipped. For more information, see "Installing SMC for ICE X System Admin Controller (SAC) Software" on page 91.

To make an inventory snapshot of an SGI ICE X system, use the following command from the SAC.

```
admin:~ # ivt -M
Making a cluster inventory snapshot.  Takes a couple of minutes...
```

Each snapshot is assigned a unique number and marked with the date and time it was taken. Use the `ivt --L` command to list active snapshot information, as follows:

```
admin:~ # ivt -L
    1   2007-07-13.11:42:47
```

You can query (`-Q` option), compare (`-C` option) and analyze (`-S` option) existing snapshots. A variety of system hardware and configuration properties can be displayed. You can compare two snapshots to see what has changed or analyze a system snapshot for failed nodes and or see network fabric links.

You can use the `ivt -c cpu` command to show an inventory of the system compute blades and the number of CPUs each blade contains, as follows:

```
admin:~ # ivt -c cpu
r1i0n0 has 8 CPUs
r1i0n1 has 8 CPUs
r1i0n8 has 8 CPUs
r1i1n0 has 8 CPUs
r1i1n1 has 8 CPUs
r1i1n8 has 8 CPUs
```

You can use the ivt tool to determine which compute nodes (blades) are up or down, as follows:

```
admin:~ #  ivt -Q -w blades -f '$blade $sshstate'
r1i0n0 up
r1i0n1 down
r1i0n8 up
r1i1n0 up
r1i1n1 down
r1i1n8 up
```

You can use the ivt tool to determine the GigE Ethernet address for each compute node (blade) , as follows:

```
admin:~ # ivt -Q -w blades -f '$blade $gige_ip_addr'
r1i0n0 192.168.159.10
r1i0n1 192.168.159.11
r1i0n8 192.168.159.18
r1i1n0 192.168.159.26
r1i1n1 192.168.159.27
r1i1n8 192.168.159.34
```

For detailed information on how to use the ivt tool, see the ivt(8) man page or ivt -h, --help usage statement.

System Monitoring Overview

Ganglia is a scalable, distributed monitoring system for monitoring system for high-performance computing systems, such as the SGI ICE X system. It displays web browser-based, real-time (on demand) histograms of system metrics, as shown in Figure 6-4 on page 276.

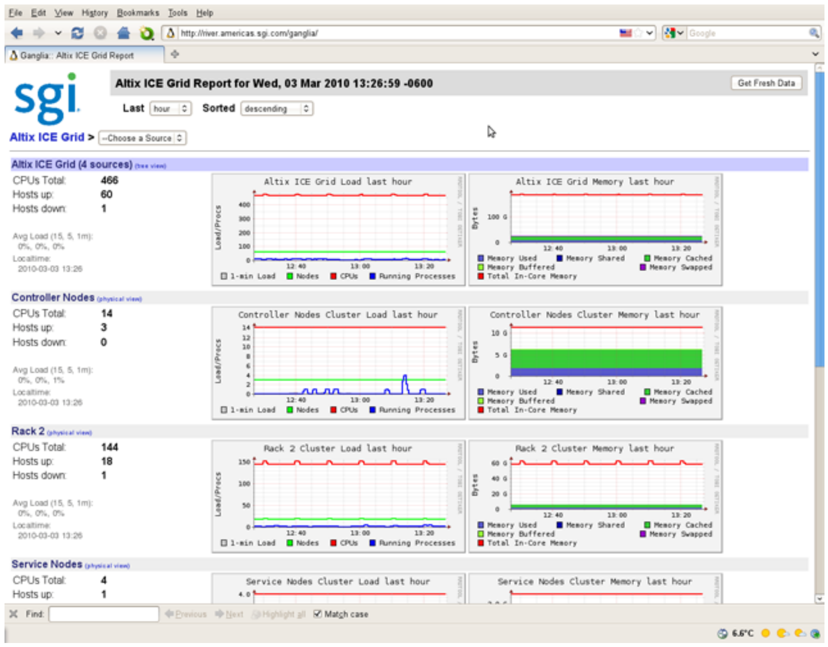

Figure 6-4 Ganglia System Monitor

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

SMC for SGI ICE X has devised a Ganglia model for the SGI ICE X 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 (RLC). Therefore, the RLC receives, at most, data from 64 blades. After collecting the data, the RLC forwards aggregated rack statistics to the system admin controller (SAC). The RLC also sends its own statistics to the SAC. The SAC presents the meta-aggregator for the entire SG ICE X system. It collects data from all RLCs and presents the cluster-wide metrics. This model enables SGI to scale-out Ganglia to very large cluster deployments.

The **Node View** as shown in Figure 6-5 on page 277 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.

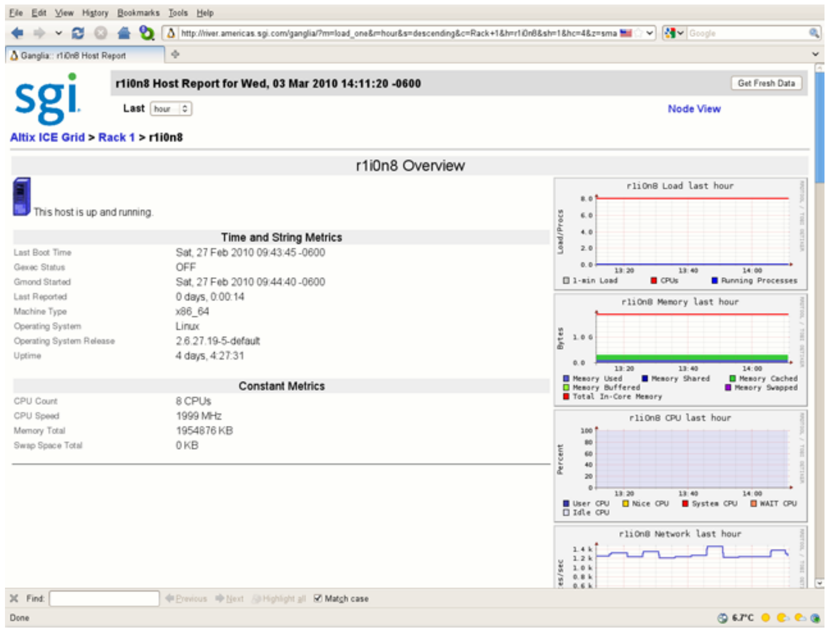

Figure 6-5 Ganglia System Monitoring Node View

System Monitoring Operation

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

- "Accessing the Ganglia System Monitor" on page 277
- "Monitoring System Metrics" on page 278
- "SEL/Hardware Event Monitoring" on page 278
- "Node Availability Monitoring" on page 279

Accessing the Ganglia System Monitor

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

Monitoring System Metrics

By default, Ganglia monitors standard operating system metrics like CPU load, memory usage. The **Grid Report** view shows an overview of your system, such as the number of CPUs, the number of hosts (compute nodes) that are up or down, service node information, memory usage information, and so on.

The **Last** pull down menu allows you to view performance data on an hourly, daily, weekly, or yearly basis. The **Sorted** pull down menu allows provides an ascending, descending, or by host view of performance data. The **Grid** pull-down menu allows you to see performance data for a particular rack or service node. The **Get Fresh Data** button allows you to see current data performance.

SEL/Hardware Event Monitoring

The system admin controller (SAC), rack leader controllers (RLCs), the service nodes, the chassis management controllers (CMCs), and all the compute nodes (blades) are equipped with a specialized controller, called the Board Management Controller (BMC). This unit provides a broad set of functions as described in the IPMI 2.0 standard. SMC for SGI ICE X software uses the BMCs predominantly for remote power management, remote system configuration, and for gathering critical hardware events.

Currently, critical hardware events are gathered for the following nodes: RLCs, CMCs, and compute nodes (blades). These events are logged in the following locations:

- `/var/log/messages` via `syslog`
- `var/log/sel/sel.log`

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

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

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*event*>	`BMC_CMC`
<*app*>	SEL-LOGGER
<*date*>	date / time of the event
<*version*>	1.0

<*text*>	Exact copy of the hardware event description from the BMC

After reading the events from the BMCs, the BMC event logs are cleared on the controller to avoid duplicate events.

Node Availability Monitoring

The availability of each node in an SGI ICE X system is monitored by a lightweight daemon called `smchbc`. Each managed service node, rack leader controller (RLC), and compute node runs this daemon and reports its status to the server which monitors it. The server daemon, which runs on the system admin controller (SAC) and RLC, reports if the client is down after approximately 120 seconds. In this event, administrator-derived actions can be triggered, for instance sending an e-mail notification to the system administrator.

The HEARTBEAT event contains the following useful information:

`MSG ::=` <*syslog-prefix*> `SMC:`<*node*> `EVENT:HEARTBEAT APP:SMCHBD Date:`<*date*> `VERSION:1.0 TEXT` <*text*>

The HEARTBEAT event is created when nodes fail or recover, described by the `TEXT` field.

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*date*>	date / time of the event
<*text*>	Description of event: `'Heartbeat not detected'` `'Heartbeat lost'`

Monitoring System Metrics with Performance Co-Pilot

A wealth of system metrics are also available through the Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The Performance Co-Pilot collection daemon (PMCD) runs on the system admin controller (SAC), rack leader controllers (RLCs), and managed service nodes. A performance metrics domain agent (PMDA) is running on the RLCs, which collects metrics from the compute nodes.

The new cluster metrics domain contains metrics that were previously available in other PMDAs. The method in which they are collected is different in a SMC for ICE X system, in order to minimize load on the compute nodes. The following metrics are available for each compute node in a system by querying the PMCD on their RLC:

```
admin:~ # pminfo -h r1lead cluster
cluster.control.suspend_monitoring
cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
cluster.network.ib.in.bytes
cluster.network.ib.in.errors.drop
cluster.network.ib.in.errors.filter
cluster.network.ib.in.errors.local
cluster.network.ib.in.errors.remote
cluster.network.ib.out.bytes
cluster.network.ib.out.errors.drop
cluster.network.ib.out.errors.filter
cluster.network.ib.total.errors.link
cluster.network.ib.total.errors.recover
cluster.network.ib.total.errors.integrity
cluster.network.ib.total.errors.vl15
cluster.network.ib.total.errors.overrun
cluster.network.ib.total.errors.symbol
```

Configuring Compute Blade Metrics

The list of metrics that are monitored by the compute node and are pushed to the PMCD on the rack leader controller (RLC) is configurable. In some cases , it may be even be desirable to disable metric collection entirely, as follows:

```
# cexec --head --all pmstore cluster.control.suspend_monitoring 1 pmstore -h r1lead cluster.control.suspend_monitorin
```

The default list of metrics that are collected by each compute node contains 41 metrics. There are dozens more available in the `cluster.* namespace`. The default list is stored on each RLC in the `/var/lib/pcp/pmdas/cluster/config` file. Changing this file will allow you to modify the default metric list with rack granularity. To change the list on a single node store a newline-delimited list of metrics to the node's instance of the `cluster.control.metrics` metric.

To see the current metric list for a compute node, perform the following:

```
# pmval -h r1lead -s 1 -i 'r1i1n0' cluster.control.metrics

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

               r1i1n0
"cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.nice
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.infiniband.port.rate
cluster.infiniband.port.in.bytes
cluster.infiniband.port.in.packets
```

```
cluster.infiniband.port.in.errors.drop
cluster.infiniband.port.in.errors.filter
cluster.infiniband.port.in.errors.local
cluster.infiniband.port.in.errors.remote
cluster.infiniband.port.out.bytes
cluster.infiniband.port.out.packets
cluster.infiniband.port.out.errors.drop
cluster.infiniband.port.out.errors.filter
cluster.infiniband.port.total.bytes
cluster.infiniband.port.total.packets
cluster.infiniband.port.total.errors.drop
cluster.infiniband.port.total.errors.filter
cluster.infiniband.port.total.errors.link
cluster.infiniband.port.total.errors.recover
cluster.infiniband.port.total.errors.integrity
cluster.infiniband.port.total.errors.vl15
cluster.infiniband.port.total.errors.overrun
cluster.infiniband.port.total.errors.symbol
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
"
```

An example that changes the metric list to only include the CPU metrics for `r1i1n0` is, as follows:

```
# pmstore -h r1lead -i 'r1i1n0' cluster.control.metrics 'cluster.kernel.percpu.cpu.user cluster.kernel.percpu.cpu.nic
cluster.kernel.percpu.cpu.sys cluster.kernel.percpu.cpu.idle cluster.kernel.percpu.cpu.intr cluster.kernel.percpu.cpu
```

Monitoring SDR Metrics

The sensor data repository (SDR) metrics are available through Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The SDR provides temperature, voltage, and fan speed information for all service nodes, rack leader controllers (RLCs), compute nodes, and CMCs. This information is collected from service and compute nodes through their BMC interface, so it is out-of-band and does not impact the performance of the node.

The following metrics are available through the PMCD:

```
admin:~ # pminfo -h r1lead sensor
sensor.value.fan
sensor.value.voltage
sensor.value.temperature
```

Each sensor will have a separate instance within the domain, with the instance of the form:

```
<nodeName>:<nodeType>:<metricName>

nodeName ::= SMC for SGI ICE X node names (rXlead, rXiYc, rXiYnZ)
nodeType ::= "service", "cmc", "blade", "leader"
```

For example, to view voltages for the RLC, perform the following

```
admin:~ # pminfo -h r1lead -f sensor.value.voltage | grep -E '(^$|^sensor|r1lead)'

sensor.value.voltage
    inst [0 or "r1lead:leader:CPU1_Vcore"] value 1.3
    inst [1 or "r1lead:leader:CPU2_Vcore"] value 1.3
    inst [2 or "r1lead:leader:3.3V"] value 3.26
    inst [3 or "r1lead:leader:5V"] value 4.9
    inst [4 or "r1lead:leader:12V"] value 11.71
    inst [5 or "r1lead:leader:-12V"] value -12.3
    inst [6 or "r1lead:leader:1.5V"] value 1.47
    inst [7 or "r1lead:leader:5VSB"] value 4.9
    inst [8 or "r1lead:leader:VBAT"] value 3.31
```

For additional examples on how to retrieve values using `pmval`(1) and for using this data in trend analysis using `pmie`(1), see the appropriate man page and the *Performance Co-Pilot Linux User's and Administrator's Guide.*

Turning Off the `temperature.pmie` Feature

Currently, in `temperature.pmie` there are values that will `"Monitor:  shut down components if temp too high"` . This feature is enabled by default as a safety mechanism. The procedure below describes how to turn it off.

Procedure 6-9 Turning Off the `temperature.pmie` Feature

To turn off the `temperature.pmie` feature, perform the following steps:

1. Edit the `/var/lib/pcp/config/pmie/control` file to comment out or remove the line that calls `/opt/sgi/lib/temperature.pmie`. For example,

```
#LOCALHOSTNAME n PCP_LOG_DIR/pmie/LOCALHOSTNAME/temperaturepmie.log -c /opt/sgi/lib/temperature.pmie
```

2. Run the **`/etc/init.d/pmie restart`** command. If you just want to adjust `temperature.pmie` values, see "Adjusting `temperature.pmie` Values" on page 284.

This has to be done on the system admin controller (SAC) and rack leader controller (RLC). In that case, it is recommended that you turn it off on the RLC images too.

Adjusting `temperature.pmie` Values

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

Procedure 6-10 Adjusting `temperature.pmie` Values

You can adjust the warning or shutdown temperature values manually on the system admin controller (SAC) and on each one of the rack leader controllers (RLCs). If you adjust the values on the RLC, adjust the values on the RLC images, too. The settings will be preserved between reboots. To change the values, perform the following steps:

1. Edit the `/opt/sgi/lib/temperature.pmie` file:

```
admin_warning_temperature = 68; // degree Celsius
admin_shutdown_temperature = 73; // degree Celsius
leader_warning_temperature = 68; // degree Celsius
leader_shutdown_temperature = 73; // degree Celsius
service_warning_temperature = 68; // degree Celsius
service_shutdown_temperature = 73; // degree Celsius
cmc_warning_temperature = 48; // degree Celsius
cmc_shutdown_temperature = 53; // degree Celsius
cn_warning_temperature = 68; // degree Celsius
cn_shutdown_temperature = 73; // degree Celsius
sensor_temperature = "sensor.value.temperature"; // degree Celsius
```

2. Perform the following command to verify that you updated the script correctly, as follows:

```
# pmie -C /opt/sgi/lib/temperature.pmie
```

If there are no errors, the `pmie -C` command returns with no message.

3. Run the **/etc/init.d/pmie restart** command or the **service pmie restart** command to restart the pmie service.

To turn off the temperature.pmie value, see "Turning Off the temperature.pmie Feature" on page 283.

