SGI® Management Center™ (SMC) Administration Guide for Clusters

007–6358–005

The SGI Management Center software stack depends on several open source packages which require attribution. They are as follows:

c3:

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

conserver:

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

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

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

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

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

pysqlite:

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

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

LIMITED RIGHTS LEGEND

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

onventions. This document is provided with limited rights as defined in 52.227-14.

TRADEMARKS AND ATTRIBUTIONS

Altix, Performance Co-Pilot, Rackable, SGI, SGI ICE, SGI Management Center, the SGI logo, and Supportfolio are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries.

Altair is a registered trademark and PBS Professional is a trademark of Altair Engineering, Inc. Intel, Xeon, and Itanium are trademarks or registered trademarks of Intel Corporation. CentOS Marks are trademarks and Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc., in the United States and other countries. InfiniBand is a trademark of the InfiniBand Trade Association. Linux is a registered trademark of Linus Torvalds. LSI Logic and MegaRAID are registered trademarks of the LSI Logic Corporation. InfiniScale is a registered trademark of Mellanox Technologies. SLES, SUSE, and YAST are registered trademarks of SUSE LLC in the United States and other countries.

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

New Features

This revision describes the SGI® Management Center™ (SMC) 3.1.3 release, which includes support for Red Hat Enterprise Linux® 7.1 and SLES® 12.

Record of Revision

Version	Description
001	November 2014 Original publication. This revision supports the SGI Management Center 3.0 release and the SGI Foundation Software 2.11 release.
002	February 2015 Revised to incorporate miscellaneous corrections.
003	May 2015 This revision supports the SGI Management Center 3.1 release and the SGI Foundation Software 2.12 release.
004	July 2015 This revision supports the SGI Management Center 3.1.2 release and the SGI Foundation Software 2.12 release.
005	August 2015 This revision supports the SGI Management Center 3.1.3 release and the SGI Foundation Software 2.12 release.

Contents

About This Guide

This guide is a reference document for system administrators of SGI® ICE™ and SGI Rackable™ clusters. It describes how to use SGI Management Center (SMC) to perform general system maintenance operations.

Cluster Terminology

This guide uses the following terms to refer to the nodes in an SGI Rackable cluster (*flat cluster*):

- admin node
- compute node

This guide uses the following terms to refer to the nodes in an SGI ICE X cluster (hierarchial cluster):

- admin node
- rack leader controller (RLC) or leader node
- compute node (flat compute node)
- *SGI ICE compute node* or compute blade

See *SGI Management Center (SMC) Installation and Configuration Guide for Clusters* for a detailed description of the types of clusters and node types.

Note: Unless otherwise noted, you can assume that descriptions pertaining to RHEL platforms also apply to CentOS platforms.

Related Publications

The SGI Foundation Software release notes and the SGI Performance Suite release notes contain information about the specific software packages provided in those products. The release notes also list SGI publications that provide information about the products. The release notes are available in the following locations:

- Online at Supportfolio. After you log into Supportfolio, you can access the release notes. The SGI Foundation Software release notes are posted to the following website:

  https://support.sgi.com/content_request/194480/index.html

  The SGI Performance Suite release notes are posted to the following website:

  https://support.sgi.com/content_request/786853/index.html

  Note: You must sign into Supportfolio, at https://support.sgi.com/login, in order for the preceding links to work.

- On the product media. The release notes reside in a text file in the `/docs` directory on the product media. For example, `/docs/SGI-MPI-1.`*x*`-readme.txt`.
- On the system. After installation, the release notes and other product documentation reside in the `/usr/share/doc/packages/`*product* directory.

All SGI publications are available on the Technical Publications Library at the following website:

http://docs.sgi.com

The following documentation might be useful to you:

- *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*, publication 007–6359–*xxx*

  Describes how to install and configure an SGI ICE cluster or an SGI Rackable cluster.

- *SGI Foundation Software (SFS) User Guide*, publication 007–6410–*xxx*

  Describes a variety of tools to tune and monitor your SGI computer system. These tools also facilitate efficient communication with SGI technical support personnel.

- *Message Passing Toolkit (MPT) User's Guide*

  Describes industry-standard message passing protocol optimized for SGI computers. This manual describes how to tune the run-time environment to improve the performance of an MPI message passing application on SGI computers. None of these ways involve application code changes.

- *MPInside Reference Guide*

Documents the SGI MPInside MPI profiling tool.

- SGI hardware documentation.

  SGI creates hardware manuals that are specific to each product line. The hardware documentation typically includes a system architecture overview and describes the major components. It also provides the standard procedures for powering on and powering off the system, basic troubleshooting information, and important safety and regulatory specifications.

  The following procedure explains how to retrieve a list of hardware manuals for your system.

Procedure 0-1 To retrieve hardware documentation

1. Type the following URL into the address bar of your browser:

   `docs.sgi.com`

2. In the search box on the Techpubs Library, narrow your search as follows:

   – In the **search** field, type the model of your SGI system.

     For example, type one of the following: **"UV 2000"**, **"ICE X"**, **Rackable**.

     Remember to enclose hardware model names in quotation marks (" ") if the hardware model name includes a space character.

   – Check **Search only titles**.

   – Check **Show only 1 hit/book**.

   – Click **search**.

- In addition to SGI documentation, the following documentation from other sources might interest you:

  – SUSE documentation for SLES 12 and for SLES 11 SP3

  – Red Hat documentation for Red Hat Linux Enterprise Server 7.1 (RHEL 7.1), RHEL 6.6, and CentOS 6.6

  – 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.
- 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 either 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:

  http://www.sgi.com/support/supportcenters.html

SGI values your comments and will respond to them promptly.

Configuring Optional Compute Node Features

This chapter contains the following topics:

Configuring a Compute Node as a Network Address Translation (NAT) Gateway

The procedure in this topic explains how to configure network address translation (NAT) for your cluster. The procedure configures NAT on a compute node. There is no need to configure NAT on SGI ICE compute nodes or any of the other node types.

Complete the procedure in this topic if you want to run a network file system (NFS) client or a network information service (NIS) client (also known as a *yp client*) on the SGI ICE compute nodes. The procedure in this topic observes the following guidelines:

- NAT is configured on node `service0`.
- `eth0` on `service0` always connects to the SGI management network.
- The house network could be `eth1` or another network, depending on your node configuration, but not `eth0`.

If you have trouble with the following procedure, see the following information in the troubleshooting chapter:

"Troubleshooting a Network Address Translation (NAT) Configuration" on page 7

The following procedure describes how to enable NAT on a compute node.

Procedure 1-1 To enable NAT on a compute node

1. Through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to retrieve information about the compute nodes in your system:

   ```
   # cnodes --compute
   service0
   service1
   service2
   ```

   This command shows the compute nodes that are available. You can configure NAT on any of these nodes. This example in this procedure uses `service0`.
3. Through an `ssh` connection, log into one of the compute nodes as the root user.

   For example:

   ```
   # ssh service0
   ```
4. Type the following command to change to the directory where the NAT configuration script resides:

   ```
   # cd /opt/sgi/docs/setting-up-NAT
   ```
5. Type the following command to enable execute permission on the file named README:

   ```
   # chmod 755 README
   ```
6. Type the following command to run the README file:

   ```
   # ./README
   net.ipv4.ip_forward = 1
   + iptables-restore
   + modprobe ip_conntrack_tftp
   + modprobe ip_nat_tftp
   ```

Output similar to the preceding appears on your screen when the README script runs correctly.

7. Retrieve the IP address of InfiniBand network card ib0.

   Use one of the following commands:

   - On RHEL 7 or SLES 12 platforms, type the following command:

     ```
     service0:~ # ip addr show ib0
     10: ib0:  <BROADCAST,MULTICAST,UP,LOWER_UP>
     mtu 65520 qdisc pfifo_fast state UP qlen 256 link/infiniband
     80:00:04:04:fe:c0:00:00:00:00:00:00:08:f1:04:03:97:86:c5 brd
     00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
     inet 10.148.0.2/16 brd 10.148.255.255 scope global ib0 inet6
     fe80::208:f104:397:86c5/64 scope link
     valid_lft forever preferred_lft forever
     ```

   - On RHEL 6 or SLES 11 platforms, type the following command:

```
service0:~ # ifconfig ib0
ib0       Link encap:InfiniBand  HWaddr 80:00:04:04:FE:C0:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:10.148.0.2  Bcast:10.148.255.255  Mask:255.255.0.0
          inet6 addr: fe80::202:c902:26:403d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:65520  Metric:1
          RX packets:1973872 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1612831 errors:0 dropped:97 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:232879516 (222.0 Mb)  TX bytes:582347073 (555.3 Mb)
```

Note: In the preceding output, lines have been wrapped for inclusion in this documentation.

   The IP address of ib0 is 10.148.0.2.

8. Type logout to log out from the compute node and return to the admin node.

9. Type the following command to change to the directory where the SGI ICE compute node update script resides:

   ```
   # cd /opt/sgi/share/per-host-customization/global
   ```

The system runs these scripts at startup. The next few steps explain how to edit the `sgi-static-routes.sh` file to point to `ib0`.

10. Use a text editor to open file `sgi-static-routes.sh`.

    The next few steps in this procedure modify the file. As a precaution, you can copy the file to a backup location before you begin to edit.

11. Search for a line that begins with `echo "default`.

    This line should include the IP address of `ib0` and the literal string `ib0`. The line might be correct in the file, but if necessary, edit the line. For this example, edit the line to remove the comments characters (#) and be as follows:

    ```
    if [ -d ${imagedir}${SLES_PATH} ]; then
            echo "default 10.148.0.2 - ib0 -" >>${imagedir}${SLES_PATH}routes
    fi
    if [ -d ${imagedir}${RHEL_PATH} ]; then
            echo "default via 10.148.0.2" >>${imagedir}${RHEL_PATH}route-ib0
    fi
    ```

    The `sgi-static-routes.sh` script customizes the network routing based upon the rack, the individual rack unit (IRU), and the slot of the compute blade. Some examples are available in the script. The next few steps boot the SGI ICE compute nodes.

12. Type the following command to shut down and stop all the SGI ICE compute nodes:

    ```
    admin node:~ # cpower node halt "r*i*n*"
    ```

13. Type the following command to propagate the changes:

    ```
    admin node:~ # cimage --push-rack
    ```

14. Type the following command to power-up all the SGI ICE compute nodes:

    ```
    admin node:~ # cpower node on "r*i*n*"
    ```

    When you power-up the computes nodes, the `sgi-static-routes` script runs and updates the default route information configured for NAT.

15. Type the following command to retrieve a list of the rack leader controller (RLC) nodes:

    ```
    admin node:~ # cnodes --leader
    ```

16. Through an ssh(1) connection, log into one of the leader nodes.

    For example:

    ```
    admin node:~ # ssh r1lead
    ```

17. Type the following command to retrieve a list of the SGI ICE compute nodes attached to this RLC:

    ```
    r1lead:~ # cnodes --ice-compute
    ```

18. Through an ssh(1) connection, log into one of the SGI ICE compute nodes.

    For example:

    ```
    r1lead:~ # ssh r1i1n0
    ```

Note: To log into a SGI ICE compute node, always log into the rack leader controller (RLC) first. You cannot log into a SGI ICE compute node directly from the admin node or from a compute node.

---

19. Type the ping(8) command, in the following format, to verify that the SGI ICE compute node can access the compute node through the InfiniBand ib0 subnetwork:

    ping -c 1 *ib0_IP_addr*

    In the preceding format, note the following:

    - The -c 1 parameter restricts the output to one ECHO_REQUEST packet.
    - For *ib0_IP_addr*, specify the IP address of the InfiniBand ib0 subnetwork. This is the IP address you retrieved in the following step:

      Procedure 1-1, step 7 on page 3

    For example:

    ```
    r1i3n0:~ # ping -c 1 10.148.0.2
    PING 10.148.0.2 (10.148.0.2) 56(84) bytes of data.
    64 bytes from 10.148.0.2: icmp_seq=1 ttl=64 time=3.90 ms

    --- 10.148.0.2 ping statistics ---
    1 packets transmitted, 1 received, 0% packet loss, time 0ms
    rtt min/avg/max/mdev = 3.904/3.904/3.904/0.000 ms
    ```

20. Verify the InfiniBand address on the SGI ICE compute node.

    Use one of the following commands:

    - On RHEL 7 or SLES 12 platforms, type the following command:

      ```
      r1i3n0:~ # ip addr show ib0
      10: ib0:  <BROADCAST,MULTICAST,UP,LOWER_UP>
      mtu 65520 qdisc pfifo_fast state UP qlen 256 link/infiniband
      80:00:04:04:fe:c0:00:00:00:00:00:00:00:08:f1:04:03:97:86:c5 brd
      00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
      inet 10.148.0.57/16 brd 10.148.255.255 scope global ib0 inet6
      fe80::208:f104:397:86c5/64 scope link
      valid_lft forever preferred_lft forever
      ```

    - On RHEL 6 or SLES 11 platforms, type the following command:

```
r1i3n0:~ # ifconfig ib0
ib0       Link encap:InfiniBand  HWaddr 80:00:04:04:FE:C0:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:10.148.0.57  Bcast:10.148.255.255  Mask:255.255.0.0
          inet6 addr: fe80::230:487a:c4e0:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:65520  Metric:1
          RX packets:125967 errors:0 dropped:0 overruns:0 frame:0
          TX packets:143324 errors:0 dropped:7 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:33073064 (31.5 Mb)  TX bytes:22602331 (21.5 Mb)
```

Note: In the preceding output, lines have been wrapped for inclusion in this documentation.

21. Type the following command to verify the default gateway to the compute node:

    ```
    r1i3n0:~ # netstat -rn
    ```

    Make sure that the default route shown in the output is to the compute node (that is, to the NAT).

22. On the compute node, type the following command(s) to verify that the compute node can communicate with the SGI ICE compute nodes:

    ```
    service0:~ # date
    Mon Dec  3 12:14:13 CST 2012
    service0:~ # date ; cexec --pipe date
    ```

```
Mon Dec  3 12:14:23 CST 2012
blades r1i3n0: Mon Dec  3 12:14:24 CST 2012
blades r1i3n1: Mon Dec  3 12:14:24 CST 2012
blades r1i3n2: Mon Dec  3 12:14:24 CST 2012
blades r1i3n3: Mon Dec  3 12:14:24 CST 2012
service0:~ # cexec --pipe date
blades r1i3n0: Mon Dec  3 12:14:48 CST 2012
blades r1i3n1: Mon Dec  3 12:14:48 CST 2012
blades r1i3n2: Mon Dec  3 12:14:48 CST 2012
blades r1i3n3: Mon Dec  3 12:14:48 CST 2012
```

Troubleshooting a Network Address Translation (NAT) Configuration

The first steps are to determine that the compute 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 following files on the admin node:

- On RHEL 7 and SLES 12 platforms, look in the following file:

  `/etc/named.conf`

- On RHEL 6 and SLES 11 platforms, look in the following file:

  `/etc/resolv.conf`

Additionally, the default gateway addresses for the compute 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 compute node to the selected IP address gets responses, network monitoring tools such as `tcpdump`(1) should be used. On the compute node, monitor the `eth1` interface and simultaneously in a separate session monitor the `ib[01]` interface. You should specify monitoring that is specific enough

to not have additional noise then attempt to execute a `ping` command from the SGI ICE compute node.

Example 1-1 `tcpdump` Command Examples

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

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

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

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

Configuring a File System on a Compute Node for Use with a Network File System (NFS) Server

The procedure in this topic explains how to configure a file system on a compute node. This procedure assumes the SUSE Linux Enterprise Server (SLES) platform. If you use the Red Hat Enterprise Linux (RHEL) platform, use your operating system documentation to complete this procedure.

Procedure 1-2 To configure an NFS home server on a compute node

1. Through an `ssh` connection, log into the admin node as the root user, and then log into the compute node as the root user.

   The example in this procedure assumes that you want to configure `service0` as an NFS server.

For example:

```
# ssh mycluster
root@admin node # ssh service1
root@service1 #
```

2. Type the following command to retrieve the name of the root device:

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

This command shows that the root device is named `sda`.

Make sure you know which device is your root device. Do not take any actions that can repartition or otherwise destroy your root device.

3. Retrieve the names of the disk partitions, and use `by-id` notation.

The steps in this procedure avoid using `/dev/sd`*X* notation in device names because device names in that style are not persistent. Those device names can change as you adjust disks and RAID volumes in your system. For example, you may assume that `/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, the instructions in this procedure use `by-id` notation.

Your goal is to retrieve the names of the non-root disk partitions. You can choose one of these partitions to host the NFS services. The following example shows the command to use and example output:

```
# ls -l /dev/disk/by-id
total 0
lrwxrwxrwx 1 root root  9 2012-03-20 04:57 ata-MATSHITADVD-RAM_UJ-850S_HB08_020520 -> ../../hdb
lrwxrwxrwx 1 root root  9 2012-03-20 04:57 scsi-3600508e000000000307921086e156100 -> ../../sda
lrwxrwxrwx 1 root root 10 2012-03-20 04:57 scsi-3600508e000000000307921086e156100-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 2012-03-20 04:57 scsi-3600508e000000000307921086e156100-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 2012-03-20 04:57 scsi-3600508e000000000307921086e156100-part5 -> ../../sda5
lrwxrwxrwx 1 root root 10 2012-03-20 04:57 scsi-3600508e000000000307921086e156100-part6 -> ../../sda6
lrwxrwxrwx 1 root root  9 2012-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a -> ../../sdb
lrwxrwxrwx 1 root root 10 2012-03-20 04:57 scsi-3600508e0000000008dced2cfc3c1930a-part1 -> ../../sdb1
lrwxrwxrwx 1 root root  9 2012-03-20 09:57 usb-PepperC_Virtual_Disc_1_0e159d01a04567ab14E72156DB3AC4FA \
 -> ../../sr0
```

The preceding output shows that ID `scsi-3600508e000000000307921086e156100` is in use by your system disk. This in-use status is revealed in the symbolic link that points to `../../sda`. This is the root disk device. Do not consider this disk device for NFS use.

The other disk in the listing has ID `scsi-3600508e0000000008dced2cfc3c1930a` and is linked to `/dev/sdb`. You can configure the NFS services on this disk because it is a separate physical disk and is not `sda`, which is the root disk.

The next few steps create a filesystem on the disk.

4. Create a new `msdos` label on the disk.

   This procedure uses the `parted`(8) utility in a command-line driven manner. If you prefer, you can use `parted`(8) interactively, or you can use a different partitioning tool.

   For example, the following command creates a new label on `/dev/disk/by-id/scsi-3600508e0000000008dced2cfc3c1930a`:

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

5. Retrieve the size of the disk.

   For example, type the following command:

```
# 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.
```

6. Create a partition that spans the size of the disk.

   For example, type the following command:

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

7. Create a filesystem on the disk.

   You can choose the filesystem type.

Example 1. This example shows how to create an ext3 filesystem. The number of blocks and the bytes-per-node ratio determine the default number of inodes that the command creates, but the command accepts parameters that enable you to control the number and size of the inodes. It can take 10 minutes or more to create one 500–GB filesystem using default `mkfs.ext3` command line parameters. The following example command uses the `-N` option to reduce the number of inodes to 20 million inodes:

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

Example 2. This example shows how to create an XFS filesystem. Generally, you can create an XFS file system in less time than it takes to create an ext3 filesystem. The command is as follows:

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

8. Use a text editor to open file `/etc/fstab`.
9. Add a line at the end of file `/etc/fstab` that defines the new filesystem.

   Make sure to use the `by-id` path for the device. This `fstab` entry enables the operating system to mount the filesystem automatically the next time the system reboots.

   Example 1. The following line defines the ext3 filesystem that was created in Procedure 1-2, step 7 on page 10:

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

   Example 2. The following line defines the XFS filesystem that was created in Procedure 1-2, step 7 on page 10:

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

10. Save and close the `/etc/fstab` file.
11. Type the following command to mount the new filesystem:

    ```
    # mount -a
    ```

12. Proceed to the following:

    "Configuring a Compute Node as an NFS Server" on page 12

Configuring a Compute Node as an NFS Server

The following procedure explains how to configure a compute node as an NFS server.

Procedure 1-3 To configure an NFS server on a compute node

1. Through an `ssh` connection, log into the admin node as the root user.
2. Through an `ssh` connection, log into one of the compute nodes as the root user.

   For example:

   ```
   admin node:~ # ssh service0
   ```

3. Determine whether the `nfsserver` service is enabled.

   On RHEL 7 platforms, type the following command:

   ```
   service0:~ # systemctl is-enabled nfs-server
   disabled
   ```

   On SLES 12, RHEL 6, and SLES 11 platforms, type the following command:

   ```
   service0:~ # chkconfig --list | grep nfs
   nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
   nfsserver               0:off  1:off  2:off  3:off  4:off  5:off  6:off
   ```

   The output from the preceding commands shows that the NFS server is not enabled on `service0`.