Cluster Performance Monitor

You can use the Cluster Performance Monitor to monitor your SGI ICE X system. Log into the system admin controller (SAC) using the ssh -X command. Execute the pmice command and the **pmice - Cluster Performance Monitor** appears, as follows:

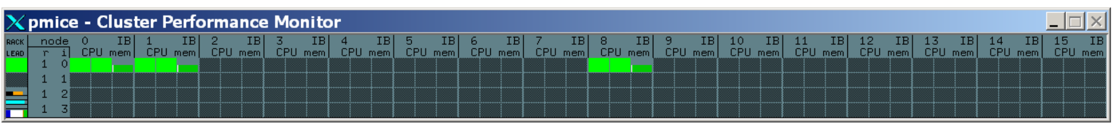

Figure 6-6 pmice- Cluster Performance Monitor

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

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

options:
  -K list  Show these CPUs. Comma-separated list
  -N list  Show these nodes. Comma-separated list
  -R list  Show these racks. Comma-separated list
  -V       Verbose/diagnostic output

pmgadgets(1) options:

  -C                      check configuration file and exit
  -h host                 metrics source is PMCD on host
  -n pmnsfile             use an alternative PMNS
  -t interval             sample interval [default 2.0 seconds]
  -z                      set reporting timezone to local time of metrics source
  -Z timezone             set reporting timezone
```

```
-zoom factor           make the gadgets bigger by a factor of 1, 2, 3 or 4
-infofont fontname     use fontname for text in info dialogs
-defaultfont fontname  use fontname for label gadgets

-display display-string
-geometry geometry-string
-name name-string
-title title-string
-xrm resource
```

Troubleshooting

This section describes some troubleshooting tools and covers these topics:

- "dbdump Command" on page 286
- "smc-info-gather Command" on page 288
- "cminfo Command" on page 289

dbdump Command

You can run the dbdump script to see an inventory of the SGI ICE X 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 (SAC).
- Use the --leader argument to dump all rack leader controllers (RLCs).
- Use the --rack argument to dump a specific rack.
- Use the dbdump command without any argument to dump the entire SGI ICE X system.

EXAMPLES

Example 6-1 dbdump Command Examples

To dump the entire database, perform the following:

```
admin:~ # dbdump
0 is { cluster=oscar ifname=service0-bmc dev=bmc0 ip=172.24.0.3 net=head-bmc node=service0
  nodetype=oscar_service mac=00:30:48:8e:
1 is { cluster=oscar ifname=service0 dev=eth0 ip=172.23.0.3 net=head node=service0
  nodetype=oscar_service mac=00:30:48:33:53:2e }
2 is { cluster=oscar ifname=service0-ib0 dev=ib0 ip=10.148.0.2 net=ib-0 node=service0
  nodetype=oscar_service }
3 is { cluster=oscar ifname=service0-ib1 dev=ib1 ip=10.149.0.2 net=ib-1 node=service0
  nodetype=oscar_service }
4 is { cluster=oscar dev=eth0 ip=128.162.244.86 net=public node=oscar_server
  nodetype=oscar_server mac=00:30:48:34:2B:E0 }
...
```

Note: Some of the sample output in this section has been modified to fit the format of this manual.

To dump just the RLC, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --leader
0 is { cluster=rack1 ifname=r1lead-bmc dev=bmc0 ip=172.24.0.2 net=head-bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:8a:a4:c2 }
1 is { cluster=rack1 ifname=lead-bmc dev=eth0 ip=192.168.160.1 net=bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
2 is { cluster=rack1 ifname=lead-eth dev=eth0 ip=192.168.159.1 net=gbe node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
3 is { cluster=rack1 ifname=r1lead dev=eth0 ip=172.23.0.2 net=head node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
4 is { cluster=rack1 ifname=r1lead-ib0 dev=ib0 ip=10.148.0.1 net=ib-0 node=r1lead
  nodetype=oscar_leader }
5 is { cluster=rack1 ifname=r1lead-ib1 dev=ib1 ip=10.149.0.1 net=ib-1 node=r1lead
  nodetype=oscar_leader }
```

To dump just one rack, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --rack 1
0 is { cluster=rack1 ifname=i0n0-bmc dev=bmc0 ip=192.168.160.10 net=bmc node=r1i0n0
  nodetype=oscar_clients mac=00:30:48:7a:a7:96 }
1 is { cluster=rack1 ifname=i0n0-eth dev=eth0 ip=192.168.159.10 net=gbe node=r1i0n0
  nodetype=oscar_clients mac=00:30:48:7a:a7:94 }
```

```
2 is { cluster=rack1 ifname=r1i0n0-ib0 dev=ib0 ip=10.148.0.3 net=ib-0 node=r1i0n0
  nodetype=oscar_clients }
3 is { cluster=rack1 ifname=r1i0n0-ib1 dev=ib1 ip=10.149.0.3 net=ib-1 node=r1i0n0
  nodetype=oscar_clients }
4 is { cluster=rack1 ifname=i0n1-bmc dev=bmc0 ip=192.168.160.11 net=bmc node=r1i0n1
  nodetype=oscar_clients mac=00:30:48:7a:a7:86 slot=1 }
5 is { cluster=rack1 ifname=i0n1-eth dev=eth0 ip=192.168.159.11 net=gbe node=r1i0n1
  nodetype=oscar_clients mac=00:30:48:7a:a7:84 slot=1 }
6 is { cluster=rack1 ifname=r1i0n1-ib0 dev=ib0 ip=10.148.0.4 net=ib-0 node=r1i0n1
  nodetype=oscar_clients slot=1 }
7 is { cluster=rack1 ifname=r1i0n1-ib1 dev=ib1 ip=10.149.0.4 net=ib-1 node=r1i0n1
  nodetype=oscar_clients slot=1 }
8 is { cluster=rack1 ifname=i0n10-bmc dev=bmc0 ip=192.168.160.20 net=bmc node=r1i0n10
  nodetype=oscar_clients slot=10 }
9 is { cluster=rack1 ifname=i0n10-eth dev=eth0 ip=192.168.159.20 net=gbe node=r1i0n10
  nodetype=oscar_clients slot=10 }
10 is { cluster=rack1 ifname=r1i0n10-ib0 dev=ib0 ip=10.148.0.13 net=ib-0 node=r1i0n10
  nodetype=oscar_clients slot=10 }
...
```

`smc-info-gather` Command

The `smc-info-gather` command enables to collect vital system data especially when troubleshooting problems. The `smc-info-gather` command collects the information about the following:

- Digital media `dminfo` files, syslogs, Dynamic Host Configuration Protocol (DHCP), network file system (NFS)
- MySQL cluster database dump
- Network service configuration files, for example, C3, Ganglia, DHCP, domain name service (DNS) configuration files
- A list of installed system images
- Log files in `/var/log/messages`
- Chassis management control (CMC) slot table for each rack
- basic input-output system (BIOS), Baseboard Management Controller (BMC), CMC and InfiniBand fabric software versions from all SGI ICE X nodes

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

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

cminfo Command

The `cminfo` command is used internally by many of the SMC for SGI ICE X scripts that are used to discover, configure, and manage an SGI ICE X 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 (RLC), 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 6-2 `cminfo` Command Examples

To see the RLC's BMC IP address, perform the following:

```
r1lead:~ # cminfo --bmc_base_ip
192.168.160.0
```

To see the RLC's DNS domain, perform the following:

```
r1lead:~ # cminfo --dns_domain
ice.domain_name.mycompany.com
```

To see the BMC NIC, perform the following:

```
r1lead:~ #  cminfo --bmc_nic
eth0
```

To see the IP address of the ib1 InfiniBand fabric, perform the following:

```
r1lead:~ # cminfo --ib_1_base_ip
10.149.0.0
```

kdump Utility

The `kdump` utility is a `kexec`-based crash dumping mechanism for the Linux operating system. You can downlonad `debuginfo` kernel RPMs for use with crash and any kernel dumps at the following location:http://support.novell.com/linux/psdb/byproduct.html.

To get a traceback or system dump, perform the following from the system console:

```
console r1i0n0
^e c l 1 8
^e c l 1 t      #traceback
^e c l 1 c      #dump
```

Note: This example shows the letter "`c`", a lowercase L "`l`", and the number one "`1`" in all three lines.

On the system admin controller (SAC), go to `/net/r1lead/var/log/consoles` for the traceback and `/net/r1lead/var/log/dumps/r1i0n0` for the system dump.

You can dump a compute node, the rack leader controller (RLC) (`r1lead`), or a service node, such as, `service0`.

System Firmware

Note: Your SGI ICE X system comes preinstalled with the appropriate firmware. See your SGI field support person for any BMC, BIOS, and CMC firmware updates.

The SGI ICE X system firmware software consists of the following components:

`sgi-ice-blade-bmc-1.43.5-1.x86_64.rpm`

Blade BMC firmware and update tool

`sgi-ice-blade-bios-2007.08.10-1.x86_64.rpm`

Blade BIOS image and update tool

`sgi-ice-cmc-0.0.11-2.x86_64.rpm`

CMC firmware and update tool

BIOS Version Interrogation

To identify the BIOS you need both the version and the release date. You can get these using the `dmidecode` command. Log onto the node on which you want to interrogate BIOS level and perform the following:

```
# dmidecode -s bios-version; dmidecode -s bios-release-date
```

BMC Revision Interrogation

The BMC firmware revision can be retrieved using the `ipmiwrapper`. For example, from the system admin controller (SAC), the following command gets the BMC firmware revision for `r1i0n0`:

```
# ipmiwrapper r1i0n0 bmc info | grep ’Firmware Revision’
```

CMC Version Interrogation

The CMC firmware version can can be retrieved using the `version` command to the CMC. For example, if you are logged onto the `r1lead` rack leader controller (RLC), the following command gets the CMC firmware version:

```
# ssh root@r1i0-cmc version
```

InfiniBand Version Interrogation

The `ibstat` command retrieves information for the InfiniBand links including the firmware version. The following command gets the InfiniBand firmware version:

```
# ibstat | grep Firmware
```

Getting Firmware Information for All System Nodes

The `firmware_revs` script on the system admin controller (SAC) collects the firmware information for all nodes in the SGI ICE X system, as follows:

```
admin:~ # firmware_revs
BIOS versions:
--------------
admin: 6.00
r1lead: 6.00
service0: 6.00
r1i0n0: 6.00
r1i0n1: 6.00
r1i0n8: 6.00
r1i1n0: 6.00
r1i1n1: 6.00
r1i1n8: 6.00


BIOS release dates:
-------------------
admin: 05/10/2007
r1lead: 05/10/2007
service0: 05/10/2007
r1i0n0: 05/29/2007
r1i0n1: 05/29/2007
```

```
r1i0n8: 05/29/2007
r1i1n0: 05/29/2007
r1i1n1: 05/29/2007
r1i1n8: 05/29/2007


BMC versions:
-------------
admin: 1.31
r1lead: 1.31
service0: 1.31
r1i0n0: 1.29
r1i0n1: 1.29
r1i0n8: 1.29
r1i1n0: 1.29
r1i1n1: 1.29
r1i1n8: 1.29


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


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

Troubleshooting

This chapter provides answers to some common problems users encounter when installing an SGI ICE X system and includes diagnosis and troubleshooting information. It covers the following topics:

Initial Installation Settings

This section describes some values you should set or verify when configuring the system for the first time before any system nodes are discovered:

- Discover management switches first. It is very important for systems with top level management switches that these switches be discovered before any other nodes. As other nodes are discovered, the already-discovered management switches are configured.
- When the system is an SGI ICE X system or a hybrid systems with SGI ICE X components and previous SGI ICE hardware components.
  - Use the `configure-cluster` GUI to enable the switch management network (see "Configuring a Switch Management Network" on page 73).
  - The `cmcdetectd` daemon sets switch management network value when it detects SGI ICE X components. Note that this only works when the top level switches have never before been configured for SGI ICE X. It is a good practice to set this value to yes using the he `configure-cluster` GUI to enable the switch management network.
  - If there are SGI ICE 8400 components in the system, be sure to add options to the `discover` command that disables the switch management network for any components on the SGI ICE 8400 side of the system.
- Use the `configure-cluster` GUI **Configure Default Max Rack IRU Setting** option to set the default number of blade enclosure pairs, formerly called individual rack units (IRUs), supported by a rack leader controller (RLC), see "Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting" on page 76. Set this value to the number of CMCs that will be served by each RLC. The default is 8.
- Use the `configure-cluster` GUI **Configure Redundant Management Network** option to turn on the redundant management network (RMN) system. If this system has switch stacks and each node has two Ethernet connections up to the stack, and both CMC-0 and CMC-1 ports on the CMC are connected, it is an RMN system.
- Use the `configure-cluster` GUI **Configure MySQL Replication (optional)** to enable MySQL Replication (see "Configuring MySQL Database Server Replication" on page 74), particularly important for large systems.

System Discovery Overview

This section describes software related to system discovery.

`configure-cluster` Command

The `configure-cluster` (see "Running the Cluster Configuration Tool" on page 61) GUI configures certain global variables, as follows:

- The redundant management network
- The switch management network
- The default max rack IRU setting
- The default `blademond` rescan interval

Warning: These settings in `configure-cluster` represent the global defaults. They are adjustable per-node by parameters to the `discover` command (see "`discover` Command" on page 84) and the `cadmin` command (see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 195). This is urgently important, for example, for mixed SGI ICE X and SGI ICE 8400 systems where some pieces of the system have legacy CMCs that manage VLANs while other parts have switches that represent the switch management network (VLANs/trunking controlled by the Ethernet switch)

`cmcdetected` Daemon

The `cmcdetectd` daemon runs on the system admin controller (SAC). When it sees a chassis management controller (CMC) asking for an IP address, it looks at the client ID of the request. That client ID contains the rack number and slot number. The `cmcdetectd` daemon provides this information to the `switchConfig` application programming interface (API) and configures the top level switch fabric (see "`cmcdetectd` Daemon" on page 91).

`discover` Command

The `discover` command is used to discover the various non-compute nodes and switches including:

- Top level management switches
- External InfiniBand switches
- Rack leader controllers (RLCs)
- Service nodes

Using options to the `discover` command, you can control the bonding mode (active-backup vs 802.3ad link aggregation), enable the switch management network (for SGI ICE X components of the system), Redundant Management Network, and many other things.

Note: Always discover management switches first. The `discover` command works with the `switchconfig` command to set up switches appropriately for RLCs, service nodes, and so on.

The `discover` command facilitates the first installation of a node (see "`discover` Command" on page 84). You can always re-install already-discovered nodes with the `cinstallman` command (see "`cinstallman` Command" on page 154).

`blademond` Daemon

Once the RLC is operating, the CMCs get their IP addresses. The `blademond` daemon creates the initial `dhcpd.conf` file (`ice.conf`) when it is started for the first time. It looks for CMCs, polls them to figure out where the compute nodes are, and configures the system for these nodes. For new CMCs/IRUs, `blademond` powers them up for the first time.

- The `blademond` daemon notices when blades are removed, swapped, added and updates the system as needed.
- You can configure how often blademond checks/polls the CMCs (see `configure-cluster` **Configure blademond rescan interval** and `cadmin --set-blademond-rescan`.

- You can decide not to run `blademond` as a daemon (use `chkconfig` to turn it off on the Rack Leader Controller (RLC) and RLC image). In that case, you can run it in `--scan-once` mode by hand or via `cron`.

For more information, see "Configuring the `blademond` Rescan Interval" on page 77.

Compute Nodes Are Taking Too Long To Boot

When compute nodes are taking a long time to boot, perform the following:

- See "Verify the Bonding Mode on the Rack Leader Controller (RLC)" on page 300 to verify that the compute nodes have the proper bonding setup.
- Verify the rack leader controller (RLC) has a MegaRAID controller. 144 nodes will not boot well with 106x controllers, for example. You can verify this with `lspci` command.

  To verify the MeagaRAID battery is working and charged, perform the following:

  ```
  # /opt/MegaRAID/MegaCli/MegaCli64 -ShowSummary -a0
  ```

  You should see '`Status :  Healthy' under 'BBU' (BBU = Battery Backup Unit`).

  Note: If this is the first time the node has booted up, it takes several hours for the BBU to be charged.

- Verify cache is set to write-back, as follows:

  ```
  # /opt/MegaRAID/MegaCli/MegaCli64 -LDGetProp -Cache -LALL -a0
  ```

  Note: Never force write-back on if bad BBU (`-CachedBadBBU`) as data loss happens with an orderly shutdown that includes a power off.

  When you see the output: `Cache Policy:WriteBack`, write-back is enabled.

  To enable the write-back policy, perform the following:

  ```
  # /opt/MegaRAID/MegaCli/MegaCli64 -LDSetProp -NoCachedBadBBU -LALL -a0
  ```

Verify the Bonding Mode on the Rack Leader Controller (RLC)

The redundant management network (RMN) is the default configuration for SGI ICE X systems. To verify the bonding mode, perform the following from the RLC:

```
r1lead:~ # cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.5.0 (November 4, 2008)

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer2+3 (2)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

802.3ad info
LACP rate: slow
Aggregator selection policy (ad_select): stable
Active Aggregator Info:
        Aggregator ID: 1
        Number of ports: 2
        Actor Key: 17
        Partner Key: 4
        Partner Mac Address: b4:0e:dc:37:4f:a7

Slave Interface: eth0
MII Status: up
Link Failure Count: 1
Permanent HW addr: 00:25:90:38:e5:22
Aggregator ID: 1

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:25:90:38:e5:23
Aggregator ID: 1
```

If you see Bonding Mode: IEEE 802.3ad Dynamic link aggregation, RMN is on.

If you see `Bonding Mode: fault-tolerance` (active-backup), it means that the bonding mode and potentially redundant management networking is disabled.

Use the `configure-cluster` GUI **Configure Redundant Management Network** option to turn on the redundant management network (RMN) system for nodes being discovered going forward.

Set the redundant management networking mode on, as follows:

```
# cadmin --enable-redundant-mgmt-network --node r1lead
```

Set the bonding mode per node, as follows:

```
# cadmin --set-mgmt-bonding --node r1lead 802.3ad
```

You need to reboot the system.

The `/proc/net/bonding/bond0` file, should show the bonding mode with link aggregation configured, as follows:

```
Bonding Mode: IEEE 802.3ad Dynamic link aggregation
```

The number of ports should be the following:

```
Number of ports: 2
```

`2` is the correct value for an RMN configuration. If the number is 1, it mean the trunk has not formed. The likely causes for this are, as follows:

- The Ethernet cable is not connected to top level switch. From the RLC, use the `/sbin/ethtool` on `eth0` and `eth1` to verify the link is present, as follows:

```
r1lead:~ # /sbin/ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
```

```
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: umbg
Wake-on: g
Current message level: 0x00000003 (3)
Link detected: yes
```

- The Ethernet cable is connected, but linking is wrong. When the `/sbin/ethtool` command output shows the link speed as 100mbit due to a bad cable the trunk leg is rejected.
- The top level Ethernet switch misconfigured: Perhaps the `switchconfig` tool did not get this port configured properly. You can either log in to the switch to try to diagnose, or try the following procedure:

  1. Find the MAC address of the `r1lead` bond interface, as follows:

     ```
     r1lead:~ # ifconfig bond0
     bond0     Link encap:Ethernet  HWaddr 00:25:90:38:E5:22
               inet addr:172.23.0.7  Bcast:172.23.255.255  Mask:255.255.0.0
               inet6 addr: fe80::225:90ff:fe38:e522/64 Scope:Link
               UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
               RX packets:286749167 errors:0 dropped:0 overruns:0 frame:0
               TX packets:328574062 errors:0 dropped:0 overruns:0 carrier:0
               collisions:0 txqueuelen:0
               RX bytes:38868281915 (37067.6 Mb)  TX bytes:153036792319 (145947.2
     ```

  2. From the system admin controller (SAC), run the `switchconfig list --switches mgmtsw0` command to list the MAC addresses trunks from the switches, as follows:

     ```
     sys-admin:~ # switchconfig list --switches mgmtsw0
     Current MAC/port configuration:

     Switch Identifier: mgmtsw0   IP Address: 172.23.0.6

           MAC          Port  Trunk  default-VLAN  allowed-VLANs
     -------------------------------------------------------------
     00-25-90-3F-16-C4   1/6                    1          1(u)
     00-30-48-F7-84-65  1/48                    1          1(u)
     00-25-90-38-E5-22   1/5      1             1  1(u),  101(t)
     00-25-90-38-E5-23   1/5      1             1  1(u),  101(t)
     ```

```
00-25-90-38-E5-22   1/5      1           101  1(u),  101(t)
00-25-90-38-85-BC   1/7      2             1           1(u)
00-25-90-38-85-BD   1/7      2             1           1(u)
                            ...
```

If the RLC `r1lead` bond interface `MAC` address shows up in the `Port` column and not the `Trunk` column, the switch is not configured correctly.

3. To properly configure the switch, from the admim node, perform a command similar to the following:

```
sys-admin:~ # switchconfig set -s mgmtsw0 -v num=1 -v num=101,tag=tagged -b lacp -d 1 -m 00:25:90:38:E5:
```

This replaces 101 with the proper VLAN number. 101 for rack group 1, 102 for rack group 2, and so on.

4. `ssh` onto the `r1lead` and verify that the RLC shows `Number of ports: 2`.

Note: This procedure only applies to RLCs.

`cimage --push-rack` Pushes Too Many (or Too Few) Expansions

When you perform `cimage --push-rack` (or when `blademond` calls `discovery-rack`), it creates read / write expansions for each compute node.

Use the `configure-cluster` GUI **Configure Default Max Rack IRU Setting** option to set the default number of blade enclosure pairs, formerly called individual rack units (IRUs), supported by a rack leader controller (RLC), see "Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting" on page 76. Set this value to the number of CMCs that will be served by each RLC. The default is 8. When you change it, it only impacts node discoveries in the future.

You can change the setting per-node with the `cadmin` command, as follows:

```
sys-admin:~ # cadmin --set-max-rack-irus --node r1lead 8
```

Cannot ping the CMCs from the Rack Leader Controller (RLC)

If this is an RLC with a brand new, never-before-discovered top level switch (or set of switches), the `cmcdetectd` daemon will see CMCs asking for IP addresses on the HEAD network. It configures the top level switch(es) so that the CMCs are on the appropriate rack VLAN. Make sure `cmcdetectd` is running, restart if needed.

You can diagnose this some by running the `tcpdump` command looking for DHCP requests. The requests should be seen on the RLC and not the system admin controller (SAC). For example:

```
r1lead:~ # /usr/sbin/tcpdump -i bond0 -s600 -nn -vv -e -t -l -p broadcast and src port 68 and dst port 6
tcpdump: listening on bond0, link-type EN10MB (Ethernet), capture size 600 bytes
00:25:90:3f:16:c4 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 590: (tos 0x0, ttl 64, id 0, offset 0, flags [
[udp sum ok] BOOTP/DHCP, Request from 00:25:90:3f:16:c4, length 548, xid 0x8b8d332a, Flags [none] (0x0000)
          Client-IP 172.24.0.2
          Client-Ethernet-Address 00:25:90:3f:16:c4
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Request
                            ...
```

If the switch was previously discovered but you are reinstalling the system or discovering a new root slot, `cmcdetectd` will not detect any CMC DHCP requests on HEAD. In this case, you need to be sure to run `configure-cluster` and set **Configure Switch Management Network** to yes. Note that changing `configure-cluster` only takes effect for nodes discovered in the future. If you have an existing RLC already discovered, you will need to run a command like the following:

```
# cadmin --enable-switch-mgmt-network --node r1lead
```

After rebooting the RLC, make sure that the `ifconfig` command shows `vlan101` as an interface and not `vlan1` or `vlan2` interfaces, as follows:

```
r1lead:~ # ifconfig
                              ...
vlan101   Link encap:Ethernet  HWaddr 00:25:90:38:E5:22
          inet addr:192.168.160.1  Bcast:192.168.160.255  Mask:255.255.255.0
          inet6 addr: fe80::225:90ff:fe38:e522/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:290550897 errors:0 dropped:0 overruns:0 frame:0
          TX packets:268387414 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:0
RX bytes:30869741447 (29439.6 Mb)  TX bytes:120262245830 (114691.0 Mb)
```

Confirm `dhcpd` is running on the RLC. If `dhcpd` is not running, CMCs will not get their IP addresses. Check for errors starting `dhcpd`. If `blademond` failed to create the `ice.conf dhcpd` configuration file (`/etc/dhcpd.conf.d`) , see "How to Make the `blademond` Daemon Start Over from Scratch" on page 306.

Verify proper CMC configuration. The CMC is configured for its rack number and slot number. If they are not configured correctly, multiple CMCs can be configured the same way resulting in problems. This can also result in the `ice.conf dhcp` configuration file being corrupted. You may need a USB serial cable to fix the CMCs if this is the case.

One troubleshooting approach is to run `tcpdump` on the RLC, as follows:

```
usr/sbin/tcpdump -i bond0 -s600 -nn -vv -e -t -l -p broadcast and src port 68 and dst port 67
```

Watch the DHCP requests over several minutes. If you see the same Client Identifier being requested by more than one MAC address, you are in a situation where the CMCs are not configured correctly.

Verify that the RLC is properly configured in the switch (see "Configure Switches for a Rack Leader Controller (RLC)" on page 311).

See "`r1lead` Configured with `vlan1`/`vlan2` and Not `vlan101`" on page 306.

Confirm the wiring rules. See "Switch Wiring Rules" on page 313.

If you moved some CMCs from one RLC number to another and you already adjusted the rack and slot number in the CMC, The switch likely does not know about the changes. The CMCs are likely "lost" in the wrong VLAN, potentially a VLAN that is no longer in use. For example, if you had the CMCs served by the `r3lead` RLC but decided to decommission `r3lead` and move the CMCs to `r1lead` instead this situation could arise. In this case, the switch must be reconfigured. Use the `switchconfig` command to configure the ports connected to those CMCs for head. The system admin controller (SAC) `cmcdetectd` daemon will move them to the correct ultimate location.

You need to know the MACs of the CMC embeded Linux for this, so perhaps record this when you change the slot/rack number in the CMC. **Hint:** `dbdump` may still have the information depending on how you removed the RLC.

An example command is, as follows:

```
# switchconfig set -v num=1 -b manual -d 1 -m 08:00:69:16:51:49 --switches mgmtsw0
```

If you have more than one management switch, then list them in a comma-separated-list for `--switches`.

In a non-redundant-management configuration (switches not stacked), if the `dhcpd` daemon shows DHCP requests from the CMC but the CMC remains unpingable, it could be that both CMC-0 and CMC-1 are connected and linked. This breaks the wiring rules. When we are **not** wired for redundant management networking, only CMC-0 should be connected.

When **not** wired for redundant management networking (when switches are not stacked), do not connect CMC-1.

`r1lead` Configured with `vlan1`/`vlan2` and Not `vlan101`

See the VLAN and switch management network settings explanation in "Cannot `ping` the CMCs from the Rack Leader Controller (RLC)" on page 304.

How to Make the `blademond` Daemon Start Over from Scratch

From the rack leader controller (RLC), perform the following steps:

1.
```
r1lead:~ # service blademond stop
```

2.
```
rm ice.conf dhcpd.conf
```

   Remove `etc/dhcpd.conf.d/ice.conf` or `/etc/dhcp/dhcpd.conf.d/ice.conf`

3.
```
r1lead:~ # rm /var/opt/sgi/lib/blademond/slot_map
```

4.
```
r1lead:~ # service blademond start
```

Interesting Log Files

All of the logs in /var/log directory. In addition to the messages log file and in some cases dhcpd file on the rack leader controller (RLC), here are some interesting /var/log directory log files:

- /var/log/discover-rack

  On the system admin controller (SAC), the discover-rack call is facilitated by blademond when new nodes are found. This log will often show problems with discovering nodes.

- var/log/blademond

  On the RLCs, this shows the blademond daemon actions. This includes showing when blade changes are found and it also shows its call to discover-rack, and so on. If there are CMC communication issues, they will often be noticed in this log.

- /var/log/cmcdetectd.log

  On the SAC, cmcdetectd logs its actions as it configures the switches for CMCs in the system. Watch for progress or errors here.

- /var/log/switchconfig.log

  On the SAC, there is a switchconfig command line tool. This tool is largely used by the discover command as nodes are discovered. Its actions are logged in to this log file. If RLC VLANs are not functioning properly, check the switchconfig log file.

CMC slot_map / blademond Debugging Hints

This section describes what to do when the blademond daemon cannot find a system blade, as follows:

- Can you ping the CMCs? See "Cannot ping the CMCs from the Rack Leader Controller (RLC)" on page 304.

- If the CMCs are pingable, verify that they have a valid slot map. If the slot map returned by the CMC is missing entries, then blademond cannot function properly. It operates on information passed to it by the CMC. Some commands to run from the rack leader controller (RLC) are, as follows:

  –

  ```
  r1lead:~ # /opt/sgi/lib/dump-cmc-slot-tables
  ```

  This command attempts to dump the slot map from each CMC to your screen.

  –

  ```
  r1lead:~ # echo STATUS | netcat r1i0c 4502
  ```

  This command can query an individual slot map.

  Note: In some software distributions, netcat is nc.

If the CMCs are pingable and the CMCs have valid slot maps, then you can focus on how blademond is functioning.

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

```
# kill -USR1 pid
```

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

The blademond daemon maintains the slot map at /var/opt/sgi/lib/blademond/slot_map on the RLCs. This appears as /var/opt/sgi/lib/blademond/slot_map.*rack_number* on the system admin controller (SAC).

For a blademond --help statement, ssh onto the r1lead RLC, as follows:

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

Discover CMCs and blades managed by CMCs.
```

```
Note: This daemon normally takes no arguments.
  --help      Print this usage and exit.
  --debug     Enable debug mode (also can be enabled by setting CM_DEBUG)
  --fakecmc   Development only: Discover fake CMCs instead of real ones
  --scan-once Initialize, scan for blades, set blades up. Do not daemonize.
              Do not keep looping - do one pass and exit.
```

ssh Commands to Compute Nodes: ssh Key Failures / Known Hosts

For information, see "Resolving CMC Slot Map Ordering Issues" on page 309.

Compute Node Hosts Seem to Actually be BMCs

For information, see "Resolving CMC Slot Map Ordering Issues" on page 309.

Resolving CMC Slot Map Ordering Issues

The CMC maintains a cache file that records which MACs are BMC-MACs and which are host-MACs. It uses this information, combined with switch port location information in the embedded Broadcom switch, to generate the slot map used by the `blademond` daemon.

In certain situations, such as, a CMC reflash, may remove the cache file but leave CMC power active. In this situation, the CMC does not know which MACs on a given embedded switch port are host and which are BMC and gets the order randomly incorrect. It then caches the incorrect order. To fix this for each CMC, turn the power off with `pfctl`, zero out the MAC cache file, and reset each CMC. Then have `blademond` start over from scratch (see "How to Make the `blademond` Daemon Start Over from Scratch" on page 306). Perform the following steps:

1. `ssh` as root to the rack leader controller (RLC), as follows:

   ```
   sys-admin:~ # ssh r1lead
   Last login: Thu Jan 26 13:57:53 2012 from admin
   r1lead:~ #
   ```

2. Disable the `blademond` daemon, as follows:

```
r1lead:~ # service blademond off
```

3. Turn off IRU power for each CMC using the `cpower` command, as follows:

```
# PDSH_SSH_ARGS_APPEND="-F /root/.ssh/cmc_config" pdsh -g cmc pfctl off
```

4. Zero out the slot map cache file, as follows:

```
# PDSH_SSH_ARGS_APPEND="-F /root/.ssh/cmc_config" pdsh -g cmc cp /dev/null /work/net/broadcom_mac_addr_c
```

5. Reboot the CMC, as follows:

```
# PDSH_SSH_ARGS_APPEND="-F /root/.ssh/cmc_config" pdsh -g cmc reboot
```

6. Restart `blademond` from scratch, see "How to Make the `blademond` Daemon Start Over from Scratch" on page 306.

In `tmpfs` Mode, File Has Date in the Future Warnings

If you boot a compute node with `tmpfs`, part of the process transfers a root tarball using multicast. This tarball is then expanded. If you see hundreds of "`file X has a time in the future`" messages, it likely means your hardware clock is not set to system time properly (see "Ensuring Hardware Clock Has the Correct Time" on page 310).

Ensuring Hardware Clock Has the Correct Time

Some software distributions do not synchronize the system time to the hardware clock as expected. As a result, the hardware clock may not get synchronized with the system time as it should. At shut down, the system time is copied to the hardware clock, but sometimes this does not happen.

To set all the compute node hardware clocks up properly, perform the following:

- Make sure the system admin controller (SAC) and rack leader controller (RLC) have the correct time
- Make sure the SAC and RLCs are synchronized with `ntp`. A SAC can show a message like the following:

```
ntpd[20489]: synchronized to 128.162.244.1, stratum 2
```

- An RLC might show a message like the following:

```
20 Jan 22:54:14 ntpd[16831]: synchronized to 172.23.0.1, stratum 3
```

- Maker sure the compute nodes have the correct time. They use `ntp` broadcast packets but still will display this:

```
20 Jan 23:05:16 ntpd[4925]: synchronized to 192.168.159.1, stratum 4
```

  You can also use a command like the following and view the output:

```
sys-admin:~ # pdsh -g leader pdsh -g compute date
```

- Issue the following command to set the hardware clock to the system clock, as follows:

```
sys-admin:~ # pdsh -g leader pdsh -g hwclock --systohc
```

- You can run the `hwclock` without options to confirm the current hardware clock time, as follows:

```
sys-admin:~ # hwclock
Thu 26 Jan 2012 10:57:27 PM CST  -0.750431 seconds
```

Configure Switches for a Rack Leader Controller (RLC)

Normally, as you discover RLCs, `switchconfig` is called automatically and the switch ports associated with the RLC are configured in the special way needed for RLCs, as follows:

- Default VLAN 1
- Accept rack VLAN packets tagged (rack 1 `vlan` is `vlan101`)
- Link Aggregation is the bonding mode between the two ports associated with the RLC

If an RLC is moved in the switch or if `switchconfig` failed during discovery for some reason, you can run `switchconfig` by hand to configure the switch, as follows:

1. Certain switch wires rules must be followed in switch configuration, see "Switch Wiring Rules" on page 313.
2. Make sure all management switches are reachable from the system admin controller (SAC).

3. Find the MAC addresses associated with the RLC interfaces. You can do this by running the following command on the RLC in question:

```
r1lead:~ # cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.5.0 (November 4, 2008)

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer2+3 (2)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

802.3ad info
LACP rate: slow
Aggregator selection policy (ad_select): stable
Active Aggregator Info:
        Aggregator ID: 1
        Number of ports: 2
        Actor Key: 17
        Partner Key: 4
        Partner Mac Address: b4:0e:dc:37:4f:a7

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:25:90:38:e5:22
Aggregator ID: 1

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:25:90:38:e5:23
Aggregator ID: 1
```

Caution: Because bonded interfaces are in play, you cannot get both MAC addresses from using the `ifconfig` command . The `ifconfig` command will show the same MAC address for `eth0` and `eth1` if redundant management networking is enabled.

4. Determine which management switches are present, as follows:

```
r1lead:~ # cnodes -mgmtsw
mgmtsw0
```

5. When you have the list of management switches and the MAC addresses of the RLCs, run a command similar to the following:

```
# switchconfig set --vlan num=1 --vlan num=101,tag=tagged --bonding=802.3ad --default-vlan 1 /
--macs 00:e0:ed:0a:f2:0d,00:e0:ed:0a:f2:0e --switches mgmtsw0,mgmtsw
```

This replaces the MACs and management switches with the proper ones. It replaces the `101` with the VLAN for the rack, normally "100 + rack number" so rack 1 is 101, rack 2 102.

Switch Wiring Rules

This section is mainly of interest to SGI ICE X system configurations that have a redundant management network setup (stacked pairs of switches) or larger systems that have switch stacks cascaded from the top level switch.

When discovering cascaded switches, it is impossible to know the connected switch ports of all trunks in advance. So when discovering cascaded switches, you can only start with one cable for discovering, then add the second one later on.

When trunks are configured, it is often hard to find the MAC address of both legs of the trunk. This is because the trunked connection just uses one MAC for the connection. Therefore, you need to rely on rules that infer the second port's connection based on the first port.

Some simple wiring rules are, as follows:

- In a redundant management network (RMN) configuration, when connecting system admin controllers (SACs), rack leader controllers (RLCs), service nodes, and CMCs, you must always use the same port number for the same node in both switches in the stack. In other words:

- If you connect `r1lead eth0` to switch A, port 43, then you must connection `r1leadeth1` to switch B, port 43.
- Likewise, if you connect CMC `r1i0c` CMC-0 port to switch A, port 2, then `r1i0c` CMC-1 port must go to switch B port 2.

- When adding cascaded switch stacks, all switch stacks must cascade from the central switch stack. In other words, there is always only, at most, one switch hop.
- When discovering cascaded switches pairs in an RMN setup, observe the following:
  - If you are connecting switch stack 1, switch A, port 48 to switch stack 2, then you must connect the second trunked connection to stack 2, switch B, port 48.
  - Until the cascaded switch stack is discovered, you must leave one trunk leg unplugged temporarily.
  - The `discover` command will tell you when it is safe to plug in the second leg of the trunk. This avoids circuit loops.

System Admin Controller (SAC) `eth2` Link in the Bond is Down

A problem occasionally occurs, especially in SGI XE270 SACs, where the active-backup or 802.3ad bonded `bond0` interface contains an Ethernet `eth2` interface that is down/not linked. To verify this, perform the following:

- Check the Ethernet port of the add-in card and confirm that it is lit.
- Confirm that the add-in card connection to the CMC (ICE 8400) or management switches (SGI ICE X) is using port "0" with port "1" not connected (so not mis-wired).
- If you look at `/proc/net/bonding/bond0` file, you can confirm that `eth2` is the link that is down.
- Use the `/sbin/ethtool eth2` command and confirm that the `Link detected:` is no.
- Run the commands `ifconfig up eth3` and the run the `/sbin/ethtool eth3` command to determine if the link detected is `yes`.

In this scenario, it is likely that the `eth2`/`eth3` interfaces have been swapped. Another clue is that if `eth2` (look at `/proc/net/bonding/bond0` since the bond enforces the same MAC address for all bonded members) has a MAC address that is larger than the MAC address of `eth3` (as seen by `ifconfig eth3`).

To correct this situation, edit the `/etc/udev/rules.d/70-persistent-net.rules` file and swap the MACs associated with `eth2` and `eth3` in the file.

When you reboot the system, the SAC comes back up with `eth2` and `eth3` properly ordered.

No InfiniBand Interfaces on Rack Leader Controller (RLC), Service, or Compute Node Images

---

Note: This section only applies to systems running SLES 11 SP1.

If you find that an RLC, service node, or compute nodes seem to lack an expected InfiniBand `ib0` interface, this is likely caused by Open Fabrics Enterprise Distribution (OFED) packages that are too old.

In addition to the `ib0` network interface not being present, you may observe the following message:

```
Loading kernel module for a network device with CAP_SYS_MODULE (deprecated). Use CAP_NET_ADMIN and alias netdev-ib0 i
```

The minimum OFED versions to avoid this problem are, as follows:

- `ofed-kmp-default-1.5.2_2.6.32.46_0.3-0.9.13.1.x86_64.rpm`
- `ofed-1.5.2-0.9.13.1.x86_64.rpm`

SGI also suggests that you use a the following kernel level (or later): `kernel-default-2.6.32.54-0.3.1.3900.0.PTF.743209.x86_64.rpm`

Find the updated SLES 11 SP1 packages in your local updates mirror, as follows:

```
/data/mirrors/novell/sles/updates/SLES11-SP1-Updates/sle-11-x86_64/rpm/x86_64
```

Out of Memory Adjustment

This section describes sample set of out of memory OOM adjust scripts for `cron` and PBS `prologue` and `epilogue`.

Example A-1 `oom_adj.user.pl.txt`: OOM Adjustment Script