4. (Conditional) Turn on the `nfsserver` service.

   Complete this step if the `nfsserver` service is not enabled at this time.

   On RHEL 7 platforms, type the following command:

   ```
   service0: # systemctl enable nfs-server
    ln -s '/usr/lib/systemd/system/nfs-server.service' \
   '/etc/systemd/system/multi-user.target.wants/nfs-server.service'
   ```

   On SLES 12, RHEL 6, and SLES 11 platforms, type the following command:

   ```
   service0:~ # chkconfig nfsserver on
   insserv: Service dbus is missed in the runlevels 4 to use service openibd
   ```

5. Type the following command to retrieve the list of file systems that NFS can export:

   ```
   service0:~ # cat /etc/exports
   # See the exports(5) manpage for a description of the syntax of this file.
   # This file contains a list of all directories that are to be exported to
   # other computers via NFS (Network File System).
   # This file used by rpc.nfsd and rpc.mountd. See their manpages for details
   # on how make changes in this file effective.
   /home *(rw,sync,no_subtree_check)
   ```

6. Type the following command to create a directory:

   ```
   service0:~ # mkdir /home
   ```

   This step creates an example directory. Alternatively, you could specify an entire file system, rather than the directory `/home`. This could be the file system that you created in the following procedure:

   "Configuring a File System on a Compute Node for Use with a Network File System (NFS) Server" on page 8

7. (Conditional) Verify that appropriate services are running and start services as needed.

   Complete this step on RHEL 7 platforms and SLES 12 platforms.

   Use the following commands:

   - On RHEL 7 platforms, type the following commands to make sure that the `rpcbind` and `nfs-lock` services are running:

```
service0 ~# systemctl status nfs-lock
rpc-statd.service - NFS status monitor for NFSv2/3 locking.
Loaded: loaded (/usr/lib/systemd/system/rpc-statd.service; static)
Active: active (running) since Wed 2015-07-22 14:21:41 CDT; 4min 51s ago
Process: 32530 ExecStart=/usr/sbin/rpc.statd --no-notify $STATDARGS (code=exited, status=0/SUCCESS)
 Main PID: 32538 (rpc.statd)
CGroup: /system.slice/rpc-statd.service
+-32538 /usr/sbin/rpc.statd --no-notify
service0 ~# systemctl status rpcbind
rpcbind.service - RPC bind service
Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; static)
Active: active (running) since Wed 2015-07-22 14:20:41 CDT; 6min ago
```

```
Process: 32544 ExecStart=/sbin/rpcbind -w ${RPCBIND_ARGS} (code=exited, status=0/SUCCESS)
 Main PID: 32545 (rpcbind)
CGroup: /system.slice/rpcbind.service
+-32545 /sbin/rpcbind -w
```

The preceding output shows the services running. If the services are not running, type the following command to start these services:

```
service0 ~# systemctrl start rpcbind; systemctrl start nfs-lock
```

- On SLES 12 platforms, type the following command to make sure that the `rpcbind` service is running:

```
service0:~ # systemctl status rpcbind
rpcbind.service - RPC Bind
Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; disabled)
Active: active (running) since Wed 2015-07-22 22:04:45 UTC; 3min 4s ago
Docs: man:rpcbind(8)
 Main PID: 13854 (rpcbind)
CGroup: /system.slice/rpcbind.service
+-13854 /sbin/rpcbind -w -f
Jul 22 22:04:45 service0 systemd[1]: Starting RPC Bind...
```

The preceding output shows the services running. If the services are not running, type the following command to start these services:

```
service0 ~# systemctrl start rpcbind
```

8. Start the NFS server.

   Type one of the following commands to start the server:

   - On RHEL 7 platforms, type the following command:

     ```
     service0:~ # systemctrl start nfs
     ```

   - On SLES 12 platforms, type the following commands:

```
service0:~ # systemctl start nfsserver
service0:~ # systemctl status nfsserver
nfsserver.service - LSB: Start the kernel based NFS daemon

Loaded: loaded (/etc/init.d/nfsserver)
Comments from page 36 continued on next page
```

```
Active: active (running) since Wed 2015-07-22 22:05:18 UTC; 1s ago
Process: 13861 ExecStart=/etc/init.d/nfsserver start (code=exited, status=0/SUCCESS)
CGroup: /system.slice/nfsserver.service
+-13839 /usr/sbin/rpc.idmapd -p /var/lib/nfs/rpc_pipefs
+-13843 /usr/sbin/rpc.mountd
+-13883 /usr/sbin/rpc.statd --no-notify

Jul 22 22:05:17 service0 startproc[13881]:
startproc: Empty pid file /var/run/rpc.statd.pid for /usr/sbin/rpc.statd
Jul 22 22:05:17 service0 rpc.statd[13883]:
Version 1.3.0 starting Jul 22 22:05:17 service0 rpc.statd[13883]:
Flags: TI-RPC
Jul 22 22:05:18 service0 nfsserver[13861]: Starting kernel based NFS server:
idmapd mountd statd nfsd sm-notify..done
```

The preceding output is wrapped for inclusion in this documentation.

- On RHEL 6 platforms and SLES 11 platforms, type the following command:

  ```
  service0:~ # /etc/init.d/nfsserver start
  Starting kernel based NFS server: idmapd mountd statd nfsd sm-notify done
  ```

9. Type the `exportfs -av` command to export the test NFS directory, `/home`.

   For example:

   ```
   service0:~ # exportfs -av
   exporting *:/home
   ```

10. Type the following command to create a file named `testfile` in the `/home` directory and to write `test` to `testfile`:

    ```
    service0:~ # echo "test" >/home/testfile
    ```

11. Type the following command to make sure that file `testfile` was created correctly:

    ```
    service0:~ # cat /home/testfile
    test
    ```

12. Use one of the following commands to retrieve the IP address of `ib0` on the compute node:

- On RHEL 7 or SLES 12 platforms, type the following command:

```
service0:~ # ip addr show ib0
10: ib0:  <BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 65520 qdisc pfifo_fast state UP qlen 256 link/infiniband
80:00:04:04:fe:c0:00:00:00:00:00:00:08:f1:04:03:97:86:c5 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
inet 10.148.0.2/16 brd 10.148.255.255 scope global ib0 inet6
fe80::208:f104:397:86c5/64 scope link
valid_lft forever preferred_lft forever
```

- On RHEL 6 or SLES 11 platforms, type the following command:

```
service0:~ # ifconfig ib0
ib0       Link encap:InfiniBand  HWaddr 80:00:04:04:FE:C0:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:10.148.0.2  Bcast:10.148.255.255  Mask:255.255.0.0
          inet6 addr: fe80::202:c902:26:403d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:65520  Metric:1
          RX packets:1973872 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1612831 errors:0 dropped:97 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:232879516 (222.0 Mb)  TX bytes:582347073 (555.3 Mb)
```

Note: In the preceding output, lines have been wrapped for inclusion in this documentation.

In this example, the IP address is `10.148.0.2`. You use this address in a later step.

13. Through an `ssh`(1) connection, log into one of the leader nodes.

    If necessary, type a `cnodes --leader` command to retrieve the ID of one of the system's RLCs.

    For example:

    ```
    service0:~ # ssh r1lead
    ```

14. Through an `ssh`(1) connection, log into one of the SGI ICE compute nodes.

    For example:

    ```
    r1lead:~ # ssh r1i3n0
    ```

15. Use the showmount command, in the following format, to retrieve mount information and display the NFS server's file system export list:

    showmount -e *ib0_IP_service0*

    For *ib0_IP_service0*, specify the IP address of ib0 on compute node 0.

    For example:

    ```
    r1i3n0:~ # showmount -e 10.148.0.2
    Export list for 10.148.0.2:
    /home *
    ```

16. Type the following command to create the mount point:

    ```
    r1i3n0:~ # mkdir /tmp/mnt
    ```

17. Use the mount command, in the following format, to mount the file system on SGI ICE compute node r1i3n0:

    mount -t nfs *ib0_IP_service0*:/home /tmp/mnt

    For *ib0_IP_service0*, specify the host name, fully qualified domain name (FQDN), or IP address of the InfiniBand ib0 subnetwork.

    For example:

    ```
    r1i3n0:~ # mount -t nfs 10.148.0.2:/home /tmp/mnt
    ```

18. Type the following command to change to the mount point of the NFS directory:

    ```
    # cd /tmp/mnt
    ```

19. Type the following command to display mount information:

    ```
    r1i3n0:/tmp/mnt # mount | grep 10.148.0.
    10.148.0.2:/home on /tmp/mnt type nfs (rw,addr=10.148.0.2)
    ```

20. Type the following command to make sure you can access the test file on service0 from the SGI ICE compute node:

    ```
    r1i3n0:/tmp/mnt # cat /tmp/mnt/testfile
    test
    ```

21. Type logout, to log out from the SGI ICE compute node and return to the compute node.

22. Type `logout` to log out from the compute node and return to the RLC node.
23. Type `logout` to log out from the RLC node and return to the admin node.
24. Use the `cd`(1) command to change to the following directory:

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

    The following steps explain how to add the new file system to the `sgi-fstab.sh` file and ensure that the new file system mounts.
25. Use a text editor to open the following file on the admin node:

    ```
    sgi-fstab.sh
    ```
26. Within file `sgi-fstab.sh`, add a line for file system's mount point, and then save and close the file.
27. Type the following command to shut down and stop all the SGI ICE compute nodes:

    ```
    admin node:~ # cpower node halt "r*i*n*"
    ```
28. Type the following command to propagate the changes:

    ```
    admin node:~ # cimage --push-rack
    ```
29. Type the following command to power-up all the SGI ICE compute nodes:

    ```
    admin node:~ # cpower node on "r*i*n*"
    ```

    When you power-up the SGI ICE compute nodes, the file system mounts on all SGI ICE compute nodes. If the file system does not mount, see the troubleshooting information in the following topic:

    "Troubleshooting a Network Address Translation (NAT) Configuration" on page 7

Configuring Network Information Service (NIS) Clients to the House Network's NIS Server

Perform the procedures in this topic if you want to configure your compute nodes or SGI ICE compute nodes as NIS clients to your house network's NIS server. You can perform the procedures in this topic at any time after you configure network address translation (NAT) on a compute node.

The SGI ICE compute nodes are enabled to access the house network at this point because previous procedures configured the default gateway on the SGI ICE compute nodes to a compute node and because you configured the compute node to run NAT. For information about how to configure NAT, see the following:

"Configuring a Compute Node as a Network Address Translation (NAT) Gateway" on page 1

The following procedures explain how to configure the compute nodes and the SGI ICE compute nodes as NIS clients:

- "Configuring a Compute Node as a NIS Client" on page 19
- "Configuring an SGI ICE Compute Node as a NIS Client" on page 20
- "Propagating a Node's Configuration to Another Node" on page 24

Configuring a Compute Node as a NIS Client

The following procedure explains how to configure a compute node as a NIS client.

Procedure 1-4 To configure a compute node as a NIS client

1. Through an `ssh` connection, log into the admin node as the root user.
2. Through an `ssh` connection, log into one of the compute nodes as the root user.

   For example:

   ```
   admin node:~ # ssh service0
   ```

3. Start the `ypbind`(8) service.

   Use one of the following commands:

   - On RHEL 7 and SLES 12 platforms, type the following command:

     ```
     service0:~ # systemctl enable ypbind
     ```

   - On RHEL 6 and SLES 11 platforms, type the following command:

     ```
     service0:~ # chkconfig ypbind on
     ```

4. Open file `/etc/yp.conf` in a text editor.
5. Add information about your site's house NIS server to file `/etc/yp.conf`, and then save and close the file.

For example:

```
domain duluth server 100.100.100.100
```

The preceding example specifies NIS server `100.100.100.100` in the `duluth` domain.

6. Start the NIS client on `service0`.

   Use one of the following commands:

   - On RHEL 7 and SLES 12 platforms, type the following command:

     ```
     service0:~ # systemctl start ypbind
     ```

   - On RHEL 6.x and SLES 11 SPx platforms, type the following command:

     ```
     service0:~ # /etc/init.d/ypbind start
     ```

7. Type the following command to verify that the compute node client is communicating with the NIS server:

   ```
   service0:~ # ypwhich
   ```

   The output should contain the address of the NIS server, for example `100.100.100.100`.

8. Proceed as follows:

   - If you have other compute nodes that you want to configure as NIS clients, repeat this procedure on those other compute nodes.
   - If you want to configure SGI ICE compute nodes as NIS clients, proceed to the following:

     "Configuring an SGI ICE Compute Node as a NIS Client" on page 20

Configuring an SGI ICE Compute Node as a NIS Client

The following procedures explain how to configure an SGI ICE compute node as a NIS client. There is more than one way to accomplish this task, so choose from the following procedures:

- Method 1 — If you want to log into an existing SGI ICE compute node and configure only that one SGI ICE compute node as a NIS client, perform the following procedure:

"Method 1 — Configuring an Individual SGI ICE Compute Node as a NIS Client" on page 21

- Method 2 — If you want to edit the master SGI ICE compute node image on the admin node, you can push the resulting master SGI ICE compute node image to any number of SGI ICE compute nodes. Use this procedure if you want all the SGI ICE compute nodes to be configured as NIS clients. Perform the following procedure:

  "Method 2 — Configuring the Master SGI ICE Compute Node Image as a NIS Client" on page 22

- Method 3 — If you want to propagate one node's image to another node, you can change the start-up scripts that run when you boot the system. This method assumes that you used one of the previous methods (Method 1 or Method 2) to configure an initial node and that you want to copy the initial node's configuration to another node. You can use this method to update the image on any kind of node. This method clones an image from one node to another node. Perform the following procedure:

  "Propagating a Node's Configuration to Another Node" on page 24

Method 1 — Configuring an Individual SGI ICE Compute Node as a NIS Client

The following procedure configures an individual SGI ICE compute node as a NIS client.

Procedure 1-5 To log into an SGI ICE compute node and configure that compute node as a NIS client

1. Through an `ssh` connection, log into the admin node as the root user.
2. Through an `ssh`(1) connection, log into one of the rack leader controllers (RLCs).

   If necessary, type a `cnodes --leader` command to retrieve the ID of one of the RLCs.

   For example:

   ```
   admin node:~ # ssh r1lead
   ```

3. Through an `ssh`(1) connection, log into one of the SGI ICE compute nodes.

   If necessary, type a `cnodes --ice-compute` command to retrieve the ID of one of the SGI ICE compute nodes.

For example:

```
r1lead:~ # ssh r1i3n0
```

4. Start the ypbind(8) service.

   Use one of the following commands:

   - On RHEL 7 and SLES 12 platforms, type the following command:

     ```
     r1i3n0:~ # systemctl enable ypbind
     ```

   - On RHEL 6 and SLES 11 platforms, type the following command:

     ```
     r1i3n0:~ # chkconfig ypbind on
     ```

5. Open file /etc/yp.conf in a text editor.

6. Add information about your site's house NIS server to file /etc/yp.conf, and then save and close the file.

   For example:

   ```
   domain duluth server 100.100.100.100
   ```

   The preceding example specifies NIS server 100.100.100.100 in the duluth domain.

7. Type the following command to verify that the compute node client is communicating with the NIS server:

   ```
   # ypwhich
   100.100.100.100
   ```

   The output contains the IP address of the NIS server.

8. Type the logout command until you have returned to the admin node.

Method 2 — Configuring the Master SGI ICE Compute Node Image as a NIS Client

The following procedure configures the master SGI ICE compute node image on the admin node as a NIS client. You can propagate this image to other SGI ICE compute nodes after you complete the following procedure.

Procedure 1-6 To log into the admin node and edit the master SGI ICE compute node image

1. Through an ssh connection, log into the admin node as the root user.

2. Type the following command to locate the SGI ICE compute node images:

```
admin node:~ # cinstallman --show-images
Image Name               BT VCS Compat_Distro
ice-sles11sp3            1  1   sles11
        3.0.76-0.11-default
sles11sp3                0  1   sles11
        3.0.76-0.11-default
lead-sles11sp3           0  1   sles11
        3.0.76-0.11-default
```

3. In a text editor, open the `yp.conf` file for the SGI ICE compute nodes.

   For example:

```
admin node:~ # vi /var/lib/systemimager/images/ice-sles11sp3/etc/yp.conf
```

4. Add information about your site's house NIS server to file `yp.conf`, and then save and close the file.

   For example:

```
domain duluth server 100.100.100.100
```

   The preceding example specifies NIS server `100.100.100.100` in the `duluth` domain.

5. Type the following command to power-down the SGI ICE compute nodes:

```
admin node:~ # cpower node off "r*i*n*"
```

6. Type the following command to push the SGI ICE compute node changes to the SGI ICE compute nodes on your system:

```
admin node:~ # cimage --push-rack ice-sles11sp3
```

7. Type the following command to boot the SGI ICE compute nodes:

```
admin node:~ # cpower node on "r*i*n*"
```

8. Through an `ssh`(1) connection, log into one of the rack leader controllers (RLCs).

   If necessary, type a `cnodes --leader` command to retrieve the ID of one of the RLCs.

For example:

```
admin node:~ # ssh r1lead
```

9. Through an ssh(1) connection, log into one of the SGI ICE compute nodes.

   If necessary, type a `cnodes --ice-compute` command to retrieve the ID of one of the SGI ICE compute nodes.

   For example:

   ```
   r1lead:~ # ssh r1i3n0
   ```

10. Type the following command to verify that the compute client is communicating with the NIS server:

    ```
    # ypwhich
    100.100.100.100
    ```

    The output contains the IP address of the NIS server.

11. (Optional) Troubleshoot the NIS configuration.

    Use one or more of the following commands to view or set the current root image on the admin node:

    ```
    cadmin --show-root-labels
    cadmin --show-default-root
    cadmin --show-current-root
    cadmin --set-root-label --slot 2 --label "xxxxx"
    cadmin --set-default-root --slot 2
    ```

12. Type the `logout` command until you have returned to the admin node.

Propagating a Node's Configuration to Another Node

You can copy the configuration of one node to another node. When the second node boots, the image is copied from the initial node to the secondary node. This topic explains how to clone the image from one node to another node.

For example, if you have a compute node that is configured as a NIS client, you can copy (or clone) the image from the initial node to a second node. The procedure in this topic uses NIS client configuration information as an example, but you can use this procedure to propagate other system characteristics.

The following procedure explains how to propagate changes to multiple nodes.

Procedure 1-7 To propagate NIS client configuration

1. Through an `ssh` connection, log into the admin node as the root user.
2. Use the `cd`(1) command to change to the following directory:

   `/opt/sgi/share/per-host-customization/global`
3. Use a text editor to open the configuration file that you need.
4. Add the information you need to the configuration file.

   For example, within file `sgi-fstab.sh`, add a line for file system's mount point, and then save and close the file.
5. Shut down the nodes you want to reconfigure.

   For example, type the following command to shut down and stop all the SGI ICE compute nodes:

   ```
   admin node:~ # cpower node halt "r*i*n*"
   ```
6. Type the following command to propagate the changes:

   ```
   admin node:~ # cimage --push-rack
   ```
7. Power up the nodes.

   For example, type the following command to power-up all the SGI ICE compute nodes:

   ```
   admin node:~ # cpower node on "r*i*n*"
   ```

   When you power-up the computes nodes, the system mounts the new filesystem on all the SGI ICE compute nodes, uses the NIS server specifications for all SGI ICE compute nodes, and starts the `ypbind` service.

RHEL Compute Node House Network Configuration

If you plan to put your compute 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 admin node and then again to connect to the compute node. This should redirect graphics over to your desktop.

The following are some guidelines for this process:

- On compute nodes, the cluster interface is `eth0`. Do not configure this interface because it is already configured for the cluster network.
- Do not make the public interface a `dhcp` client. This can overwrite the `/etc/resolv.conf` file.
- Do not configure name servers. The name server requests on a compute node are always directed to the rack leader controller (RLC) for resolution. If you want to resolve network addresses on your house network, start the `configure-cluster` command on the admin node, and enable **House DNS Resolvers**.
- Do not configure or change the search order. Doing so could corrupt the `/etc/resolv.conf` file that the cluster configuration tool created.
- Do not change the hostname using the RHEL or SLES tools. If you need to change the hostname, log into the admin node, and use the `cadmin` command.
- After you configure your house network interface, you can use the `ifupeth`*x* command to bring the interface up. Replace *x* with your house network interface.

  If you want this interface to come up by default when the compute node reboots, be sure `ONBOOT` is set to `yes` in `/etc/sysconfig/network-scripts/ifcfg-eth`*x*. Replace *x* with the proper value. The graphical tool allows you to adjust this setting, but the text tool does not.
- If you accidentally remove or corrupt the `resolv.conf` file, and you replace it, you may need to issue the following command to ensure that DNS queries work again:

  # **`nscd --invalidate hosts`**
- 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 systems is designed to avoid, especially if large files are stored there. Consider putting your home filesystems on a NAS file server.