```
#!/usr/bin/perl
use strict;

use Sys::Hostname;
my $host = hostname();
my $DEBUG=0; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];

sub ResetOomAdj {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $CMD_LINE;
my $RETURN;
$AVOID_UIDS="root|100|nobody|ntp|USER|daemon|postfix|vtunesag";
   open (PS_CMD, "-|") || exec 'ps -e -o user,pid';
   while (<PS_CMD>) {
     chomp;
     ($_userid, $tpid) = split (/\s+/, $_);

       if ( $_userid !~ m/^${AVOID_UIDS}/ && $tpid =~ /^[0-9]/ && -e
"/proc/$tpid/oom_adj" ) {
          print "$CALL_SCPT $host: Found processes to set to zero
oom_adj...\n" if $DEBUG;
          $CMD_LINE="echo 0 > /proc/$tpid/oom_adj";
          $RETURN=`$CMD_LINE`;
       }
       elsif ( $tpid =~ /^[0-9]/ && -e "/proc/$tpid/oom_adj" ) {
          print "$CALL_SCPT $host: Found processes to set to protect
oom_adj...\n" if $DEBUG;
          $CMD_LINE="echo -17 > /proc/$tpid/oom_adj";
          $RETURN=`$CMD_LINE`;
       }
```

```
    }
    close PS_CMD;

}

&ResetOomAdj();
```

Example A-2 `cronentry`: Sample `cron` Entry for `oom_adj` Script

```
-*/2 *   * * *     /root/oom_adj.user.pl
```

Example A-3 `prologue`: Sample `prologue` Script

```
#!/bin/bash
##################################################################################
#
#  Version: 2.3.1 : Updated 8/12/09
#  Date: Oct 16, 2007
#  Author: Scott Shaw, sshaw@sgi.com
#
#  Script Name: PBS Pro Prologue Script
#  The purpose of the Prologue script is to terminate leftover user processes and
#  allocated IPCs resources. The prologue script consists of two scripts, the main
#  prologue script and a chk_node.pl script.  To minimize accessing each node the
#  prologue script executes a parallel ssh shell across a set of nodes based on the
#  PBS_NODEFILE.  For large clusters over 64 nodes serial ssh access is slow so having
#  a flexible parallel ssh to help speed up the clean-up process of each node.  In
#  some cases, a PBS jobs can normally terminate but some MPI implementations do not
#  normally terminate the MPI processes due to crappy error code handling or
#  segmentation faults within the MPI application thus leaving behind user processes
#  still consuming system resources.
#
#  When the prologue script is launched by PBS MOM the ssh session is executed and will
#  execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
#  commands which are executed on each node based on the PBS_NODEFILE.
#
#  Execution of the prologue script is based on the root account.
#
#  This script needs to reside on each execution host/node
```

```
#  Location: /var/spool/PBS/mom_priv
#  File name: prologue
#  Permissions: 755
#  Owner: root
#  Group: root
#
#  ls output: ls -l /var/spool/PBS/mom_priv/prologue
#       -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/prologue
#
#  Modification of the prologalarm maybe necessay if the network access is slow to
#  each node.  30 seconds may not be enough time to check 256 nodes in a cluster.
#  prologalarm #  Defines the maximum number of seconds the prologue
#  and prologue may run before timing out. Default:
#  30. Integer. Example:
#  $prologalarm 30
#
##################################################################################


JOBID=$1
USERNAME=$2
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

spawn ()
{
        if [[ ‘jobs | grep  -v Done | wc -l‘ -ge $1 ]]; then
                wait
        fi
```

```
        shift
        $@  &
}

exec_cmd ()
{
   for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
   do
      spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
   done
   wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
   QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
   QSTAT=/opt/pbs/default/bin/qstat

else
  echo "Epilogue Error: The qstat command could not be detected, exiting..."
  exit 1
fi


prefix_flag=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $4}’ | awk -F. ’{print $1}’‘
queue=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $3}’‘

   echo "Start Prologue ${VERSION} ‘date‘ "

   if [ $( /bin/uname -m ) = "x86_64" ]; then
         echo "Prefix passed: ${prefix_flag}"
         echo "destination queue: ${queue}"

         case $prefix_flag in
            TB)
              # Enable turbo and do node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} TB"
              exec_cmd
```

```
            ;;
          BP)
            # Bypass the turbo setting and P/Elog cleanup
            echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
            ;;
          JT)
            # Enable turbo but do not run the node cleanup p/elog scripts
            CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} JT"
            exec_cmd
            ;;
          NT)
            # bypass turbo settings but run the node cleanup
            CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} NT"
            exec_cmd
            ;;
           *)
            # disable turbo and run the node cleanup scripts
            CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue}"
            exec_cmd
        esac
   else
      echo "The prologue script is intended to run on x86_64 nodes not ‘uname -m‘."
      echo "End Prologue ${VERSION} ‘date‘ "
      exit -1
   fi
   echo "End Prologue ${VERSION} ‘date‘ "


#Output the cluster details file
   if [ -f /var/spool/PBS/mom_priv/cluster_info.out ]; then
      cat /var/spool/PBS/mom_priv/cluster_info.out
   else
      echo "WARNING: The cluster info file does not exist. Contact hpc_support and report this warning."
   fi
```

Example A-4 `epilogue`: Sample `epilogue` Script

```
#!/bin/bash
##################################################################################
#
#  Version: 2.3.1 : Updated 8/12/09
#  Date: Oct 16, 2007
```

```
#  Author: Scott Shaw, sshaw@sgi.com
#
#  Script Name: PBS Pro Epilogue Script
#  The purpose of the epilogue script is to terminate leftover user processes and
#  allocated IPCs resources. The epilogue script consists of two scripts, the main
#  epilogue script and a chk_node.pl script.  To minimize accessing each node the
#  epilogue script executes a parallel ssh shell across a set of nodes based on the
#  PBS_NODEFILE.  For large clusters over 64 nodes serial ssh access is slow so having
#  a flexible parallel ssh to help speed up the clean-up process of each node.  In
#  some cases, a PBS jobs can normally terminate but some MPI implementations do not
#  normally terminate the MPI processes due to crappy error code handling or
#  segmentation faults within the MPI application thus leaving behind user processes
#  still consuming system resources.
#
#  When the epilogue script is launched by PBS MOM the ssh session is executed and will
#  execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
#  commands which are executed on each node based on the PBS_NODEFILE.
#
#  Execution of the epilouge script is based on the root account.
#
#  This script needs to reside on each execution host/node
#  Location: /var/spool/PBS/mom_priv
#  File name: epilogue
#  Permissions: 755
#  Owner: root
#  Group: root
#
#  ls output: ls -l /var/spool/PBS/mom_priv/epilogue
#       -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/epilogue
#
#  Modification of the prologalarm maybe necessay if the network access is slow to
#  each node.  30 seconds may not be enough time to check 256 nodes in a cluster.
#  prologalarm #  Defines the maximum number of seconds the prologue
#  and epilogue may run before timing out. Default:
#  30. Integer. Example:
#  $prologalarm 30
#
#################################################################################


JOBID=$1
```

```
USERNAME=$2
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

spawn ()
{
        if [[ ‘jobs | grep  -v Done | wc -l‘ -ge $1 ]]; then
                wait
        fi
        shift
        $@  &
}

exec_cmd ()
{
   for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
   do
      spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
   done
   wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
   QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
   QSTAT=/opt/pbs/default/bin/qstat
```

```
else
  echo "Epilogue Error: The qstat command could not be detected, exiting..."
  exit 1
fi



prefix_flag=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $4}’ | awk -F. ’{print $1}’‘
queue=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $3}’‘

   echo "Start Epilogue ${VERSION} ‘date‘ "
   if [ $( /bin/uname -m ) = "x86_64" ]; then
         echo "Prefix passed: ${prefix_flag}"
         echo "destination queue: ${queue}"

         case $prefix_flag in
            TB)
              # Enable turbo and do node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
              exec_cmd
              ;;
            BP)
              # Bypass the turbo setting and P/Elog cleanup
              echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
              ;;
            JT)
              # Enable turbo but do not run the node cleanup p/elog scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset JT"
              exec_cmd
              ;;
            NT)
              # bypass turbo settings but run the node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog noreset NT"
              exec_cmd
              ;;
             *)
              # disable turbo and run the node cleanup scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
              exec_cmd
```

```
          esac
else
   echo "The epilogue script is intended to run on x86_64 nodes not ‘uname -m‘."
   echo "End Epilogue ${VERSION} ‘date‘ "

   exit -1
fi
   echo "End Epilogue ${VERSION} ‘date‘ "
```

Example A-5 `chk_node.pl.txt`: Script `epilogue` and `prologue` Use.

```
#!/usr/bin/perl
#  Version: 2.3.1 : Updated 8/12/09
#  Orig Date: Oct 10, 2007
#  Author: Scott Shaw, sshaw@sgi.com
#
#  This perl script is called by PBS Pro prologue and epilogue scripts when
#  a user submits a job through PBS Pro.  The purpose of this script is to
#  sanitize a range of nodes identified by the $PBS_NODEFILE list by
#  terminating old user processes, old ipc allocations, temp files,
#  and to flush the system buffer cache.
#
#  Changes:
#   2/1/08 sshaw@sgi.com
#          - Added a subroutine to clean-up /tmp directory
#          - changed system() to exec since it was corrupting memory
#          - declared all vars to be local to subroutine, before it was loosely defined
#          - added strict checking of perl script
#   3/24/08 sshaw@sgi.com
#          - fixed debug conditional
#          - cleaned up the CleanUpProcesses procedure and added which processes
#            and user being terminated.
#          - Changed the killall to pkill due to userid > 8 chars
#   11/13/08 sshaw@sgi.com
#          - added a subroutine to clean-up /dev/shm since several users
#            use this location for temporary scratch space.
#   03/31/09 sshaw@sgi.com
#          - added subroutines to enable/disable Turbo mode on Intel series 5500 CPUs
#   04/22/09 sshaw@sgi.com
#          - added subroutines to speed step the core processor frequency to a lower freq
#   08/12/09 sshaw@sgi.com
#          - fixed minor issues with setting the frequency and fixed cpu freq to max speed
```

```
use strict;

use Sys::Hostname;
my $host = hostname();
my $DEBUG=1; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];
my $queue_destination=$ARGV[1];
my $prefix_option=$ARGV[2];
my $set_freq=0;

#####
# The following lines are added for Turbo/SMT mode starting with Intel 5500 series CPUs
my $rdmsr = "/var/spool/PBS/mom_priv/rdmsr";
my $wrmsr = "/var/spool/PBS/mom_priv/wrmsr";
my $msr = "0x199";
my $tbit = 1 <<  32;


# Several MPI implementations or MPI applications use IPC shared memory.  When
# a MPI application abnormally terminates it leaves behind allocated resources.
# this subroutine will remove any IPC resources allocated for the user’s job.
sub CleanUpIPC_Table {
my $tkey;
my $tshmid;
my $towner;
my $tperms;
my $tbytes;
my $tnattch;
my $tstatus;
my $CMD_LINE;
my $RETURN;

open(IPC_SHARMEM, "-|") || exec ’ipcs -m’;
   while () {
   chomp;
   ($tkey, $tshmid, $towner, $tperms, $tbytes, $tnattch, $tstatus) = split (/\s+/, $_);
        if ( $tkey =~ /^[0-9]/ ) {
           if ( $towner !~ m/root|^ / ) {
           print "$CALL_SCPT $host: Found IPC_SHR_MEM allocation: $tshmid $towner, terminating...\n" if $DEBUG;
```

```
            $CMD_LINE="ipcrm -m $tshmid";
            $RETURN=‘$CMD_LINE‘;
           }
        }
   }
   close IPC_SHARMEM;
}

# This subroutine will parse the process list and terminate any user processes or logins
# into the node(s)
sub CleanUpProcesses {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $tppid;
my $tcpu;
my $tstime;
my $ptty;
my $ttime;
my $tcmd;
my @TERM_USER;
my @TEMP;
my $USER;
my $CMD_LINE;
my $RETURN;

$AVOID_UIDS="root|100|101|nobody|bin|ntp|UID|daemon|postfix|vtunesag";
   open (PS_CMD, "-|") || exec ’ps -ef’;
   while () {
     chomp;
     ($_userid, $tpid, $tppid, $tcpu, $tstime, $ptty, $ttime, $tcmd) = split (/\s+/, $_);

       if ( $_userid !~ m/^${AVOID_UIDS}/ ) {
          if ( $_userid =~ /^[0-9]/ ) {
             $_userid=‘ypcat passwd | egrep ${_userid} | cut -d \":\" -f 1‘;
             chomp $_userid;
          }
          print "$CALL_SCPT $host: Found leftover processes $tcmd from $_userid terminating...\n" if $DEBUG;
          $CMD_LINE="pkill -9 -u $_userid";   # Switched to pkill due to length of usernames.
          $RETURN=‘$CMD_LINE‘;
       }
```

```
   }
   close PS_CMD;
   system("/root/oom_adj.user.pl");
}
# This subroutine will remove any temporary files created by MPI application under /tmp.
sub CleanUpTmp {
my $_filename;
my @TEMP;
my @TERM_FILE;
my $CMD_LINE;
my $RETURN;
my $_nofiles;
my $FILE;


   open (LS_CMD, "-|") || exec 'ls /tmp';
   while () {
      chomp;
      ($_filename) = split (/\s+/, $_);
      if ( $_filename =~ m/^mpd/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
      }
      elsif ( $_filename =~ m/^ib_pool/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
         }
      elsif ( $_filename =~ m/^ib_shmem/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
         }
   }
   close LS_CMD;

   foreach $FILE (@TERM_FILE) {
      $CMD_LINE="rm -f /tmp/${FILE}";
      $RETURN=`$CMD_LINE`;
   }

   $_nofiles = scalar @TERM_FILE;
      if ($_nofiles ne 0) {
```

```
          print "$CALL_SCPT $host: Found $_nofiles MPI temp files under /tmp. Removing...\n" if $DEBUG;
      }
}

# Flush the Linux IO buffer cache and the slab cache using the  bcfree command.
sub FreeBufferCache {
my $CMD_LINE;
my $RETURN;
my $BCFREE;
my $BCFREE_OPTS;

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

   if (-e "${BCFREE}") {
      $CMD_LINE="${BCFREE} ${BCFREE_OPTS}";
      $RETURN=‘$CMD_LINE‘;
   }
}

# This subroutine will remove any temporary files created by MPI application under /dev/shm.
sub CleanUpshm {
my $_filename;
my @TEMP;
my @TERM_FILE;
my $CMD_LINE;
my $RETURN;
my $_nofiles;
my $FILE;


   open (LS_CMD, "-|") || exec 'ls /dev/shm';
   while () {
      chomp;
      ($_filename) = split (/\s+/, $_);
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
   }
   close LS_CMD;

   foreach $FILE (@TERM_FILE) {
```

```
      if (${FILE} !~ m/sysconfig/) {
         $CMD_LINE="rm -rf /dev/shm/${FILE}";
         $RETURN=‘$CMD_LINE‘;
         print "${RETURN}" if $DEBUG;
         print "$CALL_SCPT $host: Found ${FILE} dir/file under /dev/shm. Removing it...\n" if $DEBUG;
      }
   }

}

sub chk_msr_state {
# Hyperthreading Assumption, if the first core has the bit set to enable/disable
# then it is assumed all other cores within the node have the same setting.

my $msr_lsmod=‘lsmod | grep -c msr‘;    # 0=not loaded, 1=msr loaded

   if ( $msr_lsmod == 0 ) {
      print "Loading MSR Kernel Modules...\n";
      ‘modprobe msr‘;  # we need the msr kernel modules loaded to read the msr values
       sleep(1);  # give time for the msr modules to load
   }
}

sub enable_turbo_mode {
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
my $i;
my $val;
my $nval;
        chk_msr_state();
        print "${host}: Enabling turbo mode...\n";
        chomp($val = ‘$rdmsr -p 0  $msr‘);
        $val = hex("100000017");
        $nval = $val ^ $tbit;
        printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
        for ($i = 0; $i < $ncpus; $i++) {
           ‘$wrmsr -p $i $msr $nval‘;
        }
        load_system_services();
}

sub disable_turbo_mode {
```

```
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
my $i;
my $val;
my $nval;
        chk_msr_state();
        print "${host}: Disabling turbo mode...\n";
        chomp($val = ‘$rdmsr -p 0  $msr‘);
        $val = hex(16);
        #$val = hex($val);
        $nval = $val ^ $tbit;
        printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
        for ($i = 0; $i < $ncpus; $i++) {
           ‘$wrmsr -p $i $msr $nval‘;
        }
}


sub load_system_services {
my $powersave_loaded=‘ps -ef | grep -v grep | grep -c power‘;

   if ($powersave_loaded == 0 ) {
      print "${host}: Loading system services...\n";
      system("(/etc/init.d/acpid start;/etc/init.d/powersaved start)&> /dev/null");
      sleep(1);
      system("/usr/bin/powersave -f");
   }
   else {
      print "Powersaved already loaded.\n";
   }
}

sub unload_system_services {
   print "${host}: Unloading system services...\n";
   system("(/etc/init.d/acpid stop;/etc/init.d/powersaved stop)&> /dev/null");
}

sub run_cleanup {
    &CleanUpshm();
    &CleanUpTmp();
    &CleanUpIPC_Table();
    &CleanUpProcesses();
```

```
    &CleanUpProcesses();
}

sub set_processor_speed {
my $freq=shift;
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
my $i;
my $file;
     load_system_services();
     $freq = $freq * 1000;
     printf("${host}: Setting Proc Core speed to: %.3f GHz\n",($freq/1000000)) ;
     for ($i = 0; $i < $ncpus; $i++) {
       $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_min_freq";
       open FILE1, ">", $file or die $!;
          print FILE1 "$freq\n";
       close FILE1;

       $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_max_freq";
       open FILE2, ">", $file or die $!;
          print FILE2 "$freq\n";
       close FILE2;
     }
}


#
#print "$prefix_option\n";
#print "$queue_destination\n";
#
#   if ( $queue_destination =~ /^f/ ) {
#      my $b=0;
#      ($a,$set_freq) = split (/f/, $queue_destination);
#      set_processor_speed($set_freq);
#   }


# Don’t run on systems with earlier than Nehalem processors
# Based on the prefix_option set turbo mode accordingly and run node cleanup routines.
   #if( $prefix_option =~ m/TB/ ){
        #enable_turbo_mode();
        #run_cleanup();
```

```
   #}
   #elsif ( $prefix_option =~ m/JT/ ) {
      #print "* * * * ENABLE TURBO and bypass PBS Prologue and Epilogue scripts * * * *\n";
      #enable_turbo_mode();
   #}
   #elsif ( $prefix_option =~ m/NT/ ) {
      #print "* * * * Bypassing the Turbo checks and run just node clean-up * * * *\n";
      #run_cleanup();
   #}
   #elsif ( $queue_destination =~ /^f/ ) {
      #my $b=0;
      #($a,$set_freq) = split (/f/, $queue_destination);
      #set_processor_speed($set_freq);
   #}
   #elsif ( $queue_destination =~ /^reset/ ) {
         #set_processor_speed(2934);
         #disable_turbo_mode();
         #unload_system_services();
         #run_cleanup();
     #}
   #else {
      #disable_turbo_mode();
      #unload_system_services();
      #run_cleanup();
   #}

run_cleanup();
```

Installing a Highly Available System Admin Controller (SAC) or Rack Leader Controller (RLC)

On an SGI ICE X system, you can install the SAC and RLC to be highly available. Use the instructions in this appendix to enable high availability.

System Admin Controller (SAC) High Availability (HA) Solution

This section describes the HA SAC solution.

Overview

The HA SAC solution makes use of qemu-KVM virtual machines managed by `libvirt` virtualization API and the high availability (HAE) components supplied by the SLES HAE to provide a high availability environment.

The two physical machines form an HA framework to host a single virtual machine. This virtual machine acts as the SAC for the SGI ICE system.

If one physical machine fails, the other physical machine starts up the virtual machine, and the SAC is brought back up.

It is important to read the release notes for the latest information, known problems, and work arounds needed to configure the system properly. In a scenario, such as HA, where one node can reset the other, it is all the more important to read the release notes prior to attempting the installation.

Figure B-1 on page 336 shows the components in an SGI ICE X system with an HA SAC.

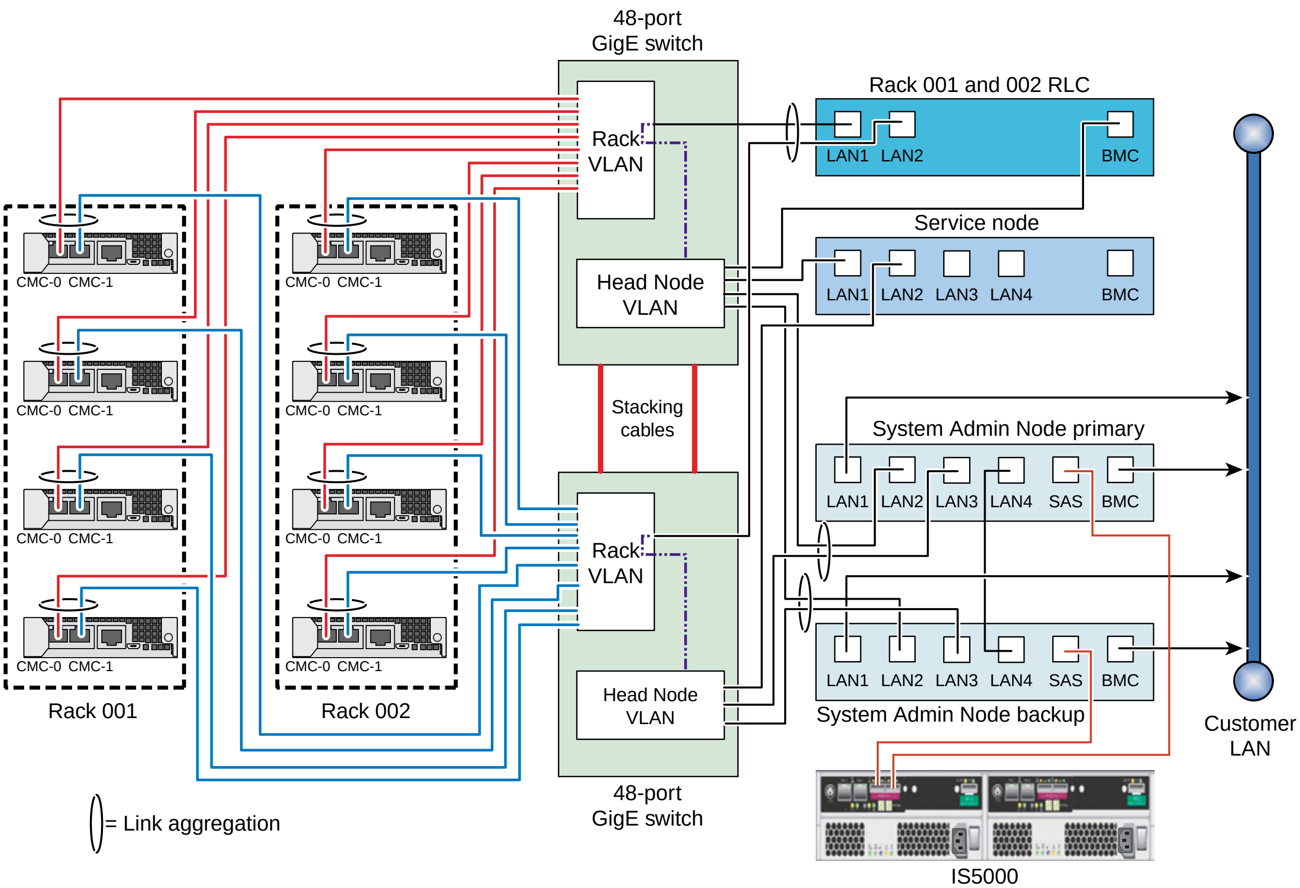

Figure B-1 SGI ICE X system with an HA SAC

Figure B-2 on page 337 shows the components in an SGI ICE X system with an HA SAC and an HA RLC.

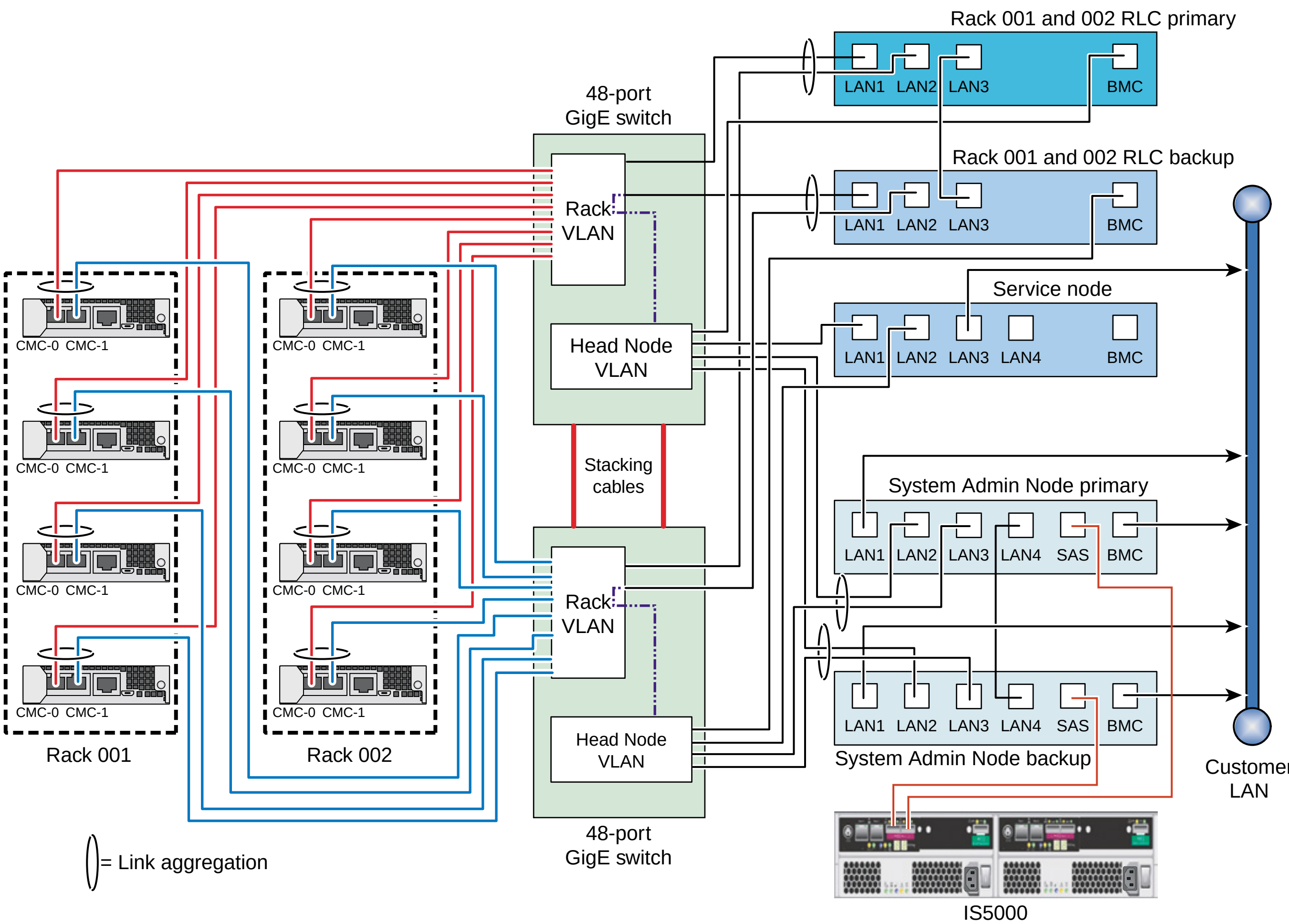

Figure B-2 SGI ICE X system with an HA SAC and an HA RLC

Hardware Requirements

The major hardware components that comprise this solution include the following:

- Two system admin controllers (SACs) with the following components
  - Add-in card with two or more Ethernet ports

  - Integrated system disks
  - Serial Attached SCSI (SAS) card for external storage
- SGI InfiniteStorage 5000 (IS5000)

For a functional diagram of the HA SAC solution, see Figure B-3 on page 338.

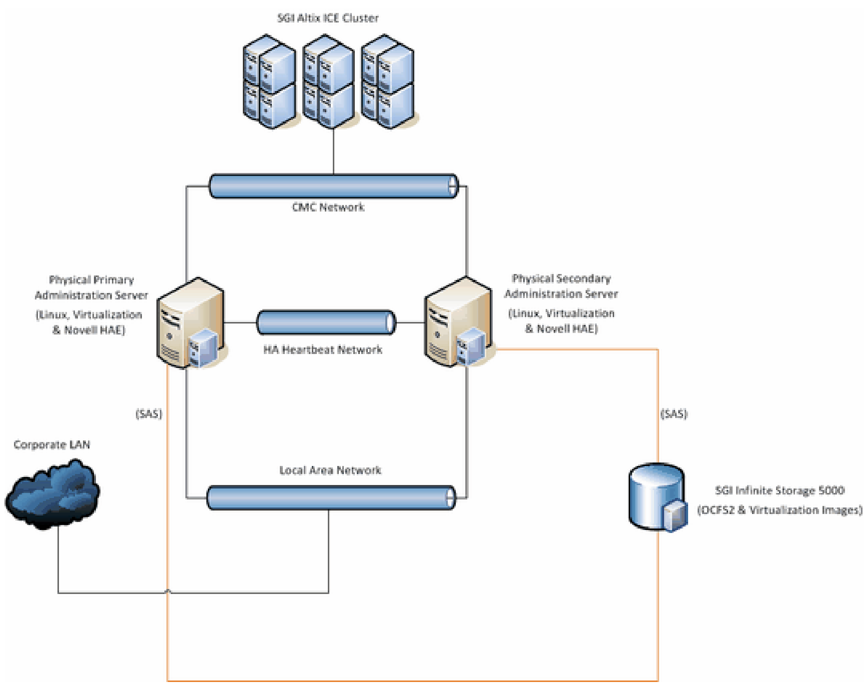

Figure B-3 HA SAC High Availability Solution Functional Diagram

Physical Connections

The network Ethernet connections are as follows:

Network Interface Card	Connection
`eth0`	House network, just as with non-HA SACs
`eth1`	ICE CMC, just as with non-HA SACs

`eth2`	ICE CMC, just as with non-HA SACs
`eth3`	Cross connection between two physical SACs

In some configurations, Ethernet switches may be necessary between the SGI ICE CMCs and the SACs due to port availability.

Software Requirements

The major software components that comprise this solution include the following:

- SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP2) base distribution
- SLES 11 SP2 High Availability Extension (HAE)
- The `vm-install` package
- The `sac-ha` package from SGI, available on SGI Management Center media
- SAC Admin Install DVD ISO from SGI

An overview of how to install the the highly available (HA) system admin controller (SAC) solution is, as follows:

- Perform a default installation of SLES11 SP2 on both physical SACs. A later step will enable the firewall.
- Be sure to disable the firewall on both physical machines.
- Then install HAE and all available SLES and HAE updates from Novell.
- Install the `vm-install` package on both physical SACs.
- Install the `sac-ha` package from SGI on both physical SACs.
- Pick a SAC to be primary and configure `sac-ha` to run the helper script.

For a detailed description of this procedure, see "HAE, Virtualization and Updates installation" on page 340.

Base SLES 11 SP2 Installation Considerations

This section provides some general instructions about SLES 11 SP2 when installing the system admin controllers (SACs), as follows:

- Boot the SLES 11 SP2 installer the way you normally would at your site.
- When asked for a scenario, choose **Physical Machine** (Also for **Fully Virtualized Guests**) for both physical SACs.
- Be sure to use the root device inside the chassis, and **not** the external storage for the installation. The external storage will only house virtual machine images in a future step. If you see any storage named SGI-IS400 or similar, that is external storage. If you see storage related to LSILOGIC or Megaraid, that ID internal storage. You want to use the root device on internal storage.
- When selecting disks, use the entire hard disk. Default values for root and swap should be sufficient.
- Install just the default SLES11 SP2 packages for now on both physical SACs.

Network Configuration

Configure the interface connected to your house network as you normally would. Do not use DHCP but rather a static configuration.

A later step will reconfigure networking and renumber the Ethernet devices to match the necessary layout. It will be told to use the interface you configure now as the reference when it reconfigures networking on the physical system admin controllers (SACs).

High Availability Extension Documentation

For information on the SUSE Linux Enterprise High Availability Extension (HAE) software package, see the SUSE *High Availability Guide* . This is required reading and the information contained in this guide is not replicated in this section.

On the physical system admin controlers (SACs), the document can be found at `/usr/share/doc/manual/sleha-guide_en-pdf`.

It is available in the `sleha-guide_en-pdf` package from Novell.

HAE, Virtualization and Updates installation

This section describes the High Availability Extension (HAE), virtualization, and updates installation, as follows:

- Add the HAE add-on product using SLES11 SP2 tools.
- Install the HAE defaults with `yast`. If using `zypper`, perform the following:

```
# zypper install -t pattern ha_sles
# zypper install sleha-guide_en-pdf
```

- Install the necessary QEMU/KVM components. If using `zypper`, perform the following:

```
# zypper install virt-manager libvirt kvm vm-install libvirt-cim virt-viewer bridge-utils
```

- Be sure to register with Novell Customer Configuration Center using YaST if you have not already. This should add update sources to your repo. This is a critical step.
- Confirm that the update repos are configured as well. Use the `zypper -lr` command. It should list the following four repos:

```
sles11sp2
sles11sp2 updates
hae
hae updates
```

- Update the system. This is a critical step because the functionality depends on updates being applied. You can do this with the `zypper` command, as follows:

```
# zypper -n update
```

At this point, both physical system admin controllers (SACs) have the exact same software stack installed including SLES11 SP2, HAE, and all updates applied.

Reboot both SACs so that you are running with new kernels, updates, and so on.

Before proceeding to the next section, confirm that both physical SACs are now up and running and both with their house network connections configured.

Install SGI Helper Scripts (sac-ha Package)

From the SGI Management Center media, copy the RPM named `sac-ha` from the media on to both physical system admin controllers (SACs).

Install the package on both physical SACs. This provides the scripts to configure both systems.

Configure the SGI Helper Scripts

On the physical system admin controller (SAC) that you want to use as primary, edit the file `/etc/opt/sgi/sac-hae-initial-setup.conf`.

On the second physical SAC, no configuration is necessary. Configuration is all done from the first SAC.

Make sure you read all of the comments in this file. The comments provide the instructions for filling out the file, which include:

- Instructions for ordering the MAC addresses
- Instructions for locating physical Ethernet ports if you are unsure how they are laid out.
- Instructions for locating and defining your shared external storage
- Instructions for filling out all the network information that is required
- You will need to know the SGI ICE Head Network you intend to use in advance. Most sites do not change from the default so the examples provided should be sufficient.

Once the configuration file is filled out entirely, you can run `/opt/sgi/sbin/sac-hae-initial-setup` command.

This script will reconfigure networking on both physical nodes and then will do the HAE setup, and so on. You will be prompted a few times in the early part of the process. These prompts are related to getting the ssh keys and known hosts configuration set up and will involve entering the root password and/or the word **yes** as prompted. After `ssh` is set up, the rest of the scripts should run automated without input.

There are places where this process can fail. It is safe to rerun the scripts if they fail in the network set up phase. However, once HAE components are configured, reconfiguring networking can result in machines resetting themselves or each other for protection.

If you do restart the scripts to configure networking, pay special attention to keeping the `phys1_current_house` and `phys2_current_house` variables up to date in the `/etc/opt/sgi/sac-hae-initial-setup.conf` file. They should always point to the currently configured house interface as it preserves the network settings and applies them to the proper interfaces.

For information about shared disk device selection, see the comments in /etc/opt/sgi/sac-hae-initial-setup.conf regarding choosing the shared storage device.

The /etc/opt/sgi/sac-hae-initial-setup.conf file describes how to choose MAC addresses. It is important that you work out the MAC address ordering as described as the network will be reconfigured on both physical SACs to match. See the configuration file's notes about using ethtool to help you figure out which physical port matches a MAC address.

Accessing crm_gui and the hacluster Account

By default, the SGI SAC HA scripts set the password for the hacluster account to sgisgi. You will need this password if you run the crm_gui tool and wish to log in as the default hacluster user in that tool.

Virtual Machine Installation

After the sac-hae-initial-setup script completes, the virtual machine for the system admin controller (SAC) should be running on one of the two physical machines. You can identify which one by using the crm_gui command and looking at the virt-clone resource (or similarly use the crm command). If it is not started for some reason, go ahead and start the service.

Next, run the virt-manager command on the physical machine that is currently hosting the guest.

It should be at the Admin Node Installation DVD prompt. From this point forward, installation of SGI Management Center ICE is the same as it is on real hardware.

Allocate Physical Node IP Addresses with SGI Management Center

After the system admin controller (SAC) virtual machine has been installed and the execution of the configure-cluster command is complete, you need to allocate the IP addresses used by the physical SACs to avoid an IP address conflict. This should be done prior to discovering any nodes.

In the example below, phys1 is the HEAD network IP for physical machine 1 and phys2 is for physical machine 2. The IP addresses provided match the IP address used with the SAC HA configuration scripts you ran earlier:

```
# discover --service 98,other
# cadmin --set-ip --node service98 --net head service98=172.23.254.253
# cadmin --set-hostname --node service98 phys1