  For information about how to use a NAS server for scratch storage or how to make home filesystems available on NAS, see "Configuring a File System on a Compute Node for Use with a Network File System (NFS) Server" on page 8. In

that topic's example, replace `service0-ib1` with the `ib1` InfiniBand host name for the NAS server. In addition, you need to know where the home filesystem is mounted on the NAS server in order to edit the `sgi-fstab.sh` script properly.

- For information about the NIS master configuration and centrally managed user accounts, see the following:

  "Configuring a Compute Node as a Network Information Service (NIS) Server" on page 27.

  The master server residing on the compute 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 export` flag.

Configuring a Compute Node as a Network Information Service (NIS) Server

You can enable a NIS server on one of the compute nodes on your SGI ICE X system. Make sure you consider the following when you configure NIS:

- You can configure a compute node to be a NIS master server, and you can configure the rack leader controllers (RLCs) as the NIS slave servers.

  Do not configure the admin node as the NIS master server. The admin node cannot mount all storage types. When you mount the storage on the NIS master server, you can use NIS to add accounts.

- If multiple compute nodes provide home filesystems, the NIS master server should mount all the remote home filesystems. You need to export home filesystems to the NIS master compute node with the `no_root_squash export` option. The examples in the following sections assume a single compute node with storage and assume that same node is the NIS master.

- NIS traffic goes over the Ethernet. No NIS traffic goes over the InfiniBand network.

  The SGI ICE compute node NIS traffic goes over the Ethernet, by way of using the `lead-eth` server name in the `yp.conf` file. This design feature prevents NIS traffic from affecting the InfiniBand traffic between the SGI ICE compute nodes.

Determine the following before you begin your NIS configuration:

- Select the compute node that you want to designate as the NIS master server. You can configure the other compute node(s) as NIS clients. The rack leader controllers (RLCs) in your SGI ICE X system can become NIS slave servers.
- Select an NIS domain name. For example: `ice`.

The procedures assume a SLES operating system. The following topics explain how to configure a compute node as a NIS master server:

- "Configuring a Network Information Service (NIS) Master Server and One or More NIS Slave Servers" on page 28
- "Configuring a Network Information Service (NIS) Client on a Compute Node" on page 30
- "Configuring a Rack Leader Controller (RLC) as a Network Information Server (NIS) Slave Server and Client (SLES)" on page 31
- "NAS Configuration for Multiple IB Interfaces" on page 35
- "Configuring the SGI ICE Compute Nodes as Network Information Service (NIS) Clients (SLES)" on page 34
- "Creating User Accounts (SLES)" on page 38

If you want to use an existing house network NIS server, see "Configuring Network Information Service (NIS) Clients to the House Network's NIS Server" on page 18.

Configuring a Network Information Service (NIS) Master Server and One or More NIS Slave Servers

The procedure in this topic explains how to configure a compute node as a NIS master server and one or more rack leader controllers (RLCs) as NIS slave servers. The procedure applies to compute nodes that run the SLES 11 operating system and uses the text-based YaST2 interface. The graphical YaST2 interface is slightly different.

Procedure 1-8 To configure a compute node as a NIS master server

1. Type the following command to start YaST2:

   ```
   # yast nis_server
   ```

2. Select **Create NIS Master Server**, and select **Next** to continue.

3. Choose an NIS domain name, and type the name into the **NIS Domain Name window**.

   This example uses `ice`.

4. Select **This host is also a NIS client**.

5. Select **Active Slave NIS server exists**.

6. Select **Fast Map distribution**.

7. Select **Allow changes to passwords**.

8. Click **Next** to continue.

   You are now in the **NIS Master Server Slaves Setup**.

   At this point, you can enter the system IDs for the RLCs. If you add new RLCs or if you reconfigure existing RLCs, you need to update this list.

9. In the **Edit Slave** screen, select **Add**, and type `r1lead`.

   If you have other RLCs that you want to configure as NIS slave servers, type the system IDs for those RLCs, too.

   After you specified all the RLCs you want to configure as NIS slave servers, select **Next** to continue.

10. On the **NIS Server Maps Setup**, select **Next**.

    You can use the default selected maps.

    Do not use the **hosts** map. The **hosts** map is not selected by default. This map can interfere with SGI ICE X system operations.

11. On the **NIS Server Query Hosts Setup** screen, select **Finish**.

    You can use the default settings, but SGI recommends that you adjust the settings for security purposes.

    At this point, the NIS master is configured. Assuming you checked the **This host is also a NIS client box**, the compute node will be configured as a NIS client to itself and start `yp ypbind` for you.

Configuring a Network Information Service (NIS) Client on a Compute Node

The procedure in this topic explains how to configure your other compute nodes to be broadcast-binding NIS clients. Do not configure the NIS client on the same compute node that you configured as the NIS master server. For information about how to configure the NIS master server on a compute node, see "Configuring a Network Information Service (NIS) Master Server and One or More NIS Slave Servers" on page 28.

The procedure applies only to compute nodes that host SLES, and the procedure uses the YaST2 interface.

Procedure 1-9 To configure a compute node as a NIS client

1. Enable the `ypbind` service.

   Use one of the following commands:

   - On SLES 12 platforms, type the following command:

     # **`systemctl enable ypbind`**

   - On SLES 11 platforms, type the following command:

     # **`chkconfig ypbind on`**

2. Use the `echo` command, in the following format, to set the default domain:

   `echo "`*NIS_domain_name*`" > /etc/defaultdomain`

   For *NIS_domain_name*, type the domain name you created in Procedure 1-8, step 3 on page 29.

   For example:

   # **`echo "ice" > /etc/defaultdomain`**

3. Edit the `/etc/yp.conf` file to add or remove RLC system IDs, as needed.

   In order to ensure that no NIS traffic goes over the IB network, SGI does **not** recommend using NIS broadcast binding on compute nodes. You can list a few rack leader controllers (RLCs) in the `/etc/yp.conf` file on non-NIS-master compute 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 compute nodes in `yp.conf` file because all resolvable names for compute 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. Type the following command to start the `ypbind` service:

   ```
   # rcypbind start
   ```

   The compute node is now bound.

5. Type the following commands to add the NIS include statement to the end of the password and group files:

   ```
   # echo "+:::" >> /etc/group
   # echo "+::::::" >> /etc/passwd
   # echo "+" >> /etc/shadow
   ```

Configuring a Rack Leader Controller (RLC) as a Network Information Server (NIS) Slave Server and Client (SLES)

The procedure in this topic explains how to set 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.

Procedure 1-10 To configure an RLC as a NIS slave server

1. Through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the InfiniBand Management Tool:

   ```
   # /opt/sgi/sbin/tempo-configure-fabric
   ```

3. On the InfiniBand Management Tool's main menu, click **C Administer InfiniBand ib0** or **C Administer InfiniBand ib1**, and click **Select**.
4. On the **Administer InfiniBand** screen, click **D Status**, and click **Select**.

5. Verify that the output is similar to the following:

```
Master SM
Host = r1lead
Guid = 0x0002c9030006938b
Fabric = ib0
Topology = hypercube
Routing Engine = dor
OpenSM = running
Standby SM
Host = r2lead
Guid = 0x0002c903000773cb
Fabric = ib0
OpenSM = running
```

   The preceding output indicates that the InfiniBand fabric is configured and running.

   If the InfiniBand fabric is not configured, a message similar to the following appears:

```
Error:  Administration of ib0 failed!
```

   If the InfiniBand fabric is not configured, use the InfiniBand Management Tool to configure it. If you prefer, run the larger cluster configuration tool, and use the instructions in the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters* to complete the configuration.

6. Enable daemons.

   The commands differ, depending on your platform, as follows:

   - On SLES 12 platforms, type the following commands:

```
admin:~ # pdsh -g leader systemctl enable ypserv
admin:~ # pdsh -g leader systemctl enable ypbind
admin:~ # pdsh -g leader systemctl enable rpcbind
admin:~ # pdsh -g leader systemctl enable nscd
admin:~ # pdsh -g leader rcrpcbind start
```

- On SLES 11 platforms, type the following commands:

```
admin:~ # pdsh -g leader chkconfig ypserv on
admin:~ # pdsh -g leader chkconfig ypbind on
admin:~ # pdsh -g leader chkconfig portmap on
admin:~ # pdsh -g leader chkconfig nscd on
admin:~ # pdsh -g leader rcportmap start
```

7. Type the following commands:

```
admin:~ # pdsh -g leader "echo NIS_domain_name > /etc/defaultdomain"
admin:~ # pdsh -g leader "ypdomainname NIS_domain_name"
```

   For *NIS_domain_name*, specify the NIS domain name at your site.

   For example, if the NIS domain name at your site is `ice`, type the following commands:

```
admin:~ # pdsh -g leader "echo ice > /etc/defaultdomain"
admin:~ # pdsh -g leader "ypdomainname ice"
```

8. Type the following commands:

```
admin:~ # pdsh -g leader "echo ypserver node_ID > /etc/yp.conf"
admin:~ # pdsh -g leader /usr/lib/yp/ypinit -s node_ID
```

   For *node_ID*, specify the compute node ID that you configured as the NIS master server.

   For example, if `service0` is the NIS master server at your site, type the following commands:

```
admin:~ # pdsh -g leader "echo ypserver service0 > /etc/yp.conf"
admin:~ # pdsh -g leader /usr/lib/yp/ypinit -s service0
```

9. Type the following commands:

```
admin:~ # pdsh -g leader rcportmap start
admin:~ # pdsh -g leader rcypserv start
admin:~ # pdsh -g leader rcypbind start
```

```
admin:~ # pdsh -g leader rcnscd start
```

Configuring the SGI ICE Compute Nodes as Network Information Service (NIS) Clients (SLES)

You can 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 admin node.

The following procedure explains how to configure the SGI ICE compute nodes (blades) as NIS clients.

Procedure 1-11 To configure the SGI ICE compute nodes as NIS clients

1. Through an `ssh` connection, log into the admin node as the root user.
2. Create a SGI ICE 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 SGI ICE compute node image, see"Propagating a Node's Configuration to Another Node" on page 24.
3. Type the following command to specify that the SGI ICE compute nodes use the cloned image/kernel pair:

```
admin:~ # cimage --set ice-sles11-clone 2.6.16.46-0.12-smp "r*i*n*"
```

4. Type the following command to configure the NIS domain:

```
admin:~ # echo "NIS_domain_name" > /var/lib/systemimager/images/ice-sles11-clone/etc/defaultdomain
```

   For *NIS_domain_name*, specify the NIS domain name at your site.

   For example:

```
admin:~ # echo "ice" > /var/lib/systemimager/images/ice-sles11-clone/etc/defaultdomain
```

5. Type the following command to enable the SGI ICE compute nodes to get NIS services from their RLC (fix the domain name as appropriate):

```
admin:~ # echo "ypserver lead-eth" > /var/lib/systemimager/images/ice-sles11-clone/etc/yp.conf
```

6.

   Enable the `ypbind` service.

   Type one of the following commands:

- On SLES 12 systems, type the following command:

```
admin:~# chroot /var/lib/systemimager/images/ice-sles11-clone systemctl enable ypbind
```

- On SLES 11 systems, type the following command:

```
admin:~# chroot /var/lib/systemimager/images/ice-sles11-clone chkconfig ypbind on
```

7. Type the following commands to configure the password, shadow, and group files with NIS includes:

```
admin:~# echo "+:::" >> /var/lib/systemimager/images/ice-sles11-clone/etc/group
admin:~# echo "+::::::" >> /var/lib/systemimager/images/ice-sles11-clone/etc/passwd
admin:~# echo "+" >> /var/lib/systemimager/images/ice-sles11-clone/etc/shadow
```

8. Type the following command to push out the updates:

```
admin:~ # cimage --push-rack ice-sles11-clone "r*"
```

NAS Configuration for Multiple IB Interfaces

You can attach storage devices to a compute node. The NAS cube needs to be configured such that each InfiniBand fabric interface is in a separate subnetwork. The following procedure logically separates the interfaces and attaches them to the same physical network. The procedure configures the large physical network into four smaller subnets. Each subnet becomes capable of containing all the nodes, including the compute nodes. The subnets you configure are as follows:

- `10.149.0.0/18`
- `10.149.64.0/18`
- `10.149.128.0/18`
- `10.149.192.0/18`

This procedure assumes the following:

- The `-ib1` InfiniBand fabric for the SGI ICE compute nodes has addresses assigned in the `10.149.0.0/16` network.
- The lowest address that the cluster management software uses is `10.149.0.1`, and the highest address, which is already assigned to the NAS cube, is `10.149.1.3`.

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, the initial storage node configuration sets the `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. You should be able to ping the IP addresses from the compute node. The compute nodes and the rack leader controllers (RLCs) remain in the 10.149.0.0/16 subnet.

Procedure 1-12 To configure NAS

1. Through an `ssh` connection, log into the admin node as the root user.
2. Use a text editor to open file `/opt/sgi/share/per-host-customization/global/sgi-setup-ib-configs.sh`.

   The next few steps in this procedure modify the file significantly. As a precaution, you can copy the file to a backup location before you begin to edit.
3. Search for `iruslot=$1`.
4. After the line that contains `iruslot=$1`, add the following lines:

```
# 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"
```

5. Search for `$iruslot/etc/opt/sgi/cminfo`.

6. After the line that contains `$iruslot/etc/opt/sgi/cminfo`, add the following lines:

```
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}"
```

7. Search for `IPADDR=’${IB_1_IP}’`, and replace it with `IPADDR=’${IB_1_NAS_IP}’`.

8. Search for `NETMASK=’${IB_1_NETMASK}’`, and replace it with `NETMASK=’255.255.192.0’`.

9. Go to the end of the file, and add the following lines:

```
# 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
```

10. Save and close the file.

11. Use a text editor to open file `/opt/sgi/share/per-host-customization/global/sgi-fstab.sh`.

    The next few steps in this procedure modify the file significantly. As a precaution, you can copy the file to a backup location before you begin to edit.

12. Modify file `sgi-fstab.sh` for the compute blades by adding lines similar to the lines you added to file `/opt/sgi/share/per-host-customization/global/sgi-setup-ib-configs.sh`.

    Perform the following steps:

    - Add a `# Compute NAS interface to use` section to this file.

      For information, see Procedure 1-12, step 4 on page 36.

    - Add lines similar to the following to specify mount points:

```
# SGI NAS Server Mounts
${NAS_IPS[${NAS_IF}]}:/mnt/data/scratch     /scratch nfs    defaults 0 0
```

Creating User Accounts (SLES)

The example in this topic's procedure 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 procedure uses commands, but you could also create accounts using YaST2.

The following procedure explains how to create user accounts.

Procedure 1-13 Creating User Accounts on a NIS Server

1. Through an `ssh` connection, log in to the NIS master compute node as the root user.
2. Use the `useradd`(8) command to add the new user and create a home directory for the new user.

   For example:

   # **useradd -c "Joe User" -m -d /home/juser juser**

3. Use the `passwd`(1) command to create a password for the new user.

   For example:

   # **passwd juser**

4. Type the following command to push the new account to the NIS servers:

   # **cd /var/yp && make**

System Operation

This chapter describes how to operate your cluster and covers the following topics:

Changing Global Cluster Configuration Settings

This topic explains how to use the cluster configuration tool, `configure-cluster`, to enable optional features. 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 use the tool's menus to set system-wide, global values. The values you set apply to all nodes that you discover after you set the value, and the effects are as follows:

- When you configure a system for the first time, you run the cluster configuration tool before you run the `discover` command. All the nodes you discover receive the global values you set in the cluster configuration tool.
- When you add nodes or change global values on a production system, you might need to use commands to reset values on older nodes that you had configured previously.

Note: The `configure-cluster` and `discover` commands work in concert with the cluster definition configuration file, which defines the roles of the various cluster nodes, global system attributes, as well as the data networks and management networks and their respective switches. This configuration file provides a more convenient and efficient method of specifying large-scale changes. For an overview and examples of the cluster definition configuration file, see *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

The following topics explain how to change global cluster configuration settings by using `configure-cluster` or related commands:

- "Changing the Network Time Protocol (NTP) Server" on page 41
- "Changing the House Network's Domain Name Service (DNS) Servers" on page 41
- "Enabling or Disabling a Backup Domain Name Service (DNS) Server" on page 42
- "Configuring a Redundant Management Network (RMN)" on page 42
- "Configuring Database Replication" on page 44
- "Configuring the Default Maximum Individual Rack Unit (IRU) Setting" on page 47
- "Configuring the `blademond` Rescan Interval" on page 48

Changing the Network Time Protocol (NTP) Server

The following procedure explains how to change or update your NTP server information in the cluster configuration database.

Procedure 2-1 To change the NTP server information

1. From the video graphics array (VGA) screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the cluster configuration tool:

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

3. On the cluster configuration tool's main menu, select **T Configure Time Client/Server (NTP)**, and select **OK**.
4. On the **This procedure will replace your ntp configuration file ...** screen, select **Yes**.
5. On the **A new ntp file has been put into position and includes server broadcast entries for the admin node cluster networks ...** screen, select **OK**.

Changing the House Network's Domain Name Service (DNS) Servers

The following procedure explains how to change or update your house DNS server information in the cluster configuration database.

Procedure 2-2 To change the DNS server information

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the cluster configuration tool:

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

3. On the cluster configuration tool's main menu, select **D Configure House DNS Resolvers**, and select **OK**.
4. On the **Enter up to three DNS resolvers IPs** screen, type the IP addresses you want to configure, and select **OK**.

Enabling or Disabling a Backup Domain Name Service (DNS) Server

Typically, the DNS on the admin node provides name services for the cluster. When you configure a backup DNS, however, the SGI ICE compute nodes can use a compute node as a secondary DNS server if the admin node is not available. You can configure a backup DNS only after you run the `discover` command to configure the cluster. This is an optional feature.

The following examples show how to use commands to enable or disable a backup DNS.

- Example 1. To retrieve current DNS backup information, type the following:

```
# /opt/sgi/sbin/backup-dns-setup --show-backup
service0
```

- Example 2. To disable the backup DNS, type the following:

```
# /opt/sgi/sbin/backup-dns-setup --delete-backup
Shutting down name server BIND waiting for named to shut down (28s) done
sys-admin: update-configs: updating SMC configuration files
sys-admin: update-configs: -> dns
. . .
```

- Example 3. To enable a backup DNS on `service0`, type the following:

```
# /opt/sgi/sbin/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 configuration files
sys-admin: update-configs: -> dns
  ...
```

If you want to use the cluster configuration tool to enable or disable the backup DNS, see the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters.*

Configuring a Redundant Management Network (RMN)

By default, an SGI cluster includes an RMN. An RMN is a secondary network from the nodes to the cluster network. When an RMN is enabled, the Linux bonding mode

for RLCs and compute 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 admin node, rack leader controllers (RLCs), and most compute 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.

You can use the cluster configuration tool to enable an RMN. 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 2-3 To enable the RMN from the cluster configuration tool

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the cluster configuration tool:

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

3. On the **Main Menu** screen, select **M Configure Redundant Management Network (optional)**, and select **OK**.
4. On the pop-up window that appears, select **Y yes** (default), and select **OK**.

You can also enable or disable the RMN with the `discover` command's `redundant_mgmt_network` parameter and with the `cadmin` command's `--enable-redundant-mgmt-network` or `--disable-redundant-mgmt-network` parameter. If you use the `cadmin` command to change a compute 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 --enable-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 management network on an RLC, perform the following command:

```
# cadmin --enable-redundant-mgmt-network --node r1lead yes
r1lead should now be rebooted.
# cpower leader reboot r1lead
```

For more information about the RMN, see *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

Configuring Database Replication

SGI clusters store cluster information in an internal database. Database replication is disabled by default on all clusters. On RHEL 7 and SLES 12 platforms, the database is a MariaDB database. On RHEL 6 and SLES 11 platforms, the database is a MySQL database.

SGI recommends that you enable database replication on very large systems with 20 or more rack leader controllers (RLCs) or 20 or more compute nodes. Database replication keeps the internal cluster database synchronized. The master database server resides on the admin node. When you enable replication, data from the master database server is replicated to the database slaves on the RLCs and compute nodes. If your site has a large number of racks, using this feature can reduce the amount of contention for database resources on the admin node.

The following are some situations in which you might need to enable or disable database replication:

- If database replication is enabled, and the database becomes corrupt, you can disable replication on the entire cluster during the debugging session and reenable it later.

In some situations, you might need to keep database replication disabled, either for the entire system or only for selected nodes.

- If the system hosts software that cannot be used when database replication is enabled, SGI recommends that you keep database replication disabled. You can also disable the database replication on a particular node. When you disable synchronization on a specific node, that node uses the admin node for database queries.

To verify whether database replication is enabled on an RLC or compute node, type the following command:

```
# cadmin --show-replication-status --node node
```

For information about how replication is implemented and configured, see one of the following:

- For RHEL 7 or SLES 12 platforms, see the MariaDB documentation for standard replication: https://mariadb.com/kb/en/mariadb/standard-replication/. The relevant sections are the following:

  https://mariadb.com/kb/en/mariadb/replication-commands/
  https://mariadb.com/kb/en/mariadb/setting-up-replication/

- For RHEL 6 or SLES 11 platforms, see the *MySQL 5.0 Reference Manual*. This manual is available at http://dev.mysql.com/doc/refman/5.0/en/replication.html.

The following topics describe how to disable and how to enable database replication:

- "Disabling Database Replication" on page 45 explains how to disable database replication.

- "Enabling Database Replication" on page 46 explains how to enable database replication.

Disabling Database Replication

The following procedures explain how to disable database replication:

- Procedure 2-4, page 46 explains how to disable database replication on one node.

- Procedure 2-5, page 46 explains how to disable database replication on a system-wide basis.

Procedure 2-4 To disable database replication on one compute node

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to disable database replication:

   # **`cadmin --disable-replication --node`** ***node***

   For *node*, type the system ID for the node. For example: `service0`.
3. Type the following command to confirm that database replication is disabled and to ensure that the database exits at the beginning of the script that configures replication:

   # **`cadmin --show-replication-status --node`** ***node***

   At this point, if you run `90-update-mysql` again, you are returned to the system prompt. Unlike the example in the previous step, the command does not issue any messages.

Procedure 2-5 To disable database replication on an SGI ICE X system

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the cluster configuration tool:

   # **`/opt/sgi/sbin/configure-cluster`**
3.

   On the **Main Menu** screen, select **Q Configure Database Replication (optional)**, and select **OK**.
4. On the pop-up window that appears, select **N no**, and select **OK**.

Enabling Database Replication

The following procedures explain how to enable database replication:

- Procedure 2-6, page 47 explains how to enable database replication on one node.
- Procedure 2-7, page 47 explains how to enable database replication on a system-wide basis.

Procedure 2-6 To enable database replication on one compute node

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to enable database replication:

   # **`cadmin --enable-replication --node`** ***node***

   For *node*, type the system ID for the node. For example: `service0`.
3. Type the following command to confirm that database replication is enabled:

   # **`cadmin --show-replication-status --node`** ***node***

4. Type the following command to ensure that the database exits at the beginning of the script that configures replication:

   # **`cattr set --node service0 ignore_my_sql_replication yes`**

   At this point, if you run `90-update-mysql` again, you are returned to the system prompt. Unlike the example in the previous step, the command does not issue any messages.

Procedure 2-7 To enable database replication on a cluster

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Type the following command to start the cluster configuration tool:

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

3. On the **Main Menu** screen, select **Q Configure Database Replication (optional)**, and select **OK**.
4. On the pop-up window that appears, select **Y yes**, and select **OK**.

Configuring the Default Maximum Individual Rack Unit (IRU) Setting

You can configure the maximum number of IRUs 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 SGI ICE

compute nodes in an IRU. If you change this value, the system assigns the new value to any IRUs that you configure.

Procedure 2-8 To configure the default maximum IRU setting from the cluster configuration tool

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.
2. Use the `cadmin` command to retrieve the maximum number of IRUs managed by existing, configured RLCs.

   Type the following command to retrieve the current setting:

   # **`cadmin --show-max-rack-irus --node admin`**

   For SGI ICE X systems, this setting should always be `8`.
3. Type the following command to start the cluster configuration tool:

   # **`/opt/sgi/sbin/configure-cluster`**
4. On the **Main Menu** screen, select **U Configure Default Max Rack IRU Setting (optional)**, and select **OK**.
5. On the window that appears, verify that the value is set to `8`.

   If the value is not `8`, type `8`, and select **OK**. When the maximum IRU setting is configured correctly, the system manages the changes to your system more efficiently.

Configuring the `blademond` Rescan Interval

When enabled, the system checks every two minutes for changes to the number of SGI ICE compute nodes in the system. If you remove or add a new SGI ICE compute node, 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 2-9 To configure the `blademond` rescan interval from the cluster configuration tool

1. From the VGA screen, or through an `ssh` connection, log into the admin node as the root user.

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

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

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

4. 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**.

`discover` Command

The `discover` command configures rack leader controllers (RLCs) and compute nodes (and their associated BMC controllers) in an entire system or in a set of one or more racks that you select. Generally, RLC numbering starts at one and compute node numbering starts at zero. The `discover` command also configures external InfiniBand switches and system management switches.

The *SGI Management Center (SMC) Installation and Configuration Guide for Clusters* describes the process you need to complete in order to configure components in a cluster, including the use of the cluster definition configuration file in conjunction with the `discover` command.

For a complete description of `discover` command usage, type the following:

```
[sys-admin ~]# discover --h
```

The installation and configuration guide describes how you use the `discover` command to configure cluster components. This section describes additional tasks you can perform with the `discover` command:

Using the generic Hardware Type

You can use the `discover` command to configure a cluster component with a hardware type of `generic`. The `generic` hardware type is used for hardware that should be discovered, has only one IP address associated with it, and is to be treated by SMC as an unmanaged cluster component. One likely use is for Ethernet switches needed to extend the management network in large configurations.

When the `generic` hardware type is used for external management switches on large systems, observe the following guidelines:

- The management switches should be the first hardware discovered in the system.
- The management switches should both start with their power cords unplugged, which is analogous to how the system discovers RLCs and compute nodes.
- Explicitly give the external switches high numbers for node numbers if your site does not want SMC to assign them low numbers.
- You can also elect to give these switches an alternate host name by using the `hostname1` flag or by using the `cadmin` command after discovery is complete.

The following example configures two such external switches:

```
admin:~ # discover --nodeset 98,2,generic
```

Marking Cluster Nodes for Deletion

You can use the `--delleader` and `--delnode` options of the `discover` command to mark RLCs and compute nodes for deletion. The two options do not remove nodes completely from the database. Instead, the node is marked with the administrative status `NOT_EXIST`. Later, if you reconfigure a node that previously existed, the system assigns the same IP allocation that it had previously, and the node is marked with the administrative status of `ONLINE`.

For example, if you have a compute node `service0` that has a custom host name of `myhost`, and you later delete `service0` using the `discover --delnode` command, the host name associated with it would still be present. This can cause conflicts if you want to reuse the custom host name of `myhost` on a node other than `service0` in

the future. To completely purge `service0` from the database, use the `cadmin --db-purge --node service0` command. You can then reuse the `myhost` name.

For more information, see "`cadmin`: the Administrative Interface" on page 78.

Configuring a Compute Node to Use a Non-Default Image

The following example configures compute node `0` and uses `service-myimage` instead of the default image:

```
admin:~ # /opt/sgi/sbin/discover --node 0,image=service-myimage
```

Note: For information about how to direct a compute node to image itself with a custom image later, without rerunning the `discover` command, see "`cinstallman` Command" on page 124.

Skipping a Node While Configuring

The following example command configures rack `1`, rack `4`, and the first compute node, and it ignores MAC address `00:04:23:d6:03:1c`:

```
admin:~ # /opt/sgi/sbin/discover --ignoremac 00:04:23:d6:03:1c --leader 1 --leader 4 --node 0
```

Marking a Switch as Deleted

The following example uses the `discover` command to mark an Infiniband switch as deleted:

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

The following example uses the `discover` command to mark a management network switch as deleted:

```
admin:~ # discover --delmgmtswitch num
```

To completely delete a switch from the database, use the `cadmin` command with the `---db-purge` and `---node` parameters.

Enabling or Disabling a Redundant Management Network

The `discover` command includes a parameter that enables or disables a redundant management network for a node at the time you add the node into your configuration. The following example turns off the redundant management network for rack leader 1:

```
admin:~ # discover --leader 1,redundant_mgmt_network=no
```

Omitting Unneeded Switch Configurations When Reconfiguring

By default, the `discover` command performs top-level switch configuration operations each time it runs. If you need to configure additional nodes on a system that is otherwise completely configured, you can direct the `discover` command to omit the unneeded switch configuration steps. By omitting the switch configuration steps, the `discover` command completes its work in less time.

The following procedure explains how to configure a new node for a cluster and skip the switch configuration steps.

Procedure 2-10 To omit switch configuration steps

1. Log into the system as root.
2. Type the following command to retrieve the switch configuration status.

   ```
   # cadmin --show-discover-skip-switchconfig
   ```

   The output from this command is one of the following:

   - `no`, which means that the `discover` command is set to perform the switch configuration processing when it runs
   - `yes`, which means that the `discover` command is set to suppress switch configuration processing when it runs

3. (Conditional) Reset the `discover` command's behavior.

   Perform this step if the previous step returned `no` and your goal is to suppress switch configuration processing when the `discover` command runs.

   Type the following command:

   ```
   # cadmin --disable-discover-switchconfig
   ```

Conversely, if you want to enable switch processing, type the following command:

```
# cadmin --enable-discover-switchconfig
```

4. Type the following `discover` command to configure the new node and omit switch configuration:

```
# cattr set discover_skip_switchconfig yes
```

5. Type the following command to configure the additional node:

```
# discover --rack 1 --macfile macfile-cb14-20130813
```

6. Type the following command to set the `discover_skip_switch` value to `yes` in the database:

```
# cattr set discover_skip_switchconfig yes
```

7. Type the following command to show the value in the database:

```
# cattr list -g discover_skip_switchconfig
```

Managing Slots

The following topics explain how to manage multiple slots:

Retrieving Slot Information

The following procedure explains how to figure out which slot is booted and how to retrieve information about the slots that are confugured currently.

Procedure 2-11 To retrieve slot information

1. Log in as the root user to the admin node.

2. Type the following command to verify the current boot slot:

```
# cadmin --show-current-root
admin node currently booted on slot: 1
```

3. Type the following command to retrieve information about the slots available to be booted:

```
# cadmin --show-root-labels
slot 1: tempo 2.9.0 / sles11sp3: installed on 02/25/2014
slot 2: tempo 2.9.0 / sles11sp2: installed on 02/14/2014
slot 3: tempo 2.9.0 / sles11sp3: backup slot
slot 4: tempo 2.6 / rhel6.5: installed on 06/30/2013
slot 5: tempo 2.8.1 / sles11sp3: installed on 02/17/2014
```

Booting from a Different 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 2-12 To change the boot partition and enable the system to boot from a different slot

1. Log in as the root user to the admin node.
2. Change the default slot.

   You can specify the new slot now, or you can specify the new slot during the reboot. This step explains how to change the boot slot now. Type the `cadmin` command in the following format:

   `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
   ```

   For information about the operating systems installed in each slot, see "Retrieving Slot Information" on page 53.

3. Type the following command to shut down the entire system:

   ```
   # cpower system shutdown
   ```

4. Type the following command to reboot the admin node:

   # **reboot**

5. Connect to the system console to monitor the reboot and, optionally, select a nondefault 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 and press `Enter`.

   If you do not select a new slot, the system boots from the highlighted slot after approximately 10 seconds.

6. Log in as the root user again.

7. Type the following command to reboot all the rack leader controllers (RLCs) and compute nodes:

   # **cpower system reboot**

   If the IP addresses are configured differently within different slots, the `cpower` command might not be able communicate with the baseboard management controllers (BMCs) immediately after you reboot the admin node. If you have trouble connecting to the RLC and compute node BMCs after you change slots, wait a few minutes and issue the `cpower` command again. The wait enables the nodes to obtain new IP addresses.

Cloning a Slot

You can clone, or copy, the installation in one slot to a different slot at any time. SGI recommends that you clone a slot configuration, for example, if you want to modify a slot's images or reconfigure it in any other way. The cloned copy provides a back-up if you need to revert to the original configuration.

The cloning process copies the software for the admin node, the rack leader controller (RLC), and the compute nodes to the slot you specify. The SGI ICE compute nodes do not participate in the cloning process because they are diskless.

Procedure 2-13 To clone a slot

1. Log into the admin node node as the root user.

2. Type the `clone-slot` command in the following format:

   `clone-slot --source` *source_slot_number* `--dest` *destination_slot_number*

   For *source_slot_number*, specify the slot number that contains the configuration you want to clone.

   For *destination_slot_number*, specify the slot number to receive the copy of the configuration.

Note: The cloning process completely destroys all data in the *destination_slot_number*. Be careful not to destroy data in a slot you need when you use this command.

   The `clone-slot` command synchronizes the data and configures the `grub` and `fstab` entries to make the cloned slot a viable booting choice. If the *source_slot_number* slot is the mounted, or active, slot, the `clone-slot` command shuts down the cluster database on the admin node before it starts the backup operation, and the `clone-slot` command the cluster database again when the backup is complete. This sequence ensures that that cluster database does not change during the cloning operation and ensures there is no data loss.

For more information, type the following command:

```
# clone-slot --help
```

For example, the following command clones the configuration in slot 1 to slot 2 and overwrites the contents of slot 2:

```
# clone-slot --source 1 --dest 2
```

Customizing Slot Labels

You can use the `cadmin` command to label the slots on a multiple-boot cluster. After an installation, the slot label is `(none)`.

Procedure 2-14 To customize the slot labels

1. Log into the admin node 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: tempo 3.0.0 / sles11sp3: SLES
slot 2: tempo 3.0.0 / rhel6.6  : RHEL
slot 3: tempo 3.0.0 / centos6.5: CENTOS
slot 4: tempo 3.0.0 / sles11sp3: (none)
slot 5: tempo 3.0.0 / sles11sp3: (none)
```

Modifing Boot Options

You can use the `cadmin` command to set extra kernel boot parameters for SGI ICE compute nodes, compute nodes, and rack leader controller (RLC) nodes on a per-image basis.

For example, to add `cgroup_disable=memory` to the kernel boot parameters for any node that boots the `ice-sles11sp3` image, type the following command:

```
% cadmin --set-kernel-extra-params --image ice-sles11sp3 cgroup_disable=memory
```

If you decide to change the boot parameters, you can issue additional `cadmin` commands. The following additional arguments might be useful to you when you update boot parameters:

- `--show-kernel-extra-params`
- `--unset-kernel-extra-params`
- `--show-nfsroot-extra-params`
- `--set-nfsroot-extra-params`
- `--unset-nfsroot-extra-params`

For more information about boot options, see the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

Power-On/Off Management

This section describes how you can manage the power-on/off status of the cluster using the `cpower` command. The command allows you to manage the power status of the entire cluster or selected components of the cluster. This section consists of the following topics:

Using the `cpower` Command

The `cpower` command allows you to manipulate (power up, power down, reset, etc.) and show the power status of system components. To use the cpower command, you will need to specify a target and an action. The following is the general command format, followed by a description of the parameters. See the following subsections for examples of `cpower` command usage.

`cpower [`*option*`]` *target_type action target_list*

- *option*

  The *option* parameter can be one of the following:

Value	Description
`-h | --help`	Displays the help message. If you enter the `cpower` command without any arguments, you will also get help on command usage.
`-i` *seconds* `|` `--interval=`*seconds*	Specifies how long the identifying LED of the target will be lit. Specify an integer for the number of seconds. Use an interval of 0 to turn off the LED immediately. Valid with the `identity` action.

`-u` | `--no-unmatched`	In the command output, suppress messages that report unmatched targets, names in the target list that do not match any component with the specified target type.
`-v` | `--verbose`	Reports all details in the command output, including all errors.
`-w` *seconds* | `--wait=`*seconds*	Waits until the specified action on the target completes or until the time specified by *seconds* has expired. The *seconds* parameter is required. The command reports its progress as it executes. Valid with actions `on`, `reset`, and `reboot`.

- *target_type*

  The *target_type* parameter is required and can be one of the following:

Value	Description
`switch`	Applies the action to the blade switches specified *target_list*.
`iru`	Applies the action to the independent rack units (IRUs) specified by *target_list*. For the `on` and `off` actions, the dependent the blade switches and SGI ICE compute blades are targeted also.
`leader`	Applies the action to the rack leader nodes specified by *target_list*.
`node`	Applies the action to the compute nodes or SGI ICE compute nodes specified by *target_list*.
`system`	Applies the action to the entire cluster, excluding the admin node. Do not specify *target_list* with this target type.

- *action*

  The *action* parameter is required and can be one of the following:

Value	Description
`cycle`	Power cycles the target by sending an IPMI `cycle` command.

	Valid target types: `leader`, `node`
	The `-wait` option is available for this action.
`halt`	Halts the target by issuing a `halt` command via `ssh`.
	Valid target types: `leader`, `node`, `system`
	If the target type is `system`, compute and ICE compute nodes are halted first; then, the leaders are halted.
`identify`	Turns on the identifying LED of the target for the period specified by the `-i` *seconds* option.
	Valid target types: `leader`, `node`
`off`	Powers off the target by sending an IPMI power-off command.
	Valid target types: `switch`, `iru`, `leader`, `node`, `system`
	If the target type is `system`, compute and ICE compute nodes are powered off first; then, the leaders are powered off.
	If the target type is `iru`, the associated blade switches are also powered off.
`on`	Powers up the target by sending an IPMI power-on command.
	Valid target types: `switch`, `iru`, `leader`, `node`, `system`
	If the target type is `system`, leaders and compute nodes are powered on first; then, the ICE compute nodes are powered on.
	If the target is an ICE compute node, the `on` action ensures that the associated leader and IRU are on. If the associated leader is off, this action powers it on and waits for its successful boot with a 10–minute timeout. Then, the `on` action powers on the

	associated IRU if needed and the ICE compute node in turn. If the target type is `iru`, the associated blade switches are also powered on. For rack leaders and blade switches, the `on` action only powers on the specified target. The `-wait` option is available for this action.
`reboot`	Reboots the target even if already booted by sending a `reboot` command via `ssh`. Valid target types: `leader`, `node` The `-wait` option is available for this action.
`reset`	Performs a hard reset on the target by sending an IPMI `reset` command. Valid target types: `leader`, `node` The `-wait` option is available for this action.
`shutdown`	Shuts down and power off the target by sending a `shutdown -h now` command via `ssh`. Waits for targets to shut down. Valid target types: `node`, `leader`, `system`
`status`	Displays the power status of the target. Valid target types: `iru`, `leader`, `node`, `system` For a target type of `node`, parameter *target_list* is required. The target types of `node` and `leader`, a reported status of `BOOTED` means power is on.

- *target_list*

  The *target_list* is a comma-separated list of hostnames, IRUs, or blade switches. This parameter is required, except for a target type of `system`. To ascertain the names of the targets, use the `discover` command and the cluster definition file.

For details on the `discover` command and the cluster definition file, see *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

You can employ pattern-matching expressions (wildcards) to specify targets. The `cpower` command supports *globbing* expressions. The most commonly used expressions are the following:

Note: The shell uses the same pattern—matching expressions to match filenames. To prevent the shell from matching files in the current directory, quote your target in the target list if it contains such expressions.

Expression	Description
*	Matches one or more characters.
	Example: `"r*lead"` for all rack leaders
?	Matches exactly one character.
	Example: `"r1i?n*"` for all nodes in rack 1 whose IRU number is a single character
[]	Matches any of the range of characters specified within brackets.
	Example: `"r1i2n[1-3]"` for nodes 1, 2, and 3 in IRU 2 of rack 1

Managing the Entire Cluster

To manage the power status of the entire cluster (excluding the admin node), you need only specify the target type `system` and the desired action on the `cpower` command. This section shows examples of such commands.

- Powering down the entire cluster

  ```
  # cpower system off
  ```

  The compute nodes and ICE compute nodes are powered down first; then, the rack leaders are powered down.

- Powering up the entire cluster

```
# cpower system on
leader node r1lead power ON
600 sec wait for leader r1lead to boot
direct node service0 power ON
leader node r1lead is BOOTED
leader node r1lead is BOOTED
compute node r1i0n0 BOOTED
compute node r1i0n3 BOOTED
compute node r1i0n4 BOOTED
...
compute node r1i2n2 BOOTED
compute node r1i2n11 BOOTED
compute node r1i2n4 BOOTED
compute node r1i2n14 BOOTED
compute node r1i2n15 BOOTED
```

As noted in the preceding section, the the rack leaders and compute nodes are powered on first, followed by the ICE compute nodes.

- Querying the power-on/off status of the cluster

```
# cpower system status
service0     BOOTED
r1lead       BOOTED
r1i0n0       BOOTED
r1i0n1       BOOTED
r1i0n2       BOOTED
...
r1i3n14      BOOTED
r1i3n15      BOOTED
r1i3n16      BOOTED
r1i3n17      BOOTED
```

Managing [ICE] Compute Nodes

Managing compute nodes and ICE compute nodes with the `cpower` command requires that you specify a target type of `node`, an action, and a target list. This section shows examples of such commands.

- Powering on a compute node

```
# cpower node on service0
```

Powers on `service0`, compute node 0.

- Powering on all compute nodes

```
# cpower node on "service*"
```

Note the use of quotes with the argument with a wildcard to ensure that a matching filename is not targeted.

- Querying the status of all compute nodes

```
# cpower node status "service*"
```

Displays the status of all compute nodes.

- Powering down a compute node

```
# cpower node off service0
```

- Rebooting a compute node

```
# cpower node reboot service0 -w 180
```

Reboots compute node 0 with a three-minute timeout.

- Powering on an ICE compute node

```
# cpower node on r1i3n10
```

Powers on the ICE compute node at rack 1, IRU 3, slot 10.

Note: If the associated leader is off, this action powers it on and waits for its successful boot with a 10–minute timeout. Then, the `on` action powers on the associated IRU if needed and the ICE compute node in turn.

- Powering on a group of ICE compute nodes in a rack

```
# cpower node on "r1i0n[2-5]" -w 300
cmc node r1i0c power ON
compute node r1i0n3 already BOOTED
compute node r1i0n5 power ON
compute node r1i0n4 power ON
compute node r1i0n2 power ON
```

```
compute node r1i0n5 is BOOTED
compute node r1i0n2 is BOOTED
300 second timeout exceeded waiting for boot of r1i0n4
```

Powers on and attempts to boot ICE compute nodes in slots 2, 3, 4, and 5 in IRU 0 of rack 1. Note the 5–minute wait time for booting.

- Querying the status of all ICE compute nodes in a rack

  # **cpower node status "r1i*n*"**

- Powering off an ICE compute node

  # **cpower node off r1i3n10**

  Powers off only the specified ICE compute node. The associated rack leader and IRUs are unaffected.

- Rebooting an ICE compute node

  # **cpower node reboot r1i3n10**

  Reboots the specified ICE compute node.

- Highlighting an ICE compute node

  # **cpower node identify r1i3n10 -i 60**

  Turns on the ID LED on the specified ICE compute node for 60 seconds.

Managing Rack Leaders

Managing the power status of rack leaders is quite similar to managing that of [ICE] compute nodes. With the `cpower` command, you will need to specify a target type of `leader`, an action, and a target list. This section shows examples.

- Powering on a rack leader

  # **cpower leader on r1lead**

  Powers on only the leader for rack 1.

- Shutting down a rack leader

  # **cpower leader shutdown r3lead**
  leader node r3lead has been issued a shutdown -h now command

```
leader node r3lead is DOWN
```

Shuts down only the specified rack leader. The associated ICE compute nodes and IRUs are unaffected.

- Querying the status of all rack leaders

```
# cpower leader status "*"
r1lead       BOOTED
r2lead       BOOTED
r3lead       OFF
```

Displays the power status of all rack leaders.

- Rebooting a rack leader

```
# cpower leader reboot r3lead -w 180
```

Reboots the specified rack leader with a three-minute timeout.

- Highlighting all rack leaders

```
# cpower leader identify "r*lead" -i 60
```

Turns on the ID LED on all rack leaders for 60 seconds.

Managing IRUs

To power-manage the IRUs, use the `cpower` command with the target type of `iru`, an action, and a target list. You can specify an IRU by its rack number and its IRU number. For example, `r1i1` specifies IRU 1 on rack 1.

Note that powering on an ICE compute node does effectively power on its associated leader and IRU, but the converse is not true. Likewise, powering on/off an IRU powers on/off its associated blade switches and ICE compute blades. This section shows some examples.

- Powering on an IRU

```
# cpower iru on r1i0
```

Powers on IRU 0 in rack 1.

- Powering off an IRU

```
# cpower iru off r3i1
```

Powers off IRU 1 in rack 3 and associated blade switches and ICE compute nodes.

- Powering off all IRUs in a rack

  # **cpower iru off "r3i*"**

  Powers off all IRUs, blade switches, and ICE compute nodes in rack 3. Note the use of quotes with the argument with a wildcard to ensure that a matching filename is not targeted.

-

Managing Blade Switches

Like IRUs, the blade switches can be power-managed selectively. You can turn them on and off and query their power status. To power-manage the blade switches, use the `cpower` command with the target type of `switch`, an action, and a target list. You can specify a blade switch by its switch number qualified by its associated rack and IRU. For example, `r1i0s0` specifies switch 0 associated with IRU 0 on rack 1. This section shows some examples.

- Powering on a blade switch

  # **cpower switch on r1i0s0**

  Powers on switch 0 associated with IRU 0 in rack 1.

- Powering off a blade switch

  # **cpower switch off r3i1s1**

  Powers off switch 1 associated with IRU 1 in rack 3.

- Querying the status of all blade switches

```
# cpower switch status "*"
r1i0s0       ON
r1i0s1       ON
r1i1s0       ON
r1i1s1       ON
r1i2s0       ON
r1i2s1       ON
r1i3s0       ON
r1i3s1       ON
```

Power/Energy Management

This section describes how you can query and limit the incoming power to the cluster. The following topics are described:

Features of SMC Power/Energy Management

SMC has a power management service which is used to carry out various power-related operations. The power service supports system-level actions on behalf of the system administrator using the `mpower` command as well as job-level power operations via the workload manager PBS Pro. The system-level power management operations include the following:

- Reading power and energy data from the entire cluster, individual racks, or specific nodes
- Setting a power limit for the entire system, a rack, or specific compute nodes
- Reading back a power limit set on any target at any time

  The power limit that is set on any compute node is kept in the compute node NVRAM. Hence, the limit will be saved across reboots and power cycles.

- Inlet temperature throttling

  A safety feature to pre-program each compute node with an air temperature threshold at which the compute node initiates maximum processor power limiting. The power limiting allows the compute nodes to continue to run at their slowest frequency and minimal heat production. This can forestall an overheating shutdown brought on by a high room temperature.

The job-level power management is performed by the power service on behalf of the workload manager PBS Pro. The PBS Pro software connects to the power service using an API to define logical groupings of compute nodes into nodesets and then

reads the power and energy for the nodeset as well as gets and sets power limits. The job-level power management does not include inlet-air-temperature power limiting.

The system-level and job-level forms of power management co-exist. Should there be situations where the system-level power management set a power limit on any nodes as well as the job-level power management, the individual compute node power manager selects the most restrictive power limit of the two.

SMC power/energy management requires hardware capabilities which are available on SGI ICE X, SGI ICE XA, and Rackable systems supporting the Intel® Xeon® E3–2600 and E5-2600 families of processor platforms. The ICE 8200/8400 systems do not support power/energy management. To ensure that this feature is properly configured on your cluster, see section "Verifying Power Operations and Configuring Power Management" in the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters.*

Using the `mpower` Command

Caution: Do not set up a script to invoke the `mpower` command in a tight loop. The power service polls all compute nodes in the cluster as well as performs various housekeeping tasks. Such a script would severely disrupt the power service or the power management hardware or both and consequently cause many unintended problems.

The `mpower` command allows you to target the entire cluster, a set of racks, or a set of compute nodes. You can perform any of the following operations on the target:

- Read the power and energy.
- Set or read a power limit.
- Set or read an inlet-temperature threshold.

This section shows the general command format and describes its parameters. The following sections contain examples of `mpower` use.

`mpower [`*option*`]` *target_type action target_list limit*

The following list describes each parameter.

- *option*

Allows you to request help for the command or verbose command output. The valid values are described in the following:

Value	Description
`-h`	Requests help for the command. If you enter the `mpower` command without any arguments, you will also get help on command usage.
`-v`	Requests the verbose command output mode for reports and errors.

- *target_type*

  A required parameter that specifies the domain or type of target enumerated in the *target_list* parameter. The valid values are described in the following table.

Value	Description
`node`	Applies the action to the compute nodes or SGI ICE compute nodes specified by *target_list*.
`rack`	Applies the action to the racks specified by *target_list*. The action targets all compute nodes, rack leaders, and SGI ICE compute nodes in the specified racks.
`system`	Applies the action to the entire cluster, excluding the admin node. Do not specify the *target_list* parameter with this target type.

- *action*

  Specifies the action to be applied to the target. This parameter is required. The valid values are described in the following table.

Value	Description
`get_limit`	Displays the power limits set for the designated targets.
`set_limit`	Sets the power limit `limit` to the designated targets.
`get_power`	Displays the current power readings for the designated targets.

Note: For a target type of `node`, the power reading is a moving average of watts. The `get_power` action uses an averaging algorithm that requires a power limit to be set. To ensure that such a limit exists, set a non-restrictive power limit for the entire cluster. See section "Targeting the Entire Cluster" on page 73.

Unlike that of target type `node`, the power readings for target types `rack` and `system` are cumulative measurements, not an average.

For racks and the entire cluster, the `mpower` command performs power readings differently for Rackable clusters than it does for SGI ICE X and SGI ICE XA clusters. For Rackable clusters, the command reads the individual compute node power managers and sums the results accordingly for the rack or system. For SGI ICE X and SGI ICE XA clusters, the command reads the rack power from the chassis management controllers (CMCs), not from the power management hardware of the individual SGI ICE compute nodes.

The cumulative power reading for racks and entire cluster is relative to the last data reset. See the `reset_stats` action.

`reset_stats`	Resets the data collection window for the `get_power` action with respect to racks and the entire cluster.
`get_inlet`	Displays the inlet temperature threshold set for the designated targets.
`set_inlet`	Sets the inlet temperature threshold `limit` to the designated targets.

- *target_list*

  The *target_list* is a comma-separated list of hostnames or rack names. This parameter is required, except for a target type of `system`.

For a `target_type` of `rack`, the `target_list` format is `rack1`, `rack2`, etc.

For a `target_type` of `node`, use a list of hostnames. To ascertain the hostnames, use the `discover` command and the cluster definition file. For details on the `discover` command and the cluster definition file, see *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

You can employ pattern-matching expressions (wildcards) to specify targets. The `mpower` command supports *globbing* expressions. The most commonly used expressions are the following:

Note: The shell uses the same pattern-matching expressions to match filenames. To prevent the shell from matching files in the current directory, you can quote your target list if it contains such expressions.

Expression	Description
*	Matches one or more characters.
	Example: `"rack*"` for all racks
?	Matches exactly one character.
	Example: `"r1i?n*"` for all nodes in rack 1 whose IRU number is a single character.
`[]`	Matches any of the range of characters specified within brackets.
	Example: `"rack1[1-4]"` for racks 11, 12, 13 and 14

- *limit*

  The *limit* argument is only required when you specify the `set_limit` or the `set_inlet` action.

  – Coupled with the `set_limit` action, the *limit* argument specifies the power limit in watts to be enforced on the target. Specify the value in the format of *n*`W`, where *n* is an integer in the 50–1000 range, inclusive.

    The ICE compute nodes typically cannot limit power below 100 watts and Rackable compute nodes, below 150 watts. The maximum power for an ICE compute node varies by processor and memory configuration but typically cannot reach or exceed 400 watts. The maximum power for a Rackable

compute node also varies by configuration (processor / memory / onboard disks) and may exceed 600 watts.

– Coupled with the `set_inlet` action, the *limit* argument specifies the maximum inlet temperature in degrees Centigrade allowed on the target before SMC initiates remedial actions. Valid values are integers in the 20–45 range, inclusive.

Targeting the Entire Cluster

To manage the power of the entire cluster, you need to specify the target type `system` and the desired action on the `mpower` command. This section shows examples of such commands.

- Display the power usage for the entire cluster.

```
# mpower system get_power
System Power Stats:
Instant                           : 46036.03
Minimum during sampling period    : 452.31
Maximum during sampling period    : 9761.39
Average during sampling period    : 2301.01
KwH during sampling period        : 32137.07
```

  Displays the power and energy use of the cluster since the last time the data was reset.

- Reset the starting point for calculating the energy use for the entire cluster.

```
# mpower system reset_stats
```

- Get the power limit for the entire cluster.

```
# mpower system get_limit
r1i5n0         1000W
r1i5n1         1000W
r1i5n2         1000W
...
r1i5n12        1000W
r1i5n13        1000W
r1i5n14        1000W
r2i6n0         1000W
```

```
r2i6n1          1000W
r2i6n2          1000W
...
r2i6n12         1000W
r2i6n13         1000W
r2i6n14         1000W
```

- Set a power limit of 1000W for the entire cluster.

  # **mpower system set_limit 1000W**

  The 1000W power limit, the maximum value allowed, is a non-restrictive power limit. It allows the system to operate as if no power limit were in place. However, it does ensure that the averaging algorithms associated with `get_power` action work properly.

- Get the inlet temperature threshold set for the entire cluster.

```
# mpower system get_inlet
r1i5n0          40
r1i5n1          40
r1i5n2          40
...
r1i5n12         40
r1i5n13         40
r1i5n14         40
r2i6n0          40
r2i6n1          40
r2i6n2          40
...
r2i6n12         40
r2i6n13         40
r2i6n14         40
```

- Set an inlet temperature threshold for the entire cluster.

  # **mpower system set_inlet 30**

  Sets the inlet temperature threshold for the entire cluster to 30°C. Effectively sets this threshold for all nodes in the cluster.

Caution: The inlet temperature threshold should be a value higher than the typical operating temperature of the data center, but below 45°C. When the room air temperature (ambient temperature) reaches the inlet temperature setting, the compute nodes begin maximum power limiting until the room air temperature falls below the threshold setting. At 45°C, the compute nodes shut down completely, regardless of the inlet temperature setting.

Targeting the Racks

To manage power at the rack level, you need to specify the target type `rack`, the desired action, and a target list on the `mpower` command. This section shows examples of such commands.

- Display the power usage for rack 1 of the cluster.

```
# mpower rack get_power rack1
r1lead:
Instant                            : 45966.63
Minimum during sampling period     : 452.31
Maximum during sampling period     : 9761.39
Average during sampling period     : 2301.42
KwH during sampling period         : 32146.15
Sampling period                    : 6411536.12
```

  Displays the power use of rack1 in the cluster since the last time the data was reset.

- Reset the starting point for calculating the power use for all racks.

```
# mpower rack reset_stats "rack*"
```

  Note the use of quotes with the argument with a wildcard to ensure that a matching filename is not targeted.

- Display the power limit for rack 1.

```
# mpower rack get_limit rack1
r1i15n0       1000W
r1i15n1       1000W
```

```
r1i15n2       1000W
...
r1i15n14      1000W
r1i15n15      1000W
r1i15n16      1000W
```

- Set a power limit of 400W for rack1.

  # **mpower rack set_limit rack1 400W**

- Display the inlet temperature threshold for all compute nodes in a rack1.

```
# mpower rack get_inlet rack1
r1i14n9       40
r1i14n10      40
r1i14n11      40
r1i14n12      40
r1i14n13      40
r1i14n14      40
r1i14n15      40
r1i14n16      40
r1i14n17      40
```

- Sets an inlet temperature threshold of 30°C for all compute nodes in a rack1.

  # **mpower rack set_inlet rack1 30**

Targeting the Nodes

To manage the power for compute nodes or ICE compute nodes, you need to specify the target type `node`, the desired action, and a target list on the `mpower` command. This section shows examples of such commands.

- Display the power usage for ICE compute node `r1i1n2`.

```
# mpower node get_power r1i1n2
r1i1n2    126W
```

  Displays the power usage for the node. The power reading returned is a moving average. This command fails if no power policy exists. See the description of the `get_power` action in section "Using the `mpower` Command" on page 69.

- Display the power limit for ICE compute node `r1i1n2`.

  ```
  # mpower node get_limit r1i1n2
  r1i1n2    1000W
  ```

- Set a power limit of 200W for ICE compute node `r1i1n2`.

  ```
  # mpower node set_limit r1i1n2 200W
  ```

- Display the power limit for compute node `service0`.

  ```
  # mpower node get_limit service0
  service0    1000W
  ```

- Set a power limit of 300W for compute node `service0`.

  ```
  # mpower node set_limit service0 300W
  ```

- Display the inlet temperature threshold for compute node `service0`.

  ```
  # mpower node get_inlet service0
  service0    40
  ```

- Set an inlet temperature threshold for compute node `service0`.

  ```
  # mpower node set_inlet service0 29
  ```

  Sets the inlet temperature threshold to 29°C for that node.

pdsh and pdcp Commands

The `pdsh`(1) command is the parallel shell utility. The `pdcp`(1) command is the parallel copy/fetch utility. The system software populates some `dshgroups` files for the various node types. On the admin node, the system software populates the `leader` and `compute` groups files, which contain the list of online nodes in each of those groups.

On the rack leader controller (RLC), software populates the `ice-compute` group for all the online SGI ICE compute nodes in that group.

On the compute 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.

From the admin node, to run the `hostname` command on all the RLCs, perform the following:

```
# pdsh -g leader hostname
```

To run the `hostname` command on all the SGI ICE compute nodes in the system, via the RLCs, perform the following:

```
# pdsh -g leader pdsh -g ice-compute hostname
```

To run the `hostname` command on just `r1lead` and `r2lead`, perform the following:

```
# pdsh -w r1lead,r2lead hostname
```

`cadmin`: the Administrative Interface

After you log into the admin node, you can use the `cadmin` command to administer the cluster. To retrieve the `cadmin` usage statement, type the following command:

```
[sys-admin ~]# cadmin --h
```

The following sections include examples that show how to use the `cadmin` command:

Bringing a Node Online or Setting a Node Offline

The following examples show how to bring a node online or set a node offline:

- To set `r1i0n0` offline, type the following command:

  # **cadmin --set-admin-status --node r1i0n0 offline**

- To set `r1i0n0` offline, type the following command:

  # **cadmin --set-admin-status --node r1i0n0 online**

Changing Compute Node Information

The following examples show how to change information for a compute node:

- To retrieve the IP addresses currently configured for `service0`, type the following command:

```
admin:~ # cadmin --show-ips --node service0
IP Address Information for SMC 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
```

- To set the boot order for compute node `service0`, type the following command:

  # **cadmin --set-boot-order --node service0 2**

- To change the IP address on `service0-ib0`, type the following command:

  admin:~ # **cadmin --set-ip --node service0 --net head service0=172.23.0.199**

- To add an additional, discretionary IP address to compute node `service0`, type the following command:

  # **cadmin --add-ip --node service0 --net ib-0 my-new-ib0-ip=10.14.0.2**

- To delete an additional, discretionary, site-added IP address from compute node `service0`, type the following command:

  admin:~ # **cadmin --del-ip --node service0 --net ib-0 my-new-ib0-2-ip=10.14.0.2**

Note that you cannot delete the IP addresses that the system requires.

- To change the hostname of `service0` to `myservice`, type the following command:

  ```
  admin:~ # cadmin --set-hostname --node service0 myservice
  ```

Changing the Admin Node Hostname and IP Address on the House Network

The following procedure explains how to retrieve information about the admin node and update the admin node hostname or IP address. The examples also show how to change the address information for the admin node on the house network.

Procedure 2-15 To change the admin node's house network IP address

1. Log into the admin node as the root user.
2. Use the `cadmin` command to retrieve information about the current house network IP address.

   For example:

   ```
   admin:~ # cadmin --show-house-network-info
   -----Network Information-----
   broadcast       :       137.38.82.255
   base_ip :       137.38.82.0           # the IP of the house network
   netmask :       255.255.255.0
   gateway :       137.38.82.254
   ip      :       137.38.82.166         # the IP address of the admin node
   ```

3. Use the `cadmin` command in the following format to assign a new IP address to the admin node:

   `cadmin --set-house-network` *ip_addr*, *netmask*, *gateway_info*

   The arguments to the command are as follows:

   - For *ip_addr*, specify the new IP address that you want to assign to the admin node.
   - For *netmask*, specify the network mask you want to assign to the new IP address.
   - For *gateway_info*, specify either the default gateway you want to assign to the new IP address or the keyword `no_gateway`.

4. Use the `service network restart` command to propagate changes throughout the cluster.

   When you use the `cadmin` command's `--set-house-network` parameter to change any of the networking information, you need to restart the network services and propagate your changes to the cluster.

   The following examples show how to use the `service network restart` command:

   Example 1:

   ```
   admin:~ # cadmin --set-house-network 137.38.82.165,255.255.255.0,137.38.82.253
   admin:~ # service network restart
   ```

   Example 2:

   ```
   admin:~ # cadmin --set-house-network 137.38.82.165,255.255.255.0,no_gateway
   admin:~ # service network restart
   ```

You can use the `cadmin --set-house-network` command to specify a new network mask or new gateway information for the admin node, too. In that case, specify the existing admin node IP address, the new network mask, and/or the new default gateway.

To change the hostname associated with the admin node to be `newname`, type the following command:

```
admin:~ # cadmin --set-hostname --node admin newname
```

Displaying Network Information

The following examples show how to use the `cadmin` command to display network information:

- To set and show the cluster subdomain, type the following commands:

  ```
  admin:~ # cadmin --set-subdomain mysubdomain.domain.mycompany.com
  admin:~ # cadmin --show-subdomain
  The cluster subdomain is: mysubdomain
  ```

- To retrieve the admin node house network domain, type the following command:

  ```
  admin:~ # cadmin --show-admin-domain
  The admin node house network domain is: domain.mycompany.com
  ```

Changing Switch Management Network Settings

The following examples show how to use the cadmin command to change the switch management network settings:

- To retrieve the current switch management value for a specified node, type the following command:

  ```
  admin:~ # cadmin --show-switch-mgmt-network --node admin
  no
  ```

  In the preceding example, the returned value of no means that there is no switch management network, which is the default for SGI 8200/8400 systems. Note that this is not the default for other platforms.

- To enable the switch management network for a specificed node that is connected to managed top level switches, type the following command:

  ```
  admin:~ # cadmin --enable-switch-mgmt-network --node admin
  ```

- To disable the switch management network for a specificed node that is connected to managed top level switches, type the following command:

  ```
  admin:~ # cadmin --disable-switch-mgmt-network --node admin
  ```

Changing Database Replication Settings

Database replication is disabled by default. The following examples show how to use the cadmin command to change replication settings:

- To retrieve the replication status for a specified admin node, rack leader controller (RLC), or compute node, type the following command:

  ```
  admin:~ # cadmin --show-replication --node r2lead
  yes
  ```

- To enable database replication on a specified admin node, RLC, or compute node, type one the following commands:

  On RHEL 6 or SLES 11 platforms

  ```
  admin:~ # cadmin --enable-replication --node r2lead
  Running 'ssh r2lead /etc/opt/sgi/conf.d/90-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
```

On RHEL 7 platforms

```
# cadmin --enable-replication --node r1lead
Running 'ssh r1lead /etc/opt/sgi/conf.d/90-update-mysql' ...
MySQL slave running.
MySQL DB replicated.
```

On SLES 12 platforms

```
# cadmin --enable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/90-update-mysql' ...

Note: This output shows SysV services only and does not include native
systemd services. SysV configuration data might be overridden by
native systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use
'systemctl list-dependencies [target]'.

mysql                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
MySQL slave running.
MySQL DB replicated.
```

- To disable database replication on a specified admin node, RLC, or compute node, type one the following commands:

  On RHEL 6 or SLES 11 platforms

  ```
  admin:~ # cadmin --disable-replication --node r2lead
  Running 'ssh r2lead /etc/opt/sgi/conf.d/90-update-mysql' ...
  Shutting down service MySQL ..done
  mysql                   0:off  1:off  2:off  3:off  4:off  5:off  6:off
  ```

  On RHEL 7 platforms

  ```
  # cadmin --disable-replication --node r1lead
  Running 'ssh r1lead /etc/opt/sgi/conf.d/90-update-mysql' ...
  rm '/etc/systemd/system/multi-user.target.wants/mariadb.service'
  ```

On SLES 12 platforms

```
# cadmin --disable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/90-update-mysql' ...
```

Changing Console Management Settings

To avoid excessive console logging and `ipmitool` processes when there are hundreds of flat compute nodes connected to the system, you can suppress console logging and reduce the number of active IPMI processes. You can do so by managing the following two attributes:

- Console logging

  This feature is enabled by default. The `cadmin --show-conserver-logging` and `--set-conserver-logging` options control the logging for the targeted node(s). The global value is set to enable this feature. It can also be set per node if desired. The global setting impacts the SGI ICE compute nodes tied to leaders.

- Console ondemand

  This feature is disabled by default. If enabled, this feature allows IPMI to connect to the BMC to access the console (and only log if logging is enabled) only when there is an active console session per the `console`command. The `cadmin --show-conserver-ondemand` and `--set-conserver-ondemand` options can manage this feature.

Managing UDP Multicast (UDPcast) Provisioning

SMC supports UDP multicast provisioning, which allows you to quickly install hundreds of compute nodes at once. UDPcast allows for a large number of nodes to join a multicast stream of the content being transported. With all of the nodes sharing a single stream at a time, the network is protected from being saturated by disjoint installations.

UDPcast is used in the following three areas of the system:

- Leaders booting SGI ICE compute nodes in the tmpfs boot mode
- A compute image being pushed from the admin node to the leader node for the first time
- An admin node installing flat compute (service) nodes when they are configured with the `transport=udpcast` parameter

Overview of UDPcast

Udpcast is the basic tool used for multicast installation. It has two primary commands:

- `udp-sender` — Sends a single image stream to one or more receivers.
- `udp-receiver` — Issued by the recipients to listen to the stream.

Flamethrower

SMC uses the wrapper program Flamethrower to manage UDPcast and make it into a solution suitable for installing systems and pushing images.

It maps `udp-sender` commands to content to be transported. It starts a `udp-sender` on a unique port for each component to be transported. When `udp-sender` terminates (due to a transfer being complete), Flamethrower starts a new one.

The content managed by Flamethrower includes the Flamethrower directory itself, the systemimager boot environment, and any available images. For each image, there are two components: the image itself and the overrides associated with the image.

On a system with three images, there are typically 10 different pieces of content to manage, each with a dedicated `udp-sender` process running on a unique port.

On the admin node, `udp-sender` is run in tar-pipe mode, which means the image is run through tar via a pipe. This means that separate tar files for each image do not need to be maintained. What is being transported is always the current image.

Flamethrower Directory

All of the content managed by Flamethrower is listed in the Flamethrower directory. The directory contains a module file for each piece of content that is to be sourced by Bash.

When a node is interested in multicast content, it first uses `udp-receiver` to transfer the Flamethrower directory. Once the node has the directory, it has the list of components to transport and the port numbers to use. It then uses `udp-receiver` to transfer the desired content.

Management Ethernet

The management Ethernet switches need to be configured to properly handle multicast traffic. If switches supported and configured by SMC are used, the switches should be set up automatically for this. If switches that are not configured by SMC

are used, then the switches need to be carefully configured to allow multicast traffic to be transported.

The multicast IP addresses in question are adjustable for the RDV address (the address used for nodes to find each other). At this time, the data transport IP addresses are not configurable. The admin node use 239.0.0.1 by default for RDV, which often requires special switch configuration to work properly. The leader node (serving SGI ICE computes) use 224.0.0.1 for RDV by default. More information on these IP addresses and configuration adjustments that can be made are described later in "UDPcast Configuration Tuning" on page 87.

Node Memory Use for Flat Compute and Leader Nodes

Flat compute (service) nodes and leader nodes installed using UDPcast need to have enough system memory to hold the image. The image is stored in to a tmpfs filesystem on the node during installation to make the transport more efficient. With hundreds of nodes listening to a stream, writing the data directly to disk would slow down the transfer for all nodes. For this reason, the data is saved to tmpfs first and then expanded onto the system disk. If you have nodes with very little memory, UDPcast installation could fail for this reason.

Node Memory Use for SGI ICE Compute Nodes in tmpfs Mode

The UDP receiver is used in tar-pipe mode; that is, the files are expanded from a pipe directly to the tmpfs filesystem, which is used as the root filesystem.

Provisioning Flat Compute Nodes or Leaders

For the first installation and initialization (discovery) of a flat compute node or leader nodes, you can select UDPcast provisioning by using the `transport=udpcast` option on the `discover` command. Optionally, you can specify `transport=udpcast` parameter in the node definition in the cluster definition file. For more details on the `discover` command and the cluster definition file, see *SGI Management Center (SMC) Installation and Configuration Guide for Clusters.*

If the node is already discovered, you can re-install using UDPcast by using the `cinstallman` command with the `--transport` option as shown in the following example:

`cinstallman --next-boot` *imagename* **`--node n\* --transport udpcast`**

Note: If you have adjusted UDPcast settings using `cadmin` or `cattr` commands, the images need to be re-pushed to the leaders. This ensures the following:

- Flamethrower on the leader node serving SGI ICE compute nodes is set up and the needed UDP sender processes are running on the designated ports.
- The SGI ICE compute node tmpfs network boot files have the appropriate configuration details.

UDPcast Configuration Tuning

There are a number of settings you can fine tune to optimize the the performance of UDPcast. The goal is to get the majority of the nodes to listen to a stream at the same time. Various settings affect the wait time for neighbors to join. Note that it is not necessarily bad if not all nodes join the stream at the same time. The UDP receiver will happily wait for the current stream to complete and join when the new UDP sender and its new stream starts. In this case, a batch of nodes could grab the first stream and nodes missing that stream could join the second. You can tune the following attributes:

- `flamethrower-directory-portbase`
- `udpcast-min-receivers`
- `udpcast-min-wait`
- `udpcast-max-wait`
- `udpcast-max-bitrate`
- `udpcast-mcast-rdv-addr`
- `udpcast-rexmit-hello-interval`

flamethrower-directory-portbase

The `flamethrower_directory_portbase` attribute is the port number for the Flamethrower directory itself. This is important because all nodes need access to the Flamethrower directory to find the appropriate port number for pertinent content. This port number is provided as a kernel parameter for flat compute (service) and leader nodes when using the UDPcast transport as well as SGI ICE compute nodes when in tmpfs mode. The default is 9000.

Note: SGI does not expect you to change this value. If you do need to adjust this value, you can do so with the `cattr` command. Additionally, if you need to adjust this value, please contact SGI Support as SGI would like to better accommodate such cases.

udpcast-min-receivers

This defines the minimum number of receivers that must be present before the UDP sender will start the stream.

Note: This UDP sender behavior is modified by `udpcast-max-wait` attribute.

This value can be changed with the `cadmin` command:

- See the help for options `--set-udpcast-min-receivers` and `--show-udpcast-min-receivers`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.
- The admin node value is what is used by the admin node to serve flat compute and leader nodes using UDPcast transport.

See the `udp-sender` man page for additional details.

udpcast-min-wait

The `udpcast-min-wait` attribute defines the minimum time that the UDP sender waits before starting a given stream. UDP sender will wait the minimum time for `udpcast-min-receivers` receivers (described earlier) to join the stream.

See also descriptions for `udpcast-min-receivers` and `udpcast-max-wait` in this section and the `udp-sender` man page for additional details.

This value can be changed with the `cadmin` command:

- See the help for options `--set-udpcast-min-wait` and `--show-udpcast-min-wait`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.

- The admin node value is what is used by the admin node to serve flat compute and leader nodes using UDPcast transport.

udpcast-max-wait

The `udpcast-max-wait` attribute defines the maximum time a UDP sender will wait before starting a stream. If the minimum number of receivers have not joined by this time, the stream is started anyway. See the `udp-sender` man page for more details.

This value can be changed with the `cadmin` command:

- See the help for options `--set-udpcast-max-wait` and `--show-udpcast-max-wait`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.
- The admin node value is what is used by the admin node to serve flat compute and leader nodes using UDPcast transport.

udpcast-max-bitrate

The `udpcast-max-bitrate` attribute defines the stream bit rate that a UDP sender attempts to achieve. This is a very important attribute. If the bit rate is set too fast, then there will be excessive re-transmits and re-tries. The default is 900m. See the `udp-sender` man page for more details.

This value can be changed with the `cadmin` command:

- See the help for options `--set-udpcast-max-bitrate` and `--show-udpcast-max-bitrate`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.
- The admin node value is what is used by the admin node to serve flat compute and leader nodes using UDPcast transport.

udpcast-mcast-rdv-addr

The `udpcast-mcast-rdv-addr` attribute is the IP address used for senders and receivers to find each other (rendez-vous). This setting has important implications for switch configuration. If you are using switches not configured by SGI tools, special

care will be needed to ensure that multicast traffic is properly routed not just inside a switch but between spine switches and leaf switches.

SGI has configured the default RDV address to 239.0.0.1 for the admin node used to install flat compute (service) nodes and leaders using UDPcast transport and pushing images for the first time to leaders. This address is used because the default 224.0.0.1 does not cross switch VLANs. The admin node needs to serve flat compute nodes across routed management networks.

For leader nodes serving SGI ICE compute nodes in tmpfs boot mode, the default is 224.0.0.1 since there is no VLAN crossing necessary.

Note: If you adjust this value, you may need to adjust the `udpcast-rexmit-hello-interval` attribute.

The `udpcast-mcast-rdv-addr` value will not take effect on leader nodes until an image is pushed (or re-pushed) from the admin node using the `cimage` command. The image push process is what re-configures Flamethrower and the node boot files on leader nodes.

Adjust this value using the `cadmin` command:

- See options `--set-udpcast-mcast-rdv-addr` and `--show-udpcast-mcast-rdv-addr`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.
- The admin node value is what is used by the admin node to serve flat compute and leader nodes using the UDPcast transport.
- This value is written to the network boot files for nodes being booted or installed with UDPcast as they need to match up with the server.

See the `udp-sender` man page for additional details.

udpcast-rexmit-hello-interval

The `udpcast-rexmit-hello-interval` attribute defines how often a UDP sender process will send a hello packet. This is especially important when the RDV address is not 224.0.0.1 (The admin node, for example, defaults to 239.0.0.1 for UDP sender processes.).

When a UDP receiver process starts for an RDV address other than 224.0.0.1, Linux will send an IGMP packet that is seen by the Ethernet switch. The Ethernet switch then updates its tables with this information allowing the multicast packets to properly route through the switch. The problem is that the UDP receiver sends its connection packet in many cases before the switch has had a chance to update the switch routing. As you read above, the UDP receiver waits forever for a UDPcast stream. If the request packet is not seen by the UDP sender on the admin node, perhaps because it was sent before the switch is set up to pass the packet, the UDP receiver could wait forever.

This value ensures that the UDP sender send a hello packet at regular intervals and then UDP receivers respond to it. In this way, even if the UDP receiver request was missed, the UDP receiver will send a fresh request after seeing a hello packet from the UDP sender.

SGI sets this value to 5000 (5 seconds) for UDP senders running on the admin node by default. The leader node UDP senders have this set to 0 (disabled) because it is not normally needed in that case since 224.0.0.1 is used for the RDV address and there are no VLANs being crossed.

If you need to change the RDV address used by leader nodes to serve tmpfs SGI ICE compute nodes, you should also adjust the `udpcast-rexmit-hello-interval` value so that the situation described earlier does not present a problem.

This value can be changed with `cadmin` command:

- See options `--set-udpcast-rexmit-hello-interval` and `--show-udpcast-rexmit-hello-interval`.
- The global value is what is used by leader nodes serving SGI ICE compute nodes in tmpfs mode.
- The admin node value is what is used by the admin node to serve flat compute and leader nodes using the UDPcast transport.

See the `udp-sender` man page for additional details.

Console Management

SGI clusters use the open-source console management package called `conserver`. For detailed information on `consever`, see the following:

http://www.conserver.com/

The conserver performs the following functions:

- Manages the console devices of all managed nodes in a cluster.
- A conserver daemon runs on the admin node and the rack leader controllers (RLCs). The admin node manages RLC and compute 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 admin node.
- All consoles are logged. These logs can be found at /var/log/consoles on the admin node and RLCs. An autofs configuration file is created to allow you to access RLC-managed console logs from the admin node, 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 admin node and the RLCs from the /opt/sgi/sbin/generate-conserver-files script on the admin node. 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 admin node.

Note: The conserver package replaces cconsole for access to all consoles (blades, RLCs, managed compute 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 2-16 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

SGI clusters use network time protocol (NTP) as the primary mechanism to keep the nodes in your cluster synchronized. This section describes this mechanism operates on the various cluster components and covers these topics:

Admin Node NTP

When you used the `configure-cluster` command, it guided you through setting up NTP on the admin node. The NTP client on the admin node 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 admin node 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 admin node when it is booted and then stays in sync by connecting to the admin node for time. The NTP server on the leader node provides NTP service to SGI ICE X components so that SGI ICE compute nodes can sync their time when they are booted. The RLC sends NTP broadcasts to some networks to keep the SGI ICE 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 compute nodes, SGI ICE 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.

Compute Node NTP

The NTP client on *managed* compute nodes sets its time at initial booting from the admin node. It listens to NTP multicast transmissions from the admin node to stay in sync. It does not provide any NTP service.

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

"`discover` Command" on page 49

SGI ICE Compute Node NTP

The NTP Client on the SGI ICE compute node sets its time at initial booting from the rack leader controller (RLC). It listens to NTP multicast transmissions 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 compute nodes may fail to get time from the admin node as they come online. SGI ICE 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 admin node and RLCs have the `time` service enabled (`xinetd`).
- All system node types have the `netdate` command.
- A special startup script is on RLC, compute, and SGI ICE 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 fixes the time when the NTP servers start offering time service.

Changing the Size of /tmp on SGI ICE Compute Nodes

This section describes how to change the size of /tmp on SGI ICE compute nodes.

Procedure 2-17 Increasing the /tmp Size

To change the size of /tmp on your system SGI ICE compute nodes, perform the following steps:

1. From the admin node, 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 SGI ICE 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 125.

Enabling or Disabling the SGI ICE 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 SGI ICE compute nodes and the rack leader controller (RLC) becoming unresponsive during the heavy write-out to the per-node iSCSI swap devices.

Procedure 2-18 Enabling the iSCSI Swap Device

If you wish to enable the iSCSI swap device in a given SGI ICE compute node image, perform the following steps:

1. Change root (`chroot`) into the SGI ICE compute node image on the admin node and enable the `iscsiswap` service, as follows:

```
# chroot /var/lib/systemimager/images/ice-sles11 chkconfig iscsiswap on
```

2. Then, push the image out to the racks, as follows:

   # **cimage --push-rack ice-sles11 r***

Procedure 2-19 Disabling the iSCSI Swap Device

To disable the iSCSI swap device in a SGI ICE compute node image where it is currently enabled, perform the following steps:

1. Disable the service, as follows:

```
# chroot /var/lib/systemimager/images/ice-sles11 chkconfig iscsiswap off
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack ice-sles11 r\*
```

Changing the Size of Per-node Swap Space

This section describes how to change per-node swap space on your SGI ICE X system.

Procedure 2-20 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 "Power-On/Off Management" on page 58).
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 "Power-On/Off Management" on page 58).

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 admin node. The images are in the `/var/lib/systemimager/images` directory. For more information on customizing these images, see "Power-On/Off Management" on page 58.

Switching SGI ICE Compute Nodes to a `tmpfs` Root

This section describes how to switch your system SGI ICE compute nodes to a `tmpfs` root.

Procedure 2-21 Switching SGI ICE Compute Nodes to a `tmpfs` Root

To switch your SGI ICE compute nodes to a `tmpfs` root, from the admin node perform the following steps:

1. To switch SGI ICE compute nodes to a `tmpfs` root, use the optional `--tmpfs` flag to the `cimage --set` command, for example:

```
adminadmin:~ # cimage --set --tmpfs ice-sles11 2.6.27.19-5-smp r1i0n0
```

Note: To use a `/tmpfs` root with the standard SGI ICE compute node image, the SGI ICE compute node needs to have 4GB of memory or above. A standard `/tmpfs` mount has access to half the system memory, and the standard SGI ICE compute node image is just over 1 GB in size.

2. You can view the current setting of a SGI ICE compute node, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: ice-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 ice-sles11 2.6.27.19-5-smp r1i0n0
```

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

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: ice-sles11 2.6.27.19-5-smp nfs
```

For help information, use the `cimage --h` option.

About Configuring Local Storage Space for Swap and Scratch Disk Space

You can configure the SGI ICE X system to support local storage space on SGI ICE compute nodes, which are also known as *blades*. Solid state drive (SSD) devices and 2.5" disks are available for this purpose. SGI supports a set of parameters that you can use to configure partitions on your system. You can define the size and status for both swap and scratch partitions.

You can set the partition values on a global basis or on an individual basis. If you set a value on a global basis, the value applies to all SGI ICE compute nodes. You can also set the value to apply to only one node. By default, the disks are partitioned only if blank; swap is off; scratch is set to occupy the whole disk space; and scratch is mounted at `/tmp/scratch`.

You can use the `cattr` command to retrieve the status of a setting, to enable a setting, or to disable a setting. If you do not set any parameters, the system uses the defaults.

The SMC `/etc/init.d/set-swap-scratch` script configures the swap and scratch space based on the settings you specify with the `cattr` command.

The following list explains the local storage space settings:

Setting **Effect**

`blade_disk_allow_partitioning`

Determines whether you can repartition and reformat the local storage disk. Specify `on` or `off`. Default is `on`.

To protect user data, SMC prevents you from repartitioning a disk that is already partitioned. In this case, you need a blank disk to use for the `swap` and `scratch` partitions.

`blade_disk_swap_status`

Determines whether SMC creates a swap partition on the on the local storage disk. Specify `on` or `off`. Default is `off`, which means that SMC does not create a swap partition.

SMC assigns the label `SGI_SWAP` when it partitions the disk. It enables the swap only if an `SGI_SWAP` label exists.

`blade_disk_swap_size`

Specifies the swap size, in megabytes. Specify one of the following values:

Value	Meaning
`-0`	Uses all free space when partitioning.
`0`	Does not create a swap partition on the local storage disk. Prevents SMC from creating a swap partition. Default.
`1`, `2`, ...	Specifies an integer number of megabytes for the swap partition.

`blade_disk_scratch_status`

Determines whether SMC creates a scratch partition on the on the local storage disk. Specify `on` or `off`. Default is `off`, which means that SMC does not create a scratch partition.

SMC assigns the label `SGI_SCRATCH` when it partitions the disk. It mounts the scratch on the partition labeled `SGI_SCRATCH`.

`blade_disk_scratch_size`

Specifies the scratch size, in megabytes. Specify either an integer number of megabytes or one of the special values, as follows:

Value	Meaning
`-0`	Uses all free space for scratch when partitioning. Default.
`0`	Does not create a scratch partition on the local storage disk. Prevents SMC from creating a scratch partition.
`1`, `2`, ...	Specifies an integer number of megabytes for the scratch partition.

`blade_disk_scratch_mount_point`

Specifies the mount point for the scratch partition. Default is `/tmp/scratch`.

You can mount the disk to any mount point. SMC creates the mount point directory if it does not already exist. SMC needs to have permission to create the mount point at the mount point you specify. On the SGI ICE compute nodes, the root mount point (`/`) is not writeable. If you want to mount to `/scratch`, make sure to create that folder as part of the SGI ICE compute node image.

`blade_disk_raid_level`

Specifies whether you can enable RAID0 when you have two disks for swap and scratch. The values are as follows:

Value	Meaning
`off`	Does not enable RAID0. Default.
`0`	Enables RAID0 for the swap and scratch partitions.

`blade_disk_reformat_swap_at_boot`

Specifies whether you are allowed to format the swap partition every time the SGI ICE compute node boots. The values are as follows:

Value	Meaning
`off`	Prevents formatting of the swap partition at boot. Default.
`0`	Enables formatting of the swap partition every time the SGI ICE compute node boots.

`blade_disk_reformat_scratch_at_boot`

Specifies whether you are allowed to format the scratch partition every time the SGI ICE compute node boots. The values are as follows:

Value	Meaning
`off`	Prevents formatting of the scratch partiton at boot. Default.
`0`	Enables formatting of the scratch partition every time the SGI ICE compute node boots.

The following topics show the `cattr` commands you can use to configure the swap and scratch disk space:

- "Retrieving the Current Status of a Local Storage Space Setting" on page 104
- "Enabling, Disabling, or Respecifying a Local Storage Space Setting" on page 105

Retrieving the Current Status of a Local Storage Space Setting

The following procedure explains how to display the status of a local storage space setting.

Procedure 2-22 To retrieve the status of a storage space setting

1. Log into the admin node as the root user.
2. Type the `cattr get` command, in the following format, to retrieve the current setting:

   `cattr get` *setting* [`-N` *node_id*] `--default` *default*

   - For *setting*, specify one of the local storage space settings. For the list of settings, see the following:

"About Configuring Local Storage Space for Swap and Scratch Disk Space" on page 101

- For *node_id*, specify the system ID for one SGI ICE compute node. Specify this argument only if you want to set one of the local storage space settings for an individual SGI ICE compute node.
- For *default*, specify the default value for this setting.

Example 1. The following command returns `on`, which indicates that the setting is enabled and applies to all SGI ICE compute nodes:

```
# cattr get blade_disk_allow_partitioning --default on
on
```

Example 2. Assume that you set the `blade_disk_scratch_size` to 2 megabytes. To retrieve the current scratch size, type the following command:

```
# cattr get blade_disk_scratch_size --default -0
2
```

Enabling, Disabling, or Respecifying a Local Storage Space Setting

The following procedure explains how to modify a local storage space setting.

Procedure 2-23 To enable, disable, or respecify a local storage space setting

1. Log into the admin node as the root user.
2. Type the `cattr set` command, in the following format, to enable, disable, or specify a value for a local storage space setting:

   `cattr set [-N` *node_id*`]` *setting value*

   - For *node_id*, specify the system ID for one SGI ICE compute node. Specify this argument only if you want to set one of the local storage space settings for an individual SGI ICE compute node.
   - For *setting*, specify one of the local storage space settings.
   - For *value*, specify `on`, `off`, an integer value that represents megabytes, or a mount point. For information about possible values, see the individual setting information in the following topic:

"About Configuring Local Storage Space for Swap and Scratch Disk Space" on page 101

3. Type the following `cimage` command:

   # **`cimage --push-rack`**

Example 1. The following command turns on the `blade_disk_allow_partitioning` setting for all SGI ICE compute nodes:

`cattr set blade_disk_allow_partitioning on`

Example 2. The following command turns on `blade_disk_allow_partitioning` for SGI ICE compute node `r1i0n0`:

`cattr set -N r1i0n0 blade_disk_allow_partitioning on`

Example 3. The following command sets the scratch partition mount point for the local disk associated with SGI ICE compute node `r1i0n0` to `/tmp/scratch22`:

`cattr set -N r1i0n0 blade_disk_mount_point /tmp/scratch22`

Using the `cattr` Command to Modify System Attributes

You can use the `cattr` command to assign attributes to cluster nodes. You can assign attributes either on a global basis, to the entire system, or on an individual node basis.

The `cattr` command is divided into operations that let you retrieve attribute settings, set attributes, remove atributes, and perform other functions.

Note: When possible, use the `cadmin` command, rather than the `cattr` command, to modify system attributes. When you use the `cadmin` command to modify an attribute, the `cadmin` command regenerates the configuration and eliminates the need for you to issue an `update-configs` command.

The following procedure explains this command sequence.

Procedure 2-24 To change attributes and propagate the changes to all nodes

1. Log into the admin node as the root user.
2. Use the `cattr` command to specify an attribute.

You can type the following command to retrieve the `cattr` command help statement and the list of attributes you can manipulate:

```
# cattr -h
```

Example 1:

```
 # cattr -h
Usage:
  cattr [--help] OPERATION [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 OPERATION --help'.
You have new mail in /var/mail/root
```

Example 2:

```
 # cattr get -h
Usage:
  cattr get [OPTION]... KEY

Options:
      --debug              enable debugging output
      --default=DEFAULT    print DEFAULT if no value is found
      --cascade            search multiple levels for a value
  -h, --help               print usage and exit
  -N, --node=NODE          print attribute for NODE
      --no-cascade         search only one level for the value
  -S, --self               print attribute for the current node

Details:
  If the attribute is unset (has no value) and the --default option is
  specified, the value given as DEFAULT will be echoed back as output.  This
  can reduce scripting conditionals.

  Similarly, the --cascade option will cause a search to occur from most
  to least specific: node, global, default.  The first value found is
```

```
  printed.  --cascade is on by default.

Examples:
  Get the global value of redundant_mgmt_network:
    % cattr get redundant_mgmt_network
quots
  Get the value of redundant_mgmt_network attached to r1lead:
    % cattr get --node r1lead redundant_mgmt_network1
```

You can modify one of the local storage space attributes in the following topic:

"About Configuring Local Storage Space for Swap and Scratch Disk Space" on page 101

About Disk Quotas

Within the compute image for an SGI ICE X rack leader controller (RLC), SGI sets default per-directory disk *quotas*, which can also be called *project quotas*. The quota mechanism prevents a disk from filling up and inhibiting the node's ability to boot.

Soft quotas and hard quotas apply to any entity that writes to disk, whether that be a user writing to disk actively or a user job that writes to disk. Quotas prevent a SGI ICE compute node from accidentally filling its RLC's disk space over the network file system (NFS). Quotas apply when a SGI ICE compute node is booted with NFS root directories, not `tmpfs` directories.

SGI sets default quota settings in each software image, rather than in each node. You can adjust these quota settings at your site. The soft quotas and hard quotas are as follows:

- A soft quota is an initial limit. After a SGI ICE compute node exceeds a soft quota, the SGI ICE compute node can continue to use resources up until it reaches the upper hard limit.
- A hard quota is a firm limit.

The default quotas are as follows:

- Soft quota = 2048 minutes
- Hard quota = 2148 minutes
- Quota timer = 1 day

If a hard quota is exceeded, or if a soft quota is exceeded past the time set in the timer, the SGI ICE compute nodes might fail to boot properly. The SGI ICE X system prevents additional writes to a disk when either of the following events occur:

- A disk reaches its hard limit.
- A disk reaches its soft limit and the timer has expired.

The hardware event tracker (HET) monitors the disk quota system. HET writes a message to the following log file when a quota limit is met:

`/var/log/het/het_trap_processor.log`

You can monitor the quota messages in the preceding log file, or you can configure HET to send an email notification to an email address or an email alias. For information about how to configure HET, see the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

The following topics provide more information about quotas:

- "Retrieving Quota Information" on page 109
- "Setting Quotas" on page 110
- "Viewing the SGI ICE Compute Node Read/Write Quotas" on page 112

Retrieving Quota Information

The following procedure explains how to retrieve quota values for a specific image.

Procedure 2-25 To retrieve quota values

1. Log into the admin node as the root user.
2. Type the following command to retrieve a list of the images on the system:

```
# cinstallman --show-images
Image Name              BT VCS Compat_Distro
ice-rhel6               1  1   rhel6
        2.6.32-504.el6.x86_64
lead-sles11sp3          0  1   sles11
        3.0.76-0.11-default
lead-rhel6.6            0  1   rhel6
        2.6.32-504.el6.x86_64
sles11sp3               0  1   sles11
```

```
        3.0.76-0.11-default
ice-sles11sp3            1  1   sles11
        3.0.76-0.11-default
```

The preceding example output shows two SGI ICE compute node images.

3. Type one of the following commands to retrieve information about one of the quotas or the quota timer:

   `cadmin --show-soft-quota --image` *image_name*
   `cadmin --show-hard-quota --image` *image_name*
   `cadmin --show-quota-timer --image` *image_name*

   For *image_name*, type one of the names from the `Image Name` column in the previous step.

   For example:

   ```
   # cadmin --show-soft-quota --image ice-sles11sp3
   2048m
   # cadmin --show-hard-quota --image ice-sles11sp3
   2148m
   # cadmin --show-quota-timer --image ice-sles11sp3
   1d
   ```

   The `cadmin` command output displays the quotas using the format of the underlying tool, which is the XFS file system project quota infrastructure. For information about the format, see the `xfs_quota`(8) man page.

4. (Optional) Set site-specific quotas.

   Proceed to the following:

   "Setting Quotas" on page 110

Setting Quotas

The following procedure explains how to change a quota or the quota timer.

Procedure 2-26 To set quotas

1. Log into the admin node as the root user.
2. Verify the current value for the quota setting you want to change.

For information about how to verify quota settings, see the following:

"Retrieving Quota Information" on page 109

3. Modify the quota setting.

- To set a site-specific *value*, use one of the following commands:

```
cadmin --set-soft-quota --image image_name value
cadmin --set-hard-quota --image image_name value
cadmin --set-quota-timer --image image_name value
```

For *image_name*, specify one of the names in the `Image Name` column of output from the `cinstallman --show-images` command. For information about the `cinstallman --show-images` command, see "Retrieving Quota Information" on page 109.

For *value*, specify an integer value followed by a unit specification, as follows:

– For `--set-soft-quota` or `--set-hard-quota` operations, specify `k` for kilobytes, `m` for megabytes, `g` for gigabytes, or `t` for terabytes.

– For the `--set-quota-timer` operation, specify `m` for minutes, `d` for days, `h` for hours, or `w` for weeks.

The following examples specify site-specific values for the quotas associated with the `ice-sles11sp3` compute image:

```
# cadmin --set-soft-quota --image ice-sles11sp3 4200m
# cadmin --set-hard-quota --image ice-sles11sp3 4196m
# cadmin --set-quota-timer --image ice-sles11sp3 3d
```

- To reset a site-specific value back to the SGI default value, use one of the following commands:

```
cadmin --unset-soft-quota --image image_name
cadmin --unset-hard-quota --image image_name
cadmin --unset-quota-timer --image image_name
```

The following examples reset site-specific values back to the SGI default values:

```
# cadmin --unset-soft-quota --image ice-sles11sp3
# cadmin --unset-hard-quota --image ice-sles11sp3
# cadmin --unset-quota-timer --image ice-sles11sp3
```

4. Push out the changes to the SGI ICE compute nodes.

   Perform the following procedure:

   "Pushing System Images from the Admin Node" on page 115

Viewing the SGI ICE Compute Node Read/Write Quotas

You can retrieve the per-compute-node read and write quota values.

The following procedure explains how to retrieve current usage.

Procedure 2-27 To view the SGI ICE compute node read/write quota

1. Log into the admin node as the root user.
2. Use the `ssh`(1) command to log into one of the rack leader controllers (RLCs).

   To retrieve a list of RLCs, type `cnodes --leader`.

   The following example shows how to retrieve a list of RLCs and how to log into one of them:

   ```
   # cnodes --leader
   r1lead
   r2lead
   # ssh r1lead
   ```

3. Type the following command to retrieve a list of projects:

   ```
   # less /etc/projects
   1:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n0
   2:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n1
   3:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n2
   4:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n3
   5:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n4
   6:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n5
   7:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n6
   8:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n7
   9:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n8
   10:/var/lib/sgi/per-host/ice-rhel6.5/1/i0n9
   . . .
   ```

   The project numbers are the left-most integers in the output. Type `q` to exit the `less`(1) command.

4. Use the xfs_quota(8) command, in the following format, to retrieve the current usage values:

   xfs_quota -x -c 'quota -ph *project_num*'

   For *project_num*, specify one of the project numbers you retrieved in the preceding step.

   For example:

```
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 [------] /
```

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 a 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
```

Backing up and Restoring the System Database

SMC captures the relevant data for the managed objects in a cluster. The system database is critical to the operation of your cluster and you need to back up the database on a regular basis.

Managed objects on a cluster include the following:

- The cluster itself

  The whole cluster is a managed object. An SGI ICE X system is modeled as a meta-cluster. This meta-cluster contains the racks each modeled as a sub-cluster.

- Nodes

  Admin node, rack leader controllers (RLCs), compute nodes, SGI ICE 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 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.

The following procedures explain how to back up and restore the system database.

Procedure 2-28 To back up the system database

1. Log into the admin node as the root user.
2. Type the following command:

   `mysqldump --opt -p'cat /etc/odapw' oscar > `*file*`.sql`

   For *file*, type a name for the database backup file.

   The `mysqldump`(1) command reads the password from file `/etc/odapw`.

   For example:

   ```
   # mysqldump --opt -p'cat /etc/odapw' oscar > oscar-db-backup.sql
   ```

Procedure 2-29 To restore the system database

1. Log into the admin node as the root user.

2. Type the following command:

   ```
   mysql -u root -p'cat /etc/odapw' oscar < file.sql
   ```

   For *file*, type the name you gave to the database backup file when you backed it up.

   For example:

   ```
   # mysql -u root -p'cat /etc/odapw' oscar < oscar-db-backup.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 limits EDNS logging. This section describes how to delete this code and allow EDNS to work unrestricted and log messages.

Procedure 2-30 Enabling EDNS

To enable EDNS on your cluster, 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; };  };
   ```

Pushing System Images from the Admin Node

The admin node can host multiple forms of system images for the other cluster nodes. For example, you can have some production images and some test images, and you can push the images to the nodes as needed. Many system operations refer to this procedure because you need to update system images and push out new images as part of several system administration tasks.

The following procedure explains how to push SGI ICE compute node system images to all SGI ICE compute nodes or to a set of SGI ICE compute nodes.

Procedure 2-31 To push software system images

1. Type the following command to stop the SGI ICE compute nodes:

   # **cpower node halt "r*i*n*"**

   The preceding command stops all SGI ICE compute nodes. Use the preceding command when you need to push an updated image out to all nodes.

2. (Optional) Provide information about the number of racks on your system.

   Perform this step if you have a small system with fewer than eight IRUs per RLC.

   The procedure pushes the updated compute image to all the SGI ICE compute nodes. This process can run for a long time on large systems. If you have a large number of IRUs, you need the system to perform expansions that enable you to change many SGI ICE compute nodes at a time. If you have fewer than eight IRUs per RLC, however, the expansions are not needed.

   The following substeps explain how to prepare an SGI ICE X system to work on a smaller number of SGI ICE compute nodes:

   - Type the following command to retrieve the identifiers for the RLCs on your system:

     # **cnodes --leader**

   - Type the following command one or more times to suppress unnecessary processing:

     `cadmin --set-max-irus --node` *rlc_id number_of_racks*

     For *rlc_id*, specify the identifier for one of the RLCs in your system.

     For *number_of_racks*, specify the number of IRUs associated with this RLC.

   For example, the following command specifies that there is only one IRU associated with the RLC identified as `r1lead`:

   # **cadmin --set-max-irus --node r1lead 1**

3. Use the `cimage` command, in the following format, to push the changes:

   `cimage --push-rack compute-`*image_name rack*

For *image_name*, specify the name of the SGI ICE compute node image that you updated.

For *rack*, specify the nodes. To specify all SGI ICE compute nodes, specify `r\*` or `r*i*n*`. To specify only selected nodes, specify `r`*x*`i`*x*`n`*x*, and substitute specific integer numbers for the *x* characters.

For example, the following command pushes changes to all the SGI ICE compute nodes:

```
# cimage --push-rack ice-rhel6.5 r\*
```

4. Type the following command to power-up the SGI ICE compute nodes:

```
# cpower node on "r*i*n*"
```

Managing Software Images

This chapter includes the following topics:

Overview of Image Management on SGI Clusters

SGI clusters include different types of nodes, and there is a unique software image for each individual node type. When you install additional software on your cluster, you might need to modify the software on some of the nodes.

Figure 3-1 on page 120 and Figure 3-2 on page 121 show example cluster configurations and the software images that reside on each node at the time the node is shipped from the SGI factory.

Figure 3-1 on page 120 shows a simple SGI Rackable cluster, and the software images that reside on each node are noted. The admin node hosts an image called `rhel6.6`, which is the default image for the compute nodes in the cluster.

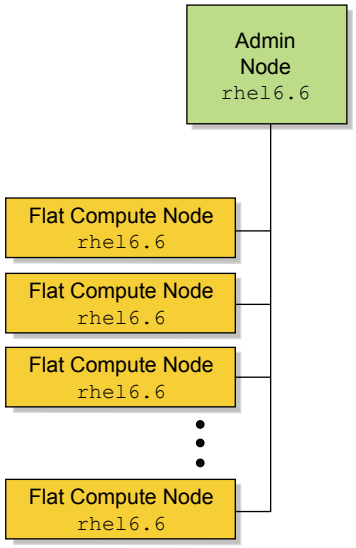

Figure 3-1 SGI Rackable Cluster — Node Software Images Noted

Figure 3-2 on page 121 shows a simple SGI ICE X cluster, and the software images that reside on each node are noted. The admin node hosts the following images, which are the default images for the other nodes in the cluster:

- `lead-rhel6.6` is the default image for the rack leader controller (RLC).
- `rhel6.6` is the default image for the compute nodes.
- `ice-rhel6.6` is the default image for the SGI ICE compute nodes.

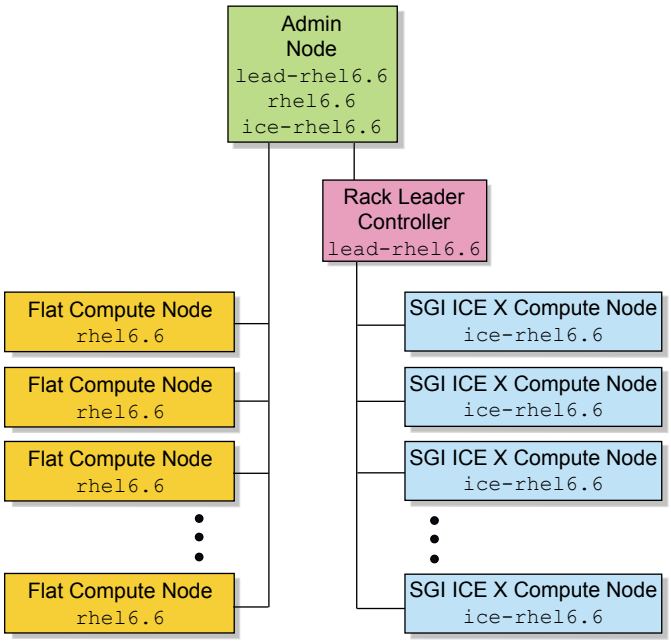

Figure 3-2 SGI ICE X Cluster — Node Software Images Noted

Figure 3-1 on page 120 and Figure 3-2 on page 121 show that all cluster nodes subordinate to the admin node have the same operating system. This is for convenience only and is not a requirement. The remainder of this chapter describes accessing the default software images, how to update them, and how they are pushed to the various nodes of the cluster.

Image Management Commands

SGI cluster software includes software images for each type of node: admin, rack leader, compute, and SGI ICE compute. You might need to add site-specific software, or you might need to change the default settings or services within a particular image. In addition to general Linux commands, SMC includes the following commands for image management:

- "`crepo` Command" on page 122

`crepo` Command

You can use the `crepo` command to manage software in the SMC, SGI Foundation, SGI Performance Suite, and the Linux distribution(s) repositories that you are using on your system. You can also use the `crepo` command to manage any custom repositories you create yourself or to add additional media.

Each repository has associated with it a name, directory, update URL, selection status, and suggested package lists. The `sync-repo-updates` command uses the update URL. For RHEL-based systems, make sure the system is subscribed as `rhel-x86_64--server-6`.

The directory is where the actual `yum` repository exists is one of the following:

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 deselected. Usually, the commands ignore deselected 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 RPM lists. For example:

```
# crepo --select SUSE-Linux-Enterprise-Server-11-SP3
Updating: /etc/opt/sgi/rpmlists/generated-ice-sles11sp3.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-sles11sp3.rpmlist
```

When generating the RPM lists, the `crepo` command combines 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 SP3 compute nodes is read from `/var/opt/sgi/sgi-repodata/SGI-Tempo-2.9.0-sles11sp3/tempo-ice-compute.rpmlist`.

The suggested `rpmlist`s can be overridden by creating an override `rpmlist` in the `/etc/opt/sgi/rpmlists/override/` directory. For example, to change the default RPM list, create file `/etc/opt/sgi/rpmlists/override/SGI-Tempo-2.9.0-sles11sp3/tempo-ice-compute.rpmlist`.

Specifically, the software looks for `/etc/opt/sgi/rpmlists/generate-*.rpmlist` and creates an image for each `rpmlist` that matches. SMC determines the default image to use for each node type by hard-coding `$nodeType-$distro` as the type, where `distro` is the admin node's `distro` and `nodeType` is `compute`, `ice-compute`, `leader`, 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. The directory contents are as follows:

```
admin:/etc/opt/sgi/rpmlists # ls
compute-minimal-sles11sp1.rpmlist  generated-lead-rhel6.5.rpmlist
generated-ice-rhel6.5.rpmlist  generated-rhel6.5.rpmlist
```

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

You can use the crepo --show command to show the available repositories on the admin node, as follows:

```
sys-admin:~ # crepo --show
* SGI-Foundation-Software-2.10-rhel6 : /tftpboot/sgi/SGI-Foundation-Software-2.10-rhel6
* SGI-Accelerate-1.8-rhel6 : /tftpboot/sgi/SGI-Accelerate-1.8-rhel6
* SGI-Tempo-2.9.0-rhel6 : /tftpboot/sgi/SGI-Tempo-2.9.0-rhel6
* SGI-MPI-1.8-rhel6 : /tftpboot/sgi/SGI-MPI-1.8-rhel6
* Red-Hat-Enterprise-Linux-6.5 : /tftpboot/distro/rhel6.5
```

For a crepo command usage statement, type the following command:

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

cinstallman Command

The cinstallman command is a wrapper tool for several operations. You can use the cinstallman command to complete the following tasks:

- 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)

- Formally track revisions to images.
- 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)

For a cinstallman command usage statement, type the following:

```
admin:~ # cinstallman --help
```

In the following example, the --refresh-node operation is used to ensure the online managed compute 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/sles11.rpmlist
```

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, type the following command:

```
admin:~ # cimage --help
```

EXAMPLES

Example 3-1 cimage Command Examples

The following examples walk you through some typical cimage command operations.

To list the available images and their associated kernels, perform the following:

```
# cimage --list-images
image: ice-sles11
        kernel: 2.6.27.19-5-carlsbad
        kernel: 2.6.27.19-5-default
image: ice-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: ice-sles11 2.6.27.19-5-default nfs
r1i0n8: ice-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 `ice-sles11` image, perform the following:

```
# cimage --set ice-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: ice-sles11 2.6.27.19-5-smp
r1i0n1: ice-sles11 2.6.27.19-5-smp
r1i0n2: ice-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 `ice-sles11` image, perform the following:

```
# cimage --set ice-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 ice-sles11 2.6.27.19-5-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the `ice-sles11` image to a new image (so that you can modify it) , perform the following:

```
# cinstallman --create-image --clone --source ice-sles11 --image mynewimage
Cloning ice-sles11 to mynewimage ... done
```

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

To change to the cloned image created in the example, above, copy the needed RPMs into the `/var/lib/systemimager/images/mynewimage/tmp` directory, use the

`chroot` command to enter the directory and then install the RPMs, perform the following:

```
# cp *.rpm /var/lib/systemimager/images/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: ice-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 admin node and all compute nodes in all racks, perform the following:

```
# cimage --del-image mynewimage
r1lead: delete-image: mynewimage
```

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

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 non-ICE compute/service nodes, Intel Phi (MIC) nodes and
 --compute              all non-ICE compute/service nodes
 --compute-mic          all Intel Phi (MIC) nodes that are hosted by non-ICE compute/service nodes
 --ice-compute          all ICE compute nodes
 --ice-compute-mic      all Intel Phi (MIC) nodes that are hosted by ICE compute nodes
 --leader               all leader 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.
```

Example 3-2 cnodes Example

The following examples show how to display the nodes on your cluster.

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 SGI ICE compute nodes, perform the following:

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

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

```
[admin ~]#  cnodes --compute
service0
```

Retrieving the List of Supported Distributions (Distros)

To find a list of operating system distributions that the cluster nodes support, log into the admin node and type the following command:

```
# ls /opt/sgi/share/rpmlists/distro/
```

The output from the preceding command shows the software distributions that SGI supports on the various non-admin nodes. In the output, the RPM lists that pertain to each node type are represented as files that have the following prefixes:

```
ice-compute-
lead-
```

no prefix (for compute nodes)

SGI supports the RHEL 6 and SLES 11 operating systems on the admin node, on the RLCs, on the compute nodes, and on the SGI ICE compute nodes. On the compute nodes and on the SGI ICE compute nodes, SGI also supports RHEL 5 and SLES 10.

Changing the Services on the SGI ICE Compute Nodes

To improve the performance of applications running MPI jobs on SGI ICE compute nodes, most services are disabled by default in SGI ICE compute node images. The following procedure explains how to obtain information about the services that run on SGI ICE compute nodes and how to change the list of active services.

Procedure 3-1 To change the services on an SGI ICE compute node

1. Log into the admin node as the root user.

2. Type the following command to change to the directory where the compute images reside:

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

3. Type the `ls`(1) command, and examine the available compute images.

   For example:

```
# ls -F
ACHTUNG            DO_NOT_TOUCH_THESE_DIRECTORIES  sles11sp3
ice-sles11sp3  lead-sles11sp3
CUIDADO            README
```

4. Type the `cd`(1) command in the following format to change to the directory that hosts the compute image services file:

   `cd` *image_name*`/etc/opt/sgi/conf.d`

   For *image_name*, specify one of the compute image names.

   For example:

   # **cd ice-sles11sp3/etc/opt/sgi/conf.d**

5. Use a text editor or text viewer to display the services file.

   For example:

   # **less 80-compute-distro-services**

6. Peruse the file and decide if you want to change the settings for any of the services.

   If you want to change any services, complete the rest of this procedure.

7. Type the following command to copy the original services file:

   # **cp 80-compute-distro-services 80-compute-distro-services.local**

   This step shows to make a copy of the original file. Always edit a copy, not the original.

8. Open the `.local` services file from within a text editor and change the services as needed.

For example:

```
# vi 80-compute-distro-services.local
```

9. Save and close the `.local` services file.
10. Type the following command to propagate the new services file to the other SGI ICE compute nodes:

```
# cimage --push-rack
```

After this command runs, the configuration framework executes the `.local` version of the services file, and it skips the other, original file. For more information on making adjustments to configuration framework files, see "SGI ICE X System Configuration Framework" on page 143.

For more information on making adjustments to configuration framework files, see "SGI ICE X System Configuration Framework" on page 143.

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 SGI ICE compute and compute nodes. It describes how to augment software packages. Many tasks having to do with package management have multiple valid methods to use.

Performing SGI ICE Compute Node Per-Host Customizations

You can add per-host SGI ICE 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 admin node.

Note: When creating custom images for SGI ICE compute nodes, make sure you clone the original SGI images. You can fall back to the original images if necessary.

Scripts in the global directory apply to all SGI ICE 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).

Also see the README file on the admin node at /opt/sgi/share/per-host-customization/README.

SGI provides an example global script in the following file:

/opt/sgi/share/per-host-customization/global/sgi-fstab.sh

Customizing Software Images

Note: Procedures in this section describe how to work with compute node and SGI ICE compute node images. Always use a cloned image. If you are adjusting an RPM list, use your own copy of the RPM list. This section also describes how to clone an image.

The compute and SGI ICE 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 an SGI ICE compute node image, or you can create a new image based on an RPM list. For compute images, you can either clone the image and work on a copy or you can always make a new SGI ICE compute node image from the default RPM list supplied by SGI.

Procedure 3-2 Creating a Simple SGI ICE Compute Node Image Clone

To create a simple SGI ICE compute node image clone from the admin node, perform the following steps:

1. To clone the SGI ICE compute node image, perform the following:

```
# cinstallman --create-image --clone --source ice-sles11 --image ice-sles11-new
```

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

```
# cimage --list-images
image: ice-sles11
       kernel: 2.6.27.19-5-smp

image: ice-sles11-new
       kernel: 2.6.27.19-5-smp
```

3. To push the SGI ICE compute node image out to the rack, perform the following:

   # **cimage --push-rack ice-sles11-new r***

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

   # **cimage --set ice-sles11-new 2.6.27.19-5-smp "r*i*n*"**

Procedure 3-3 Manually Adding a Package to an SGI ICE Compute Node Image

To manually add a package to an SGI ICE 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 SGI ICE 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 124.

---

1. Make a clone of the SGI ICE compute node image, as described in "Customizing Software Images" on page 132.

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

```
# cimage --list-images
  image: ice-sles11
         kernel: 2.6.27.19-5-smp

  image: ice-sles11-new
         kernel: 2.6.27.19-5-smp
```

3. From the admin node, change directory to the images directory, as follows:

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

4. From the admin node, copy the RPMs you wish to add, as follows, where `ice-sles11-new` is your own SGI ICE compute node image:

   # **cp /tmp/newrpm.rpm ice-sles11-new/tmp**

5. The new RPMs now reside in `/tmp` directory in the image named `ice-sles11-new`. To install them into your new SGI ICE compute node image, perform the following commands:

   # **chroot ice-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 admin node 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 ice-sles11-new r***

   This will push the updates to the rack lead controllers and the changes will be seen by the SGI ICE compute nodes the next time they start up. For information on how to ensure the image is associated with a given node, see the `cimage --set` command and the example in Procedure 3-2, page 132.

Procedure 3-4 Manually Adding a Package to the Compute Node Image

To manually add a package to the compute 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 SGI ICE 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 124.

---

1. Use the `cinstallman` command to create your own version of the compute node image. See "`cinstallman` Command" on page 124.

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

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

3. From the admin node, copy the RPMs you wish to add, as follows, where `my-service-image` is your own compute 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 compute 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 SGI ICE compute node images, changes made to a compute node image will not be seen by compute 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.

Using `cinstallman` to Install Packages into Software Images

The packages that make up the SGI Foundation Software, 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 work 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 compute node image so that the next time you image or install a compute node, it will have this new package.

```
# cinstallman --yum-image --image my-sles11 install zlib-devel
```

You can perform a similar operation for SGI ICE compute node images. Note the following:

- If you update a SGI ICE compute node image on the admin node, you have to use the `cimage` command to push the changes. For more information on the `cimage` command, see "`cimage` Command" on page 125.
- If you update a compute node image on the admin node, that compute 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 compute node to reimage itself with a specified image the next time it boots.

Using `yum` to Install Packages on Running Compute Nodes or Rack Leader Controllers (RLCs)

Note: These instructions only apply to managed compute nodes and RLCs. They do not apply to SGI ICE compute nodes.

You can use the `yum` command to install a package on a compute node. From the admin node, you can issue a command similar to the following:

```
# cinstallman --yum-node --node service0 install zlib-devel
```

Note: To get all compute nodes, replace `service0` with `service\*`.

For more information on the `cinstallman` command, see "`cinstallman` Command" on page 124.

Creating SGI ICE Compute and Compute Node Images Using the `cinstallman` Command

You can create compute node and SGI ICE compute node images using the `cinstallman` command. This command generates a root directory for images automatically. Fresh installations of the SMC software create these images during the `configure-cluster` installation step.

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/ice-sles11.rpmlist` (see "`crepo` Command" on page 122). You should **NOT** edit the default lists. These default files are recreated by the `crepo` command when repositories are added or removed. Therefore, you should only use the default RPM lists as a model for your own.

Note: The procedure below uses SLES.

Procedure 3-5 Using 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 example compute node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/sles11.rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.
3. Use the `cinstallman` command with the `--create-image` option to create the images root directory, as follows:

```
# cinstallman --create-image --image my-service-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

   This example uses `my-service-node-image` as the home/name of the image.

   Output is logged to `/var/log/cinstallman` on the admin node.

4. After the `cinstallman` command finishes, the image is ready to be used with compute nodes. You can supply this image as an optional image name to the `discover` command, or you may assign an existing compute node to this image using the `cinstallman --assign-image` command. You can tell a compute node to image itself at the next reboot by using `the cinstallman --next-boot` option.

Procedure 3-6 Use the `cinstallman` Command to Create an SGI ICE Compute Node Image

To create an SGI ICE compute node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the SGI ICE compute node image RPM list and work on the copy, as follows:

   # cp /etc/opt/sgi/rpmlists/ice-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 `my-compute-node-image` as the name.

   Output is logged to `/var/log/cinstallman` on the admin node.

   The `cinstallman` command makes the new image available to the `cimage` command.

4. For information on how to use the `cimage` command to push this new image to rack leader controllers (RLCs), see "`cimage` Command" on page 125.

Re-Installing a Compute Node with a Non-Default Image

The following example shows how to reinstall an already discovered compute node with a new image:

```
# cinstallman --assign-image --node service2 --image my-service-node-image --kernel 3.0.76-0.11-default
# 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 49. For more information on the `cinstallman` command, see "`cinstallman` Command" on page 124.

Retrieving a Compute Node Image from a Running Compute Node

To retrieve a compute node image from a running compute node, perform the following steps:

1. As **root user**, log into the compute node from which you wish to retrieve an image. You can use the `si_prepareclient`(8) program to extract an image. Type the following command to start the program:

```
service0:~ # si_prepareclient --server admin
Welcome to the SystemImager si_prepareclient command. This command may modify
the following files to prepare your golden client for having its image
retrieved by the imageserver.  It will also create the /etc/systemimager
directory and fill it with information about your golden client. All modified
files will be backed up with the .before_systemimager-3.8.0 extension.

 /etc/services:
   This file defines the port numbers used by certain software on your system.
   Entries for rsync will be added if necessary.

 /tmp/filetlOeP5:
   This is a temporary configuration file that rsync needs on your golden client
   in order to make your filesystem available to your SystemImager server.

 inetd configuration:
   SystemImager needs to run rsync as a standalone daemon on your golden client
   until its image is retrieved by your SystemImager server.  If rsyncd is
   configured to run as a service started by inetd, it will be temporarily
   disabled, and any running rsync daemons or commands will be stopped.  Then,
   an rsync daemon will be started using the temporary configuration file
   mentioned above.

See "si_prepareclient --help" for command line options.

Continue? (y/[n]):
```

Enter **y** to continue. After a few moments, you are returned to the command prompt. You are now ready to retrieve the image from the admin node.

2. Exit the **service0** node, and as **root user** on the admin node, perform the following command: (Replace the image name and compute 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 compute 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 --kernel 3.0.76-0.11-default
# cinstallman --next-boot image --node service0
```

6. Reboot the compute node.

Using a Custom Repository for Site Packages

You can maintain software packages specific to your site and have them available to the `crepo` command. SGI recommends that you put site-specific packages in a separate location. They should not reside in the same location as SGI or operating system packages.

For information about the `crepo` command, see the following:

"`crepo` Command" on page 122

The following procedure explains how to create a custom repository.

Procedure 3-7 To create a custom repository for site—specific software packages

1. Log into the admin node as the root user, and create a directory for your site-specific packages on the admin node.

   For example:

   # **mkdir -p /tftpboot/site-local/site-rpms**

2. Verify that the site-specific software package you want to add to the repository is in an accessible directory.

   Download the software if you have not already done so.

3. Copy the site-specific software package to the new directory.

   For example:

   # **cp site-package-1.0.rpm /tftpboot/site-local/Site-RPMs**

4. Use the `crepo` command in the following format to create a site-specific repository and add the contents of the new directory to the repository:

   `crepo--add` *directory_for_site-specific_packages* `--custom '`*site-specific_repository*`'`

   For *directory_for_site-specific_packages*, specify the directory you specified in step 1.

   For *site-specific_repository*, create a name for a new site-specific repository.

   This command also creates the `yum` and `repomd` metadata.

   For example:

   # **crepo --add /tftpboot/site-local/site-rpms --custom 'Site-RPMs'**

5. Use `crepo` command in the following format to make the new site-specific repository available to the `cinstallman` command:

   `crepo --select` *site-specific_repository*

   For example:

   # **crepo --select Site-RPMs**

6. (Optional) Add the new RPM base names to an existing RPM list.

   This step makes your site-specific RPMs available by default when you create new node images in the future.

The substeps are as follows:

- Use the cp(1) command to copy an existing generated RPM list.
- Open the new RPM list file with a text editor. That is, open the copy.
- Add each new RPM as line in the file.
- Save and close the file.

For example, assume that you want to add the following site-specific RPMs to the RPM list called generated-rhel6.5.rpmlist:

```
kernel-debug-debuginfo-2.6.32--431.el6.x86_64.rpm
kernel-debuginfo-2.6.32--431.el6.x86_64.rpm
kernel-debuginfo-common-x86_64--2.6.32--431.el6.x86_64.rpm
```

Complete the following steps:

- Type the following commands:

```
# cp /etc/opt/sgi/rpmlists/generated-rhel6.5.rpmlist \
/etc/opt/sgi/rpmlists/site-rhel6.5.rpmlist
# vi /etc/opt/sgi/rpmlists/site-rhel6.5.rpmlists
```

- Use the vi(1) editor to add the following lines to file site-rhel6.5.rpmlists:

  ```
  kernel-debug-debuginfo
  kernel-debuginfo
  kernel-debuginfo-common
  ```

- Save and close the file.

7. Use the cinstallman command to install the new packages into an image, onto a node, or into a new image that contains these packages.

   - To install the new packages into an existing image, use the following format:

     cinstallman --yum-image --image *image* install *package package* ...

     For *image*, specify the image into which you want to install the packages.

     For *package*, specify one or more of the packages you wrote to the repository.

For example:

```
# cinstallman --yum-image \
--image ice-rhel6.5 install kernel-debuginfo kernel-debug-debuginfo kernel-debuginfo-common
```

If necessary, type the `cimage --list-images` command to retrieve a list of existing images.

- To install the new packages onto a running node, use the following format:

  `cinstallman --yum-node --node` *node_ID package package* `...`

  For *node_ID*, specify the node ID of a compute node or an SGI ICE compute node.

  For example:

```
# cinstallman --yum-node \
--node service0 install kernel-debuginfo kernel-debug-debuginfo kernel-debuginfo-common
```

- To create a new image that includes the packages, use the following format:

  `cinstallman --create-image --image` *new_image* `--rpmlist` *path*

  For *new_image*, specify a name for the new image.

  For *path*, specify the full path to the new image.

  For example:

```
# cinstallman --create-image --image my-image \
--rpmlist /etc/opt/sgi/rpmlists/site-rhel6.5.rpmlists
```

8. (Conditional) Push the changes to the SGI ICE compute nodes.

   Perform the following procedure if the image you created or updated was for an SGI ICE compute node:

   "Pushing System Images from the Admin Node" on page 115

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 SGI ICE compute node images.
- The SGI ICE system configuration framework can be used with all node types.

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 framework operate?

These files could be added, for example, to a running compute node, or to an already created service or compute image. Remember that images destined for SGI ICE compute nodes need to be pushed with the `cimage` command after being altered. For more information, see "`cimage` Command" on page 125.

- 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 configuration 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 run only if the node in question is running that distro. For example, `/etc/opt/sgi/conf.d/99-foo.sles` runs only when the node is running `sles`. This example shows the order of operations.

  If you had `88-myscript.sles11`, `88-myscript.sles`, and `88-myscript`:

  – On a `sles11` system, `88-myscript.sles11` runs.

  – On a `sles` system that is not `sles11`, `88-myscript.sles` runs.

  – On all other distros, `