# discover --service 99,other
# cadmin --set-ip --node service99 --net head service99=172.23.254.254
# cadmin --set-hostname --node service99 phys2
```

High Availability Customization

The sac-ha-initial-setup script sets up a basic default set of high availability (HA) rules for the cluster. They are meant to be a starting point for your site.

SGI will likely come up with additional suggested HA rules and configuration details over time. SGI will update the sac-hae-initial-setup.conf file for new releases but since that file is only used once ever, you could miss out on configuration updates.

SGI intends to also release HA configuration suggestions as service bulletins available on SGI SupportFolio.

SMC Licenses

Many of the components in SGI Management Center require a license. The license is associated with the virtual machine, and not the physical machine. For this reason, you will need to request the license based on the virtual machine environment once SGI Management Center ICE is installed.

Virtual System Admin Controller (SAC) Guest Service

The default HAE rules supplied by SGI will ensure a virtual machine named sac is always running. If the machine shuts down, it will be restarted on one of the physical machines.

The machine can be accessed with the virt-manager command from the physical machine.

In to avoid unexpected results, it is best to use crm_gui or crm to stop the virt-clone service when you wish to shut down the SAC virtual machine. Likewise, use crm or crm_gui to turn the service back on when desired. If you use virt-manager to shut the machine down, HAE will turn it back on for you automatically.

Virtual Admin Node Resource

The default csync2 configuration set up by SGI will keep your system admin controller (SAC) virtual machine configuration up to date on both nodes. However, csync2 is not automatic.

Therefore, if you make adjustments to the virtual machine configuration, such as CPU or memory use, you should run csync2 -xv to synchronize to the other physical SAC.

Some sites may prefer a more targeted approach to the csync2 configuration as it relates to the libvirt configuration files.

One thing to keep in mind is that you need the virtual machine configuration files to be the same on both physical SACs because the guest needs to run on both. This also helps facilitate live migration of the running virtual machine (VM) from one physical host to the other.

For more information on csync2, see SUSE Linux Enterprise High Availability Extension (HAE) *High Availability Guide*.

Special Virtual Machine Guest Configuration: SGI Emulator Wrapper Script

The SGI configuration helper scripts set up the virtual machine guest for you in the correct manner. There is one thing the script puts in the XML configuration file that is hard to specify in virt-manager. This is the <emulator> tag.

If you create your own virtual machine with virt-manager, it will not use the /opt/sgi/sbin/sgi-emulator-wrapper wrapper script but instead have a path to qemu-kvm.

It is important that the machine use the SGI sgi-emulator-wrapper wrapper script because this script provides proper serial number information to the virtual machine.

System admin controllers (SACs) have a special system-wide SGI ICE serial number embedded in them. The simple wrapper script for the emulator passes on this serial number information to the guest SAC.

Migrating a Virtualized System Admin Controller (SAC) Live

Using the `crm_gui` or `crm` command line, it is possible to migrate the virtual machine live from one physical machine to the other. This can be useful, for example, if the system administrator needs to perform maintenance on one physical machine.

Note that this migration method makes use of the `ssh` transport. If your configuration does not allow the root user to ssh between physical machines without a password, you will need to modify the `virt-clone` rules and associated primitives to use a different transport. This will likely involve the set up of certificates for authentication for `libvirt` between the two physical machines. That is not covered in this section.

You can use the command line or GUI to migrate a virtualized SAC. This section describes the GUI method.

Procedure 7-1 Migrating a Virtualized SAC Live

To use the `crm_gui` GUI to migrate to a virtualized SAC, perform the following:

1. Start `crm_gui` on one of the two physical machines.
2. Make sure you are in the **Management** section (click **Management** on the left panel).
3. On the right panel, locate **virt-clone**.

   It should show that it is running with a green status indicator. The **virt:0** should display with the node on which it is currently running.
4. Right-click on **virt-clone**, then choose **Migrate Resource**.
5. Choose the system to which you want to migrate.

   The live migration is initiated.
6. You can confirm it completed successfully by watching `/var/log/messages` on the machine that was previously running the virtual machine.
7. You can use **virt-manager** to connect to the machine and confirm it is still up and running.

8. It is sometimes necessary to use do a resource cleanup to see the resource in `crm` again. If **virt-clone** shows as not running after waiting a couple minutes, then right-click on it and select **cleanup resource**, click on OK. The resource should go back to green.

9. Any time you migrate, a constraint is recorded. If you wish to remove the constraint, right-click on the **virt-clone** entry and select **Clear Migrate Constraints**, then click OK.

Rack Leader Controller High Availability Solution

High Availability Rack Leader Controllers (leader nodes) for SGI ICE systems provide redundant rack leader controllers for SGI ICE racks. A rack leader controller can be taken offline or can fail, and the `tmpfs` booted rack compute nodes will remain fully functional.

This initial implementation of high availability rack leader controllers (HA-RLC) for SGI ICE clusters has two HA-RLCs per rack.

Novell, Inc. high availability software (HAE) running on each rack leader controller (RLC) manages a set of services that are moved from one RLC to the other RLC if the HAE software detects a failure in a monitored service.

Hardware Requirements

This implementation of HA-RLC for SGI ICE requires two lead nodes per rack and an extra network interface card (NIC) per HA-RLC. The extra NIC provides a direct network connection between the HA-RLC for the HAE software.

Software Requirements

HA-RLC is currently implemented only on SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP2) using the SLES 11 SP2 High Availability Extension (HAE). This restriction applies to only the admin and the rack leader nodes.

All current updates to package in the SLES 11 SP2 distribution (distro) and the SLES11 SP2 HAE product must be available to install a HA-RLC SGI ICE cluster.

The major software components that comprise this solution include the following:

- SUSE Linux Enterprise Server 11 Service Pack 1 (SLES 11 SP2) base distribution
- SLES 11 SP2 High Availability Extension (HAE)
- SGI Admin Node Autoinstallation DVD for SLES11 SP2
- SLES 11 SP2 distribution updates
- SLES 11 SP2 HAE updates

It is **required** that updates to the SLES11 SP2 distro and to the SLES11 SP2 HAE be used with HA-RLC, in particular, when creating the HA-RLC leader images.

SLES11 SP2 High Availability Extension (HAE) Documentation

For information on the SUSE Linux Enterprise High Availability Extension (HAE) software package, see the*SUSE Linux Enterprise High Availability Extension High Availability Guide.* This is required reading and the information contained in the SLES HAE guide is not replicated in this manual.

The HAE document is available in the `sleha-guide_en-pdf` package from Novell, Inc.

After the repositories are created, updated, and the HA Repository selected, this package can be installed on the admin node, as follows:

```
admin:~ # cinstallman --yum-node --node admin install sleha-guide_en-pdf
```

You can find the HAE guide at the following location:

`/usr/share/doc/manual/sleha-guide_en-pdf/SLEHA-guide_en.pdf`

Compute Node `tmpfs` Boot Requirement

It is required that all compute nodes in HA-RLC protected racks use `tmpfs` roots, which implements a memory resident operating system. With the `tmpfs` root, no compute node state information is stored on a RLC, as is the case when the compute nodes use NFS roots.

Warning: It is critical that the HA-RLC compute nodes be booted with a `tmpfs` root. Compute nodes booted with NFS root will crash if a HA-RLC fails.

Since compute nodes using `tmpfs` roots store no state information on a RLC, the HA-RLC can be rebooted, and protected services migrated, without affecting the compute node functionality.

Physical Connections

The connection of the rack leader controller (RLC) to the racks must be via the `LOCAL` port of the individual rack unit (IRU) chassis management controller (CMC).

In the case of an RLC with redundant management network (RMN) and a separate BMC port, it means that there are three Ethernet connections to the rack, two of which go to `LOCAL` ports on the CMC and one (the BMC connection) that goes to an `LL` port. In order to have four `LOCAL` ports in a rack, there must be four IRUs in that rack.

In addition, there is a direct Ethernet connection between the two HA-RLCs of each rack. This provides a communication path between HA-RLC that does not pass through the stacked CMCs and avoids cluster management traffic (see Figure B-4 on page 350).

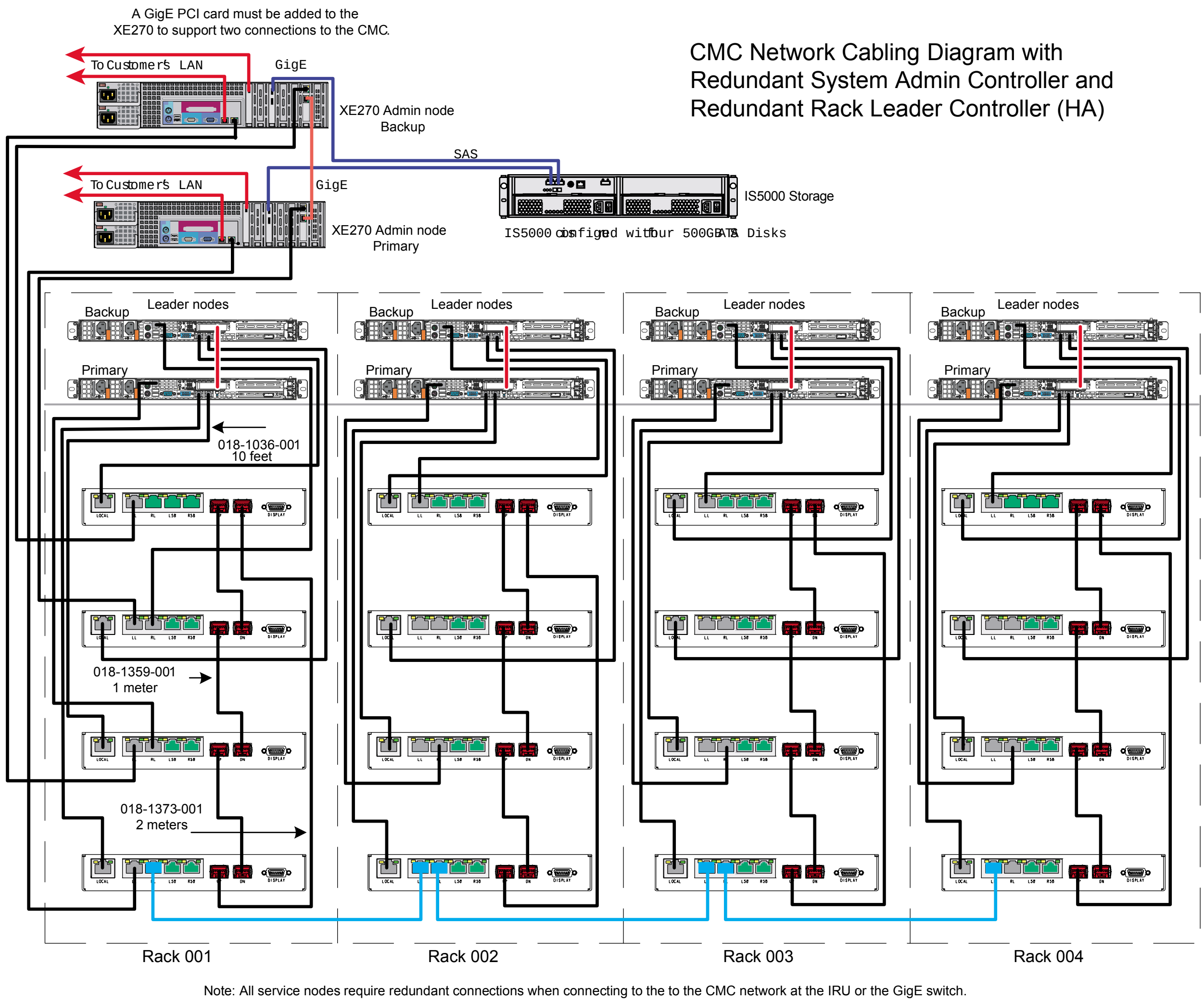

Figure B-4 CMC Network Cabling Diagram for HA-RLC

High Availability Extension

The SUSE Linux Enterprise High Availability Extension (HAE) software manages *resources*. Examples of managed resources include `blademond`, `dhcpd`, and `conserver`. The HAE software stops, starts and monitors these resources. Only a single instance of these resources runs per rack. These three resources also happen to be Linux Standards Base (LSB) services.

With HA-RLC, all resources are co-located on one or the other HA-RLCs. The HA-RLC on which the resources are running can be said to "host" the resources and can be referred to as the "active" HA-RLC.

The resources can be migrated between the two HA-RLCs for a rack, either in response to administrative commands or in response to a failure detected with one of the managed resources.

For detailed information on HAE software, see the *SUSE Linux Enterprise High Availability Extension High Availability Guide* (see " SLES11 SP2 High Availability Extension (HAE) Documentation" on page 348).

HA-RLC Naming

The HA-RLC hostnames and interfaces for a rack are distinguished from each other with the addition of a "1" or a "2" to the standard RLC interface names.

For example, here are the names of the HA-RLC interfaces for racks 1 and 2. The names of the interfaces for racks 3 to n follow this same naming scheme.

The Rack 1 HA-RLC hostnames and interfaces are named, as follows:

- `r1lead`

  Logical interface to the rack 1 RLC. Migrates.

- `r1lead1`

  Host interface to the first of the two RLCs for rack1. Associated with the physical device. Does not migrate.

- `r1lead1-bmc`

  BMC interface to the first of the two RLCs for rack1. Associated with the physical device. Does not migrate.

- `r1lead2`

Host interface to the second of the two RLCs for rack1. Associated with the physical device. Does not migrate.

- r1lead2-bmc

  BMC interface to the second of the two RLCs for rack1. Associated with the physical device. Does not migrate.

The Rack 2 HA-RLC interfaces are named, as follows:

- r2lead

  Logical interface to the rack 2 RLC. Migrates.

- r2lead1

  Host interface to the first of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- r2lead1-bmc

  BMC interface to the first of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- r2lead2

  Host interface to the second of the two RLCs for rack2. Associated with the physical device. Does not migrate.

- r2lead2-bmc

  BMC interface to the second of the two RLCs for rack2. Associated with the physical device. Does not migrate.

HA-RLC Networking Changes

Changes to default SGI ICE networking were needed to implement HA-RLC. In particular, HA-RLC addresses on the VLAN_HEAD, VLAN_BMC, and VLAN_GBE networks have been implemented as IP aliases that are managed as HAE resources.

Each HA-RLC also has a physical address via which the node is accessible from the admin node, regardless of where the HAE resources are hosted.

The IP alias address associated with r1lead is a logical rack alias and requests to r1lead will be directed to whichever HA-RLC is currently hosting the resources.

In contrast, the IP physical addresses associated with `r1lead1` and `r1lead2` always connect to that node. The BMC `console` connect is via the physical address.

```
admin:~ # console r1lead1
```

The console connections to the physical address are not disrupted by migration of the resources. The `ssh` connections to the physical addresses are also not disrupted by migration of resources. The `ssh` connections to the logical addresses are disrupted by migration.

Figure B-4 on page 350 shows the CMC network cabling diagram with a redundant system admin controller and redundant rack leader controller.

HAE with HA-RLC

Each pair of HA-RLC comprise a separate instance of a HAE cluster. There is no communication between the separate HAE instances, that is, HAE instances running on different racks are independent.

The HAE software controls resources. LSB `init` scripts such as `blademond`, `dhcpd`, and `conserver` map directly to HAE resources.

HAE supplied scripts can be used for multiple HAE resources. For example, the script `ocf::heartbeat:IPaddr2` is used to implement the three IP aliases used by HA-RLC.

All HA-RLC HAE resources are *co-located*, which means that all resources run simultaneously on one or the other of each HA-RLC pair.

A `stonith-ipmi` resource runs on each of the HA-RLCs. These `stonith` resources do not migrate with failure. The `stonith-ipmi` resources monitor peer BMC, but `stonith` action is not enabled.

If there is a failure of one of the HAE managed resources, the resources are all migrated to the other HA-RLC. Resources can also be administratively migrated between the HA-RLC of a rack.

As the HAE instance running on a rack is independent of HAE instances running on other racks, resources cannot be migrated to a HA-RLC of a different rack.

The HAE configuration is loaded when the HA-RLC boot for the first time. It is not intended that configuration be changed except via SGI update.

Warning: Making changes to the SGI supplied HAE resources is not supported. Changes to HAE resources are expected to be done only with SGI release upgrades.

Installing the ICE Admin Node for a HA-RLC Cluster

Install the ICE Admin Autoinstall SLES 11 SP2 DVD as described in Chapter 3, "Installing and Configuring an SGI ICE X System" on page 47.

Do **NOT** proceed to the steps described in the "Running the Cluster Configuration Tool" on page 61 section until you have finished creating repositories, as described in "Create SLES11 SP2 and HAE Repositories" on page 354.

Create SLES11 SP2 and HAE Repositories

Before running the `configure-cluster` command for a cluster that will include HA-RLC, you **must** create updated repositories for the SLES 11 SP2 distro and for SLES 11 SP2 HAE.

Warning: Updated repositories are required to build working HA-RLC images

Since the default leader image is created by `configure-cluster`, it is necessary that the updated repositories exist before running `configure-cluster`.

Create updated repositories for the distro and HAE using the method of your choice. The method of obtaining updates with which to create update repositories will vary from site to site. An example for creating these repositories from local update sources follows.

Use the `crepo --select` command to select the repositories so that they are used when building the default images.

Update Repositories Example

In the example in this section four separate repositories are created, as follows:

- SLES 11 SP2 distro
- SLES 11 SP2 HAE
- SLES 11 SP2 distro updates
- SLES 11 SP2 HAE updates

After these repositories are created, the four repositories must be marked as `selected` using the `crepo` command.

The SLES 11 SP2 distro and SLES 11 SP2 HAE GA release packages are obtained from the respective ISOs in a local or NFS mounted directory.

The SLES 11 SP2 distro update packages and the SLES 11 SP2 HAE update packages are obtained (from http://www.novell.com/home/, a support contract is needed, see "SLES System Admin Controllers (SACs): Update the SLES Package Repository" on page 125) and are expected to be in separate directories, either local or NFS mounted.

The bash shell is used in the following examples.

Procedure 7-2 Create Software Repositories for HA-RLC

To create software repositories for installing the HA-RLC solution, perform the following steps:

1. Create SLES 11 SP2 distro and SLES 11 SP2 HAE repositories from ISOs in the current local directory (for reference, see `crepo --help`), as follows:

```
admin:~ # crepo --add SLES-11-SP2-DVD-x86_64-GM-DVD1.iso
admin:~ # crepo --add SLE-11-SP2-HA-x86_64-GM-Media1.iso
```

2. Create a SLES11 SP2 distro updates repository from the updates packages located in the local `SLES11-SP2-Updates/sle-11-x86_64/rpm`, as follows:

```
admin# cd SLES11-SP2-Updates/sle-11-x86_64/rpm
admin# find  -name "*.rpm" | egrep -v '\.delta\.rpm$|\.patch\.rpm$|\.src\.rpm$|\.nosrc\.rpm$' | \
      sort > /tmp/distro-updates-list-$$

admin# DISTRO_UPDATES_DIR=/tftpboot/updates/sles11sp2-updates
admin# mkdir -p $DISTRO_UPDATES_DIR

admin# for f in ‘cat /tmp/distro-updates-list-$$‘; do cp -a $f $DISTRO_UPDATES_DIR; done
admin# crepo --add $DISTRO_UPDATES_DIR --custom "sles11sp2-updates"
```

3. Create a SLES11 SP2 HAE updates repository from the updates packages located in the local `SLES11-HAE-SP2-Updates/sle-11-x86_64/rpm`, as follows:

```
admin# cd SLE11-HAE-SP2-Updates/sle-11-x86_64/rpm

admin# find -name "*.rpm" | egrep -v '\.delta\.rpm$|\.patch\.rpm$|\.src\.rpm$|\.nosrc\.rpm$' | \
        sort > /tmp/hae-updates-list-$$

admin# HAE_UPDATES_DIR=/tftpboot/updates/SUSE-Linux-Enterprise-High-Availability-Extension-11-SP2-updates
admin# mkdir -p $HAE_UPDATES_DIR

admin# for f in ‘cat /tmp/hae-updates-list-$$‘; do cp -a $f $HAE_UPDATES_DIR; done
admin# crepo --add $HAE_UPDATES_DIR --custom "SLES-HAE-updates"
```

4. The newly created repositories must be `selected` so that they are used when building the default images (for reference, see `crepo --help`), as follows:

```
admin:~ # crepo --select SUSE-Linux-Enterprise-Server-11-SP2
admin:~ # crepo --select sles11sp2-updates
admin:~ # crepo --select SUSE-Linux-Enterprise-High-Availability-Extension-11-SP2
admin:~ # crepo --select SLES-HAE-updates
```

5. To display the currently selected repositories, perform the following:

```
admin:~ # crepo --show
* SLES-HAE-updates : /tftpboot/updates/SUSE-Linux-Enterprise-High-Availability-Extension-11-SP2-updates
* SUSE-Linux-Enterprise-High-Availability-Extension-11-SP2 : /tftpboot/other/SUSE-Linux-Enterprise-
                      High-Availability-Extension-11-SP2
* SUSE-Linux-Enterprise-Server-11-SP2 : /tftpboot/distro/sles11sp2
* sles11sp2-updates : /tftpboot/updates/sles11sp2-updates
```

Note: The leading "`*`" indicates that the repository is currently "`selected`".

Initial Cluster Configuration with HA-RLC

The Cluster Configuration Tool (`configure-cluster`) is used to perform initial cluster configuration (see "Running the Cluster Configuration Tool" on page 61).

It is required when installing a cluster that will include HA-RLC racks that the SLES11 SP2 distro and SLES11 SP2 HAE repositories, and update repositories, be created before running the `configure-cluster` command.

Although it is possible to recover from the failure to create the updates repositories before running `configure-cluster`, this is not the supported method of installing HA-RLC clusters.

The `configure-cluster` instructions in this section are a supplement to the installation instructions provided in "Running the Cluster Configuration Tool" on page 61.

Procedure 7-3 Initial Cluster Configuration with HA-RLC

To configure a HA-RLC cluster, perform the following steps:

1. Use the `configure-cluster` command to install the following embedded ISOs:

- SGI-Foundation-Software
- SGI-Management-Center
- SGI-Tempo

  SLES 11 SP2 distro and SLES11 HAE repositories, plus updates repositories for these products, should have already been created (see "Create SLES11 SP2 and HAE Repositories" on page 354). You do not need to recreate them.

2. Install the Admin cluster software (see Chapter 3, "Installing and Configuring an SGI ICE X System" on page 47).
3. Using `configure-cluster`, confirm default network settings or make local modifications. These settings cannot be changed after the cluster is initialized.
4. Configure the locally accessible NTP server.
5. Do not proceed unless the update repositories were created, as described "Create SLES11 SP2 and HAE Repositories" on page 354. Using `configure-cluster`, perform the infrastructure setup.

   The admin node log file of particular interest during the image creation process is `/var/log/cinstallman`.

   Note: This step can take 30 minutes or more to complete

   ---

6. Using `configure-cluster`, configure the local DNS resolvers

Discover HA—RLC Racks

After completing the **Initial Cluster Setup Tasks** with `configure-cluster` described in "Initial Cluster Configuration with HA-RLC" on page 357, the HA-RLC racks can be installed.

Discover HA-RLC Rack Overview

During the `discover rack` operations, the following events occur:

- HA-RLCs are installed.
- HAE software is initialized on the HA-RLCs.

- Default compute node images are downloaded to the HA-RLCs and are installed.
- Compute nodes are booted.

The `discover` command discovers the MAC addresses of the HA-RLC and configures the admin node for the installation of the HA-RLC. The `discover` process exits, leaving the installation of the HA-RLC to continue. The remainder of the `discover rack` operations are started when the HA-RLC boot.

The admin log file of particular interest for HA-RLC discovery and installation operations is, as follow:

admin:~ # **tail -f /var/log/messages**

In particular, monitor the messages related to DHCP and `tftp`.

Each pair of HA-RLC will boot into HA mode, and one (and only one) of the HA-RLC will start hosting the resources for each rack.

When the HAE software comes on-line, `blademond` (one of the HAE managed resources) discovers the rack hardware (CMCs, compute nodes) and pushes the rack information to the admin node. The admin node then configures the HA-RLC and pushes compute node images to the HA-RLC.

After the HA-RLCs are fully configured, the compute nodes boot from the HA-RLC.

Discover Both HA-RLCs in a Rack

SGI recommends that whenever both HA-RLCs of a rack are to be discovered, the HA-RLC be discovered together.

Either of the following commands will install both rack HA-RLCs.

Discover two HA-RLCs for rack 1, as follows:

admin:~ # **discover --rack 1,ha=all**

Discover two HA-RLCs per rack for racks 1, 2, and 3, as follows:

admin:~ # **discover --rack 1,3,ha=all**

The HA-RLCs are discovered sequentially by the `discover` command. For a single rack, the `discover --rack 1,ha=all` command requests that power be applied first to `r1lead1` and then will request that power be applied to `r21lead2`. "Apply power" means to plug-in the HA-RLC, but **not** to push the power button.

For more information on the `discover` command, see "`discover` Command" on page 84.

Monitor HA-RLC Installation

You can follow the installation of the HA-RLCs by opening BMC consoles for the two HA-RLC from different admin node windows or from different window tabs. For example, to follow the installation of the rack 1 HA-RLC, run these commands in different windows:

```
admin:~ # console r1lead1
admin:~ # console r1lead2
```

Monitor the admin node messages file for HA-RLC install messages, in particular look for the initial DHCP and tftp messages, as follows:

```
admin:~ # tail -f /var/log/messages
```

Confirm HAE After HA-RLC Boot

When the rack leader nodes boot for the first time, the HAE servers will initialize and one of the HA-RLC pair will start the HAE resources. Generally, this will be the first leader node discovered for the rack.

Use the `crm_mon` command to determine which HA-RLC is hosting the resources. This command can be run via `ssh` from the admin node or from a login to either of the HA-RLCs, as follows:

```
admin:~ # ssh r1lead crm_mon -1
============
Last updated: Mon Apr 18 14:14:22 2011
Stack: openais
Current DC: r1lead1 - partition with quorum
Version: 1.1.5-5ce2879aa0d5f43d01629bc20edc6868a9352002
2 Nodes configured, 2 expected votes
7 Resources configured.

============

Online: [ r1lead1 r1lead2 ]

blademond       (lsb:blademond):        Started r1lead1
conserver       (lsb:conserver):        Started r1lead1
```

```
dhcpd  (lsb:dhcpd):    Started r1lead1
Resource Group: alias-group
     head-alias (ocf::heartbeat:IPaddr2):       Started r1lead1
     vlan1-alias        (ocf::heartbeat:IPaddr2):       Started r1lead1
     vlan2-alias        (ocf::heartbeat:IPaddr2):       Started r1lead1
stonith-ipmi-1 (stonith:external/ipmi):        Started r1lead1
stonith-ipmi-2 (stonith:external/ipmi):        Started r1lead2
```

It takes several minutes for HAE to initialize and for the resources to be started.

Until blademond initializes the dhcpd.conf file (actually, ice.conf), crm_mon reports DHCP resource errors.

Until blademond probes the rack CMC the slot maps and the compute nodes and the conserver configuration file has been initialized, crm_mon reports errors for the conserver resource.

There should be no stonith-ipmi errors. There should be no stonith-ipmi resources if only one HA-RLC is installed. Errors are reported for the stonith-ipmi resource if an HA-RLC has been discovered and is installed but has not booted.

Monitor Compute Node Set up

On the active HA-RLC (indicated by the resource being Started r1lead1 in the output of crm_mon -1), the blade monitoring daemon blademond starts, interrogates the rack CMC for blade (slot) information, configures dhcpd and starts the discover-rack process on the admin node.

The discover-rack process running on the admin node controls the configuration of the HA-RLC for booting the compute nodes.

Progress can be monitored from the following log files on the admin node and on the active HA-RLC (in this example, assumed to be r1lead1):

```
r1lead1:~ # tail -f /var/log/blademond
admin:~ # tail -f /var/log/discover-rack
```

Completion of HA-RLC configuration for booting the compute nodes occurs when the following message is logged in the /var/log/discover-rack file:

```
setup-new-blades: booting blades finished
```

Completion messages will also be logged on the "active" HA-RLC in `/var/log/blademond` file and the log will indicate that the `blademond` is, as follows:

```
Jun 17 21:39:25 r1lead1 blademond: calling discover-rack on admin node
Jun 17 21:45:41 r1lead1 blademond: moving /var/opt/sgi/lib/blademond/slot_map.new to
                                   /var/opt/sgi/lib/blademond/slot_map
Jun 17 21:45:42 r1lead1 blademond: INFO: pushing /var/opt/sgi/lib/blademond/slot_map to
                                   r1lead2:/var/opt/sgi/lib/blademond
Jun 17 21:45:43 r1lead1 blademond: INFO: pushing /etc/dhcpd.conf.d/ice.conf to
                                   r1lead2:/etc/dhcpd.conf.d/ice.conf
Jun 17 21:45:43 r1lead1 blademond: sleeping ...
```

Confirm Boot of Compute Nodes

To confirm the boot of the compute nodes in rack 1, use a command similar to the following:

admin:~ # **cexec rack_1: uptime**

You can also use the `cpower` command to confirm that the compute nodes have booted. For example, for rack 1, perform the following:

admin:~ # **cpower --status r1i*n***

Confirm Initial HA-RLC Setup

The two HA-RLCs for a rack must always be configured to boot the same images. This can be confirmed by comparing the compute node boot links in `/tftpboot/pxelinux.cfg` of the two HA-RLCs.

With the initial install, all links will point to the `tmpfs` version of the default compute image (`compute-sles11sp2`) boot file.

For example:

```
admin:~ # pdsh -w r1lead1,r1lead2 ls -l /tftpboot/pxelinux.cfg/ | \
awk '{print $1,$9,$10,$11}' | dshbak -c
----------------
r1lead[1-2]
----------------

C0A89F0A -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F19 -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1A -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1B -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1C -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
C0A89F1D -> /tftpboot/compute-sles11sp2/pxelinux.config-2.6.32.29-0.3-default-tmpfs
```

Note that the hex number is the compute node IP address.

The links will only be present when a HA-RLC has been fully configured to boot the compute nodes.

If compute nodes are later set to boot different images, the links for those compute nodes will point to the different boot files. The links are the same on both HA-RLCs, the boot files will be `tmpfs` and not `nfs`.

Sequential Discovery of HA-RLC

The discovery of a single HA-RLC makes it possible to add a replacement HA-RLC to the cluster. The feature also makes it possible to first discover one in a pair of HA_RLCs and later discover the other HA-RLC pair.

SGI recommends that the initial HA-RLC discovery be of both of the rack HA-RLCs, that is, using `ha=all`.

For rack 1, discover first one of the HA-RLC and then discover the second HA-RLC. For example, first discover `r1lead1` and after `r1lead1` is fully configured, HAE is running, and the compute nodes have booted, then discover the second HA-RLC, `r1lead2`.

To discover HA-RLC r1lead1, perform the following:

```
admin:~ # discover --rack 1,ha=1
```

After the configuration of `r1lead1` completes, discover HA-RLC `r1lead2`, as follows:

```
admin:~ # discover --rack 1,ha=2
```

Warning: Before starting the discovery of the second HA-RLC, all configuration steps **must** have completed for the first HA-RLC. This means that the rack compute nodes will have booted. Overlapping individual discovery of HA-RLCs is **NOT** supported.

For additional information about installing a single HA-RLC, see the section below, "Discover replacement HA-RLC".

Monitor Rack HAE Status

After the HA-RLC installation completes, there are multiple methods of monitoring the HAE status of a rack as described in this section.

crm_mon Command Line Utility

The command line utility `crm_mon` can be used to check the HAE status of a rack, as follows:

```
admin:~ # ssh r1lead crm_mon -1
============
Last updated: Tue May 17 15:27:23 2011
Stack: openais
Current DC: r1lead1 - partition with quorum
Version: 1.1.5-5ce2879aa0d5f43d01629bc20edc6868a9352002
2 Nodes configured, 2 expected votes
7 Resources configured.
============

Online: [ r1lead1 r1lead2 ]

 blademond      (lsb:blademond):        Started r1lead1
 conserver      (lsb:conserver):        Started r1lead1
 dhcpd  (lsb:dhcpd):    Started r1lead1
```

```
 Resource Group: alias-group
     head-alias (ocf::heartbeat:IPaddr2):       Started r1lead1
     vlan1-alias        (ocf::heartbeat:IPaddr2):       Started r1lead1
     vlan2-alias        (ocf::heartbeat:IPaddr2):       Started r1lead1
 stonith-ipmi-1 (stonith:external/ipmi):       Started r1lead1
 stonith-ipmi-2 (stonith:external/ipmi):       Started r1lead2

The list and order of resources may differ with release version.

One stonith resource runs on each HA-RLC and only functions to monitor
the BMC of the peer HA-RLC.  If only a single HA-RLC is installed,
there will be no stonith-impi resources configured.
```

You can also use an abbreviated version of `crm_mon` to check HA-RLC HAE status, as follows:

```
sys-name:~ # pdsh -g leader crm_mon -s
r2lead2: Ok: 2 nodes online, 6 resources configured
r1lead2: Ok: 2 nodes online, 6 resources configured
r1lead1: Ok: 2 nodes online, 6 resources configured
r2lead1: Ok: 2 nodes online, 6 resources configured
```

The number of resources configured for HA-RLC may differ with release version.

crm_gui Utility

The `crm_gui` runs from a login to a HA-RLC. This tool provides significant functionality for working with HAE.

Warning: Making changes to the HAE configuration is not supported.

From the `r1lead1` leader node, launch `crm_gui`, as follows:

```
r1lead1# crm_gui &
```

The CRM GUI appears, as shown in Figure B-5 on page 366.

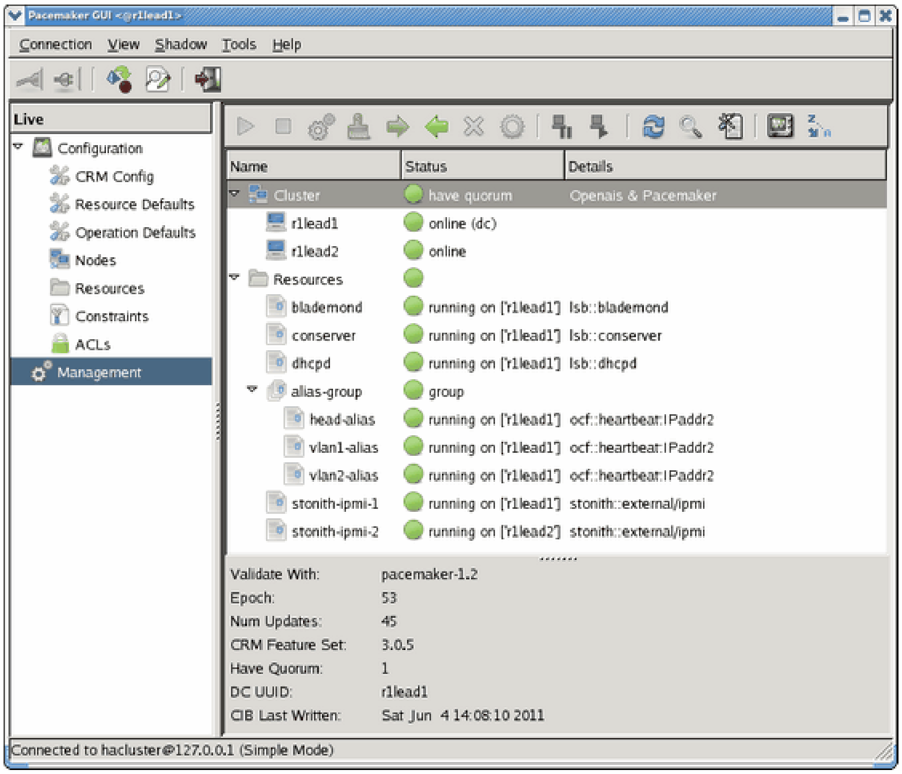

Figure B-5 CRM Cluster Status GUI

Hawk Browser Interface

The Hawk browser interface provides status and some basic commands for an individual HAE instance.

Start the Firefox browser on the admin node and point to the Hawk server (port 7630) at a HA-RLC physical interface. If pointed at a HA-RLC alias, the connection is dropped when the resources are migrated.

For example, to connect to the rack 1 HAE instance via the `r1lead1` leader node, enter the following in your Firefox browser: https://r1lead1:7630/ .

Login as **hacluster**, using the default password **sgisgi**. Confirm the security exception. The default user **hacluster** is configured for HA-RLC in /etc/opt/sgi/conf.d/80-corosync.

By left-clicking on the wrench icon of a resource, the commands: **Start**, **Stop**, **Move**, **Drop Relocation Rule** and **Clean Up** run on a resource, as shown in Figure B-6 on page 367.

By left-clicking on the wrench icon of a HA-RLC, **Online**, **Offline**, and **Fence Node** functions are available, as shown in Figure B-11 on page 373. Fencing is not implemented.

For more information, see "Starting the HA Web Console and Logging In" in the *SUSE Linux Enterprise High Availability Extension High Availability Guide.*

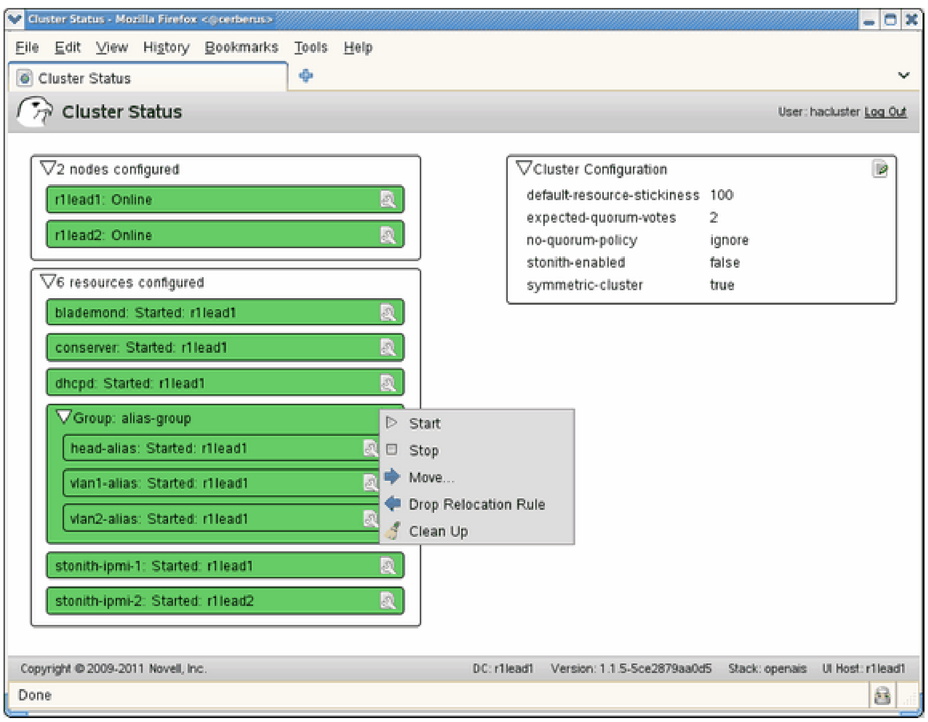

Figure B-6 Hawk Interface with Resource Control Pop-up Menu

Administrative Resource Migration

Migrating resources is very useful for testing and is also useful for moving resources from a HA-RLC if maintenance is needed on that HA-RLC.

Three tools that can be used to migrate resources are , as follows:

- `crm` command line interface
- `crm_gui` run on a HA-RLC
- The Hawk browser interface to the HAE running on the HA-RLC

Note: In response to an administrative migration using `crm` or using `crm_gui`, HAE creates a location constraint that forces the resources to migrate. This location constraint **must** be deleted for it to be possible for the resources to migrate back.

`crm` - Resource Migration

Login to a HA-RLC via a physical interface, or from the admin node send commands to the rack HA alias.

To migrate all resources from `rllead1` to `r2lead2`, perform the following:

```
admin:~ # ssh r1lead crm resource migrate alias-group r1lead2
admin:~ # ssh r1lead crm_mon -1
admin:~ # ssh r1lead crm configure show | grep prefer
admin:~ # ssh r1lead crm configure delete cli-prefer-alias-group
admin:~ # ssh r1lead crm configure show | grep prefer
admin:~ # ssh r1lead crm_mon -1
```

The `cli-prefer-alias-group` must be deleted to migrate the resources back.

It is possible that this constraint will also prevent failure driven resource migration.

crm_gui - Resource Migrate

For running the `crm_gui` from a HA-RLC, it is best to login via a physical interface. With the migration, a login via the rack HA alias is dropped.

For example, to login to `r1lead1` and start `crm_gui`, perform the following:

```
admin:~ # ssh r1lead1
r1lead1# crm_gui &
```

For migrating all of the resources (all that migrate), migrate the **alias-group**, as shown in Figure B-7 on page 369 and Figure B-8 on page 370.

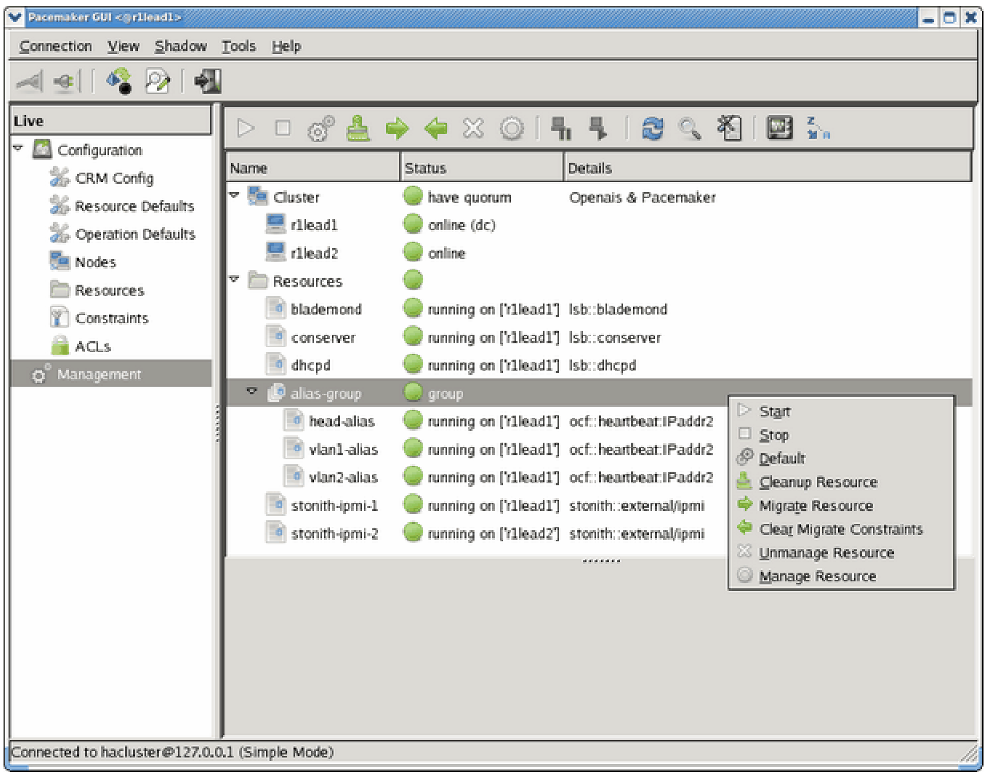

Figure B-7 `crm_gui` **Migrate `alias-group` Resource** Example

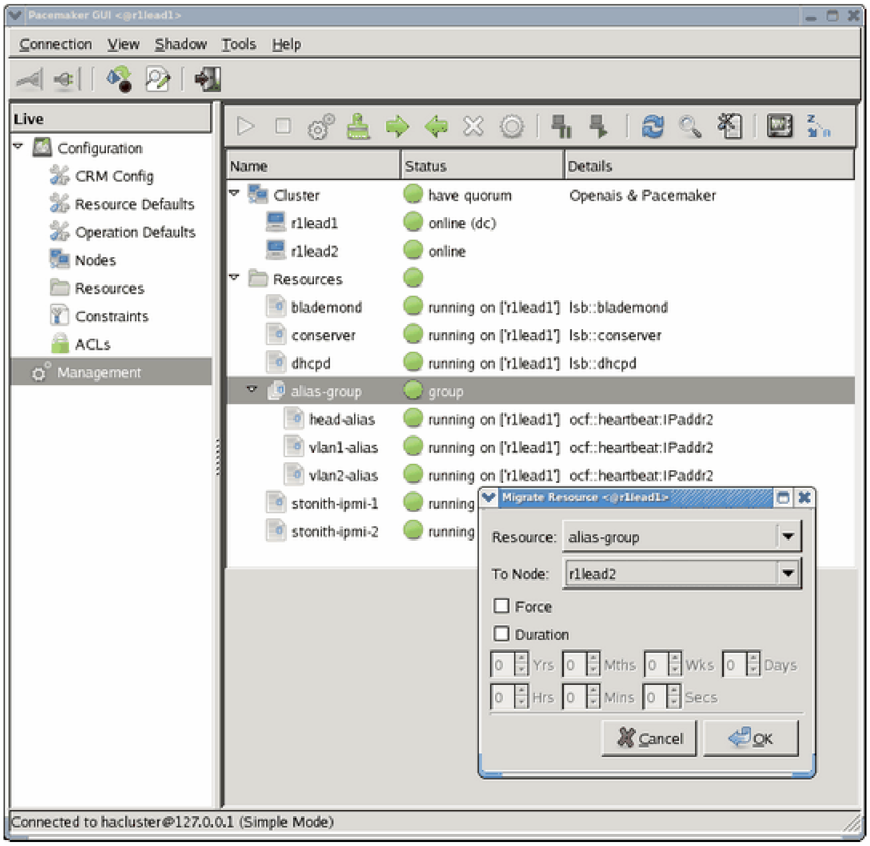

Figure B-8 `crm_gui` **Migrate `alias-group` Resource** Example

As with `crm resource migrate`, manual migration via the `crm_gui` creates a location constraint. This preferred location constraint **must** be deleted after the migration, either via the **Constraints** screen or via the button at the top of the **Management** screen, as shown in Figure B-9 on page 371.

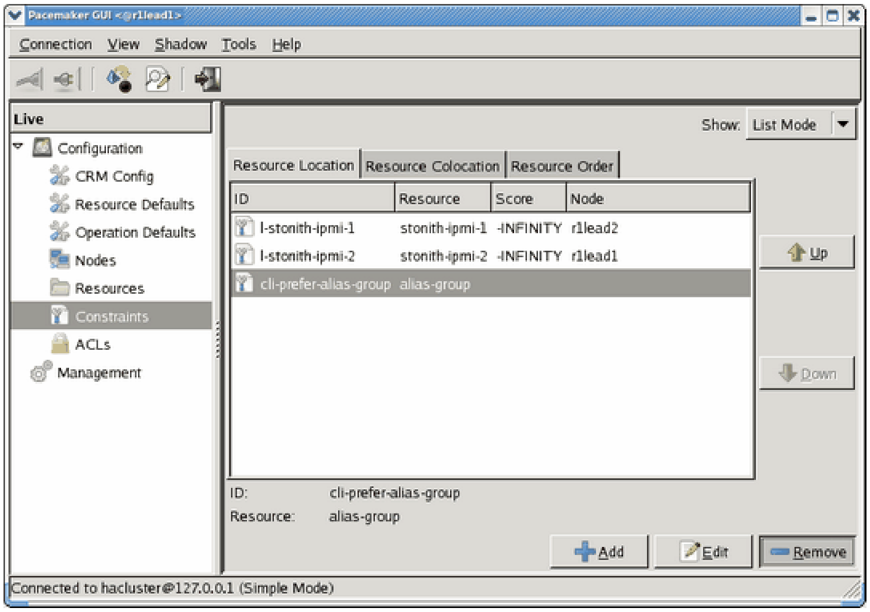

Figure B-9 CRM GUI **Configuration Constraints** Selection Example

Hawk Browser Interface — Resource Migration

The Hawk browser interface provides status and some basic commands for an individual HAE instance.

Start Firefox on the admin node and point to the Hawk server (port 7630) at a HA-RLC physical interface. If pointed at a HA-RLC alias, the connection is dropped when the resources are migrated.

For example, to connect to the rack 1 HAE instance via `r1lead1`, enter the following: `https://r1lead1:7630/`.

Login as **`hacluster`** and with the default password **`sgisgi`**. Confirm the security exception. The default **hacluster** user is configured for HA-RLC in `/etc/opt/sgi/conf.d/80-corosync`.

By left-clicking on the wrench icon of a resource, **Start**, **Stop**, **Move**, **Drop Relocation Rule**, and **Clean Up** run on a resource, as shown in Figure B-10 on page 372.

By clicking on the wrench icon of a HA-RLC, **Online**, **Offline**, and **Fence Node** functions are available, as shown in Figure B-11 on page 373. Fencing is not implemented.

For more information, see "Starting the HA Web Console and Logging In" in the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

This section describes two methods of resource migration using the Hawk browser interface.

Migrate Using Hawk - Method 1

On the Hawk interface, left-click on the **alias-group** wrench icon and select **move**, and then select the leader node to which to migrate the resources, as shown in Figure B-10 on page 372.

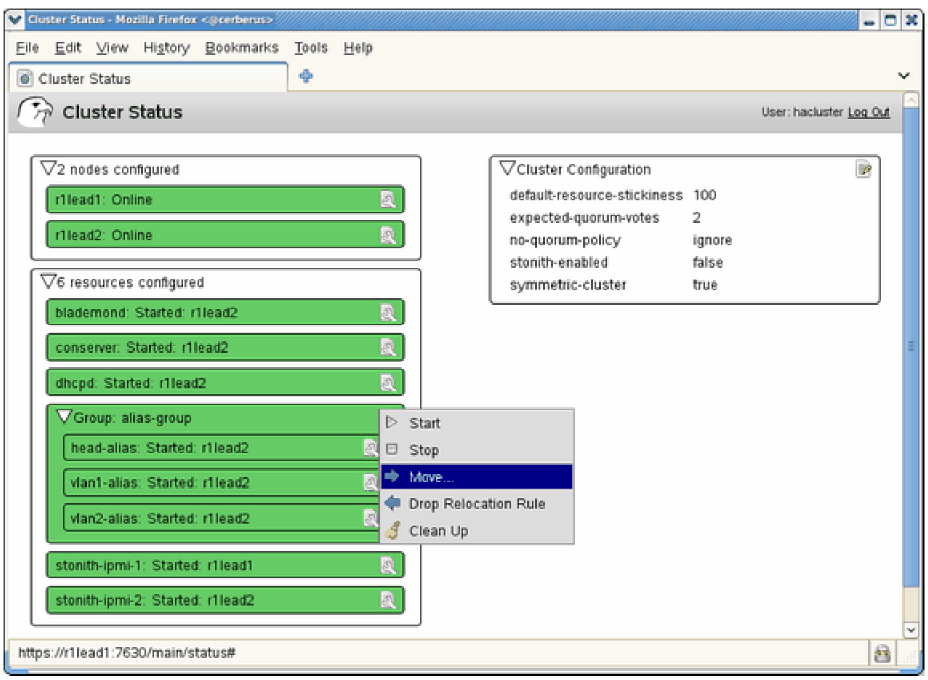

Figure B-10 Using Hawk Browser Interface to Migrate Resources (Method 1 of 2)

Monitor that the resources do migrate. If they do not migrate, it is probably necessary to clear the migration constraint.

Clear the migration constraint with **Drop Relocation Rule**, which is accessible with a left-click on the **alias-group** wrench.

Migrate Using Hawk - Method 2

The Hawk interface commands **Standby** and **Online** can be used to migrate resources between HA-RLC, as shown in Figure B-11 on page 373.

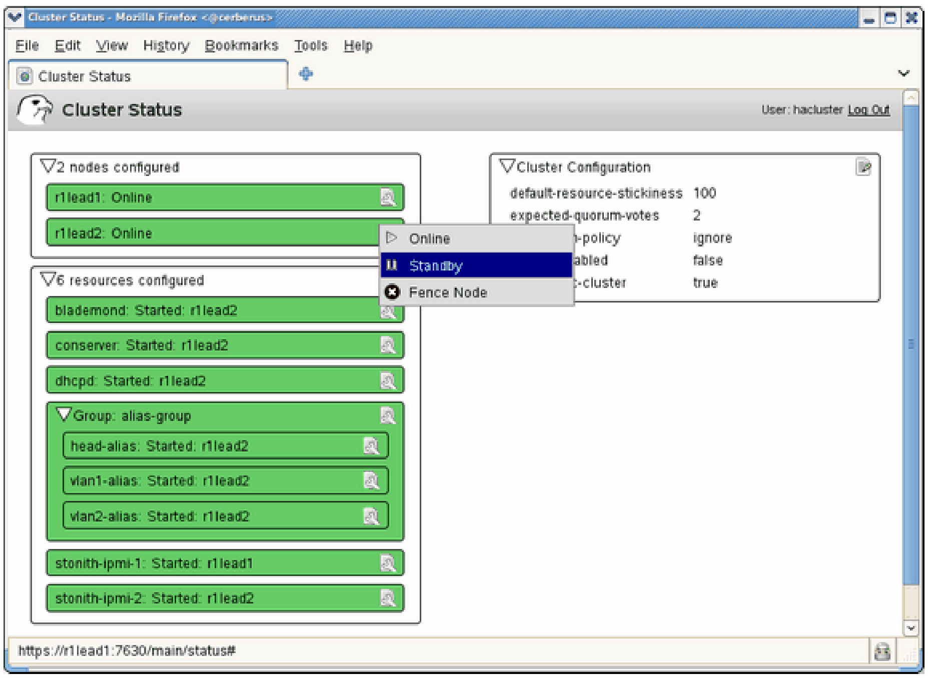

Figure B-11 Using Hawk Browser Interface to Migrate Resources (Method 2 of 2)

With both HA-RLCs online, and all of the resources started, select the wrench icon associated with the HA-RLC which is currently hosting the resources. Select **Standby** from the pop-up menu. Placing this HA-RLC in **standby** mode causes the resources to be migrated to the **online** HA-RLC.

This action does not create a migration constraint.

To migrate the resources back to the first HA-RLC, select **online** from the pop-up menu associated with the first HA-RLC (which is in standby mode) to bring it back **online**.

Then force the HA-RLC hosting the resources into **standby** mode.

After the resources have migrated, bring the HA-RLC in **standby** mode back **online**.

Replace a HA-RLC

It is possible to take a HA-RLC offline, remove it from the cluster, and to then discover a replacement HA-RLC.

In this context "take offline" can include failure of the HA-RLC.

Remove an HA-RLC from a Cluster

There are two commands associated with removing a HA-RLC from a cluster:

- `discover --delrack`

  The purpose of the `discover --delrack` command is to keep place holders in the database for the replacement HA-RLC, so that the original IP addresses are reused when the replacement HA-RLC is discovered.

  In this example, the command sets the database state of HA-RLC `r1lead1` to `NOT_EXIST` and does some cleanup.

  To delete the first HA-RLC of rack 1, perform the following:

  ```
  admin:~ # discover --delrack 1,ha=1
  ```

  Note: If the remaining HA-RLC (in this example, that would be `r1lead2`) is also deleted, that is, with `ha=2`, then the state of `r1lead2` and rack compute nodes are also set to `NOT_EXIST` and discovery of the whole rack is necessary.

- To purge the node from the database, perform the following:

  ```
  admin:~ # cadmin --db_purge --node r1lead1
  ```

  In this example, the command purges all information referring to `r1lead1` from the cluster and the cluster database. A replacement HA-RLC will be assigned the next (higher) addresses in the database tables.

Note: If the remaining HA-RLC (in this example, that would be `r1lead2`) is also purged, then the rack and compute nodes are purged from the database, and discover of the whole rack will be necessary.

Discover Replacement HA-RLC

After physically replacing the HA-RLC, the `discover` command is used to discover the single replacement HA-RLC. It is assumed that the peer HA-RLC is currently hosting the HAE resources and is fully configured.

For example, to discover the HA-RLC `r1lead1`, perform the following:

```
admin:~ # discover --rack 1,ha=1
```

The `discover` command prompts for power to be applied to the replacement HA-RLC, that is, for the HA-RLC to be plugged in.

If the BMC and IP MAC addresses for the replacement node are known, then that information could be entered into a `macfile` and the "apply power to discover" prompt can be avoided.

The discovery of the replacement HA-RLC initiates the installation of the default lead node image, the HA-RLC boots and joins the rack HAE cluster. The HAE software then synchronizes the replacement node with the existing HAE information.

The compute node images that are *in-use* on the running HA-RLC will be loaded on the replacement HA-RLC and that HA-RLC will be configured to boot compute nodes with the appropriate image.

Compute node images that are installed but not "in-use" are not installed on the replacement HA-RLC. However, it is not possible to set a compute node to use one of those images without again pushing the image to the rack.

YaST2 Navigation

The following list shows SLES YaST2 navigation key sequences:

Key	Action

`Tab`
`Alt` + `Tab`
`Esc` + `Tab`
`Shift` + `Tab`
	Moves you from label to label or from list to list.

`Ctrl` + `L`	Refreshes the screen.
`Enter`	Starts a module from a selected category, runs an action, or activates a menu item.
`Up arrow`	Changes the category. Selects the next category up.
`Down arrow`	Changes the category. Selects the next category down.
`Right arrow`	Starts a module from the selected category.
`Shift` + `right arrow` `Ctrl` + `A`	Scrolls horizontally to the right. Useful in screens if use of the `left arrow` key would otherwise change the active pane or current selection list.
`Alt` + *letter* `Esc` + *letter*	Selects the label or action that begins with the *letter* you select. Labels and selected fields in the display contain a highlighted *letter*.
Exit	Quits the YaST2 interface.

Index

T

U

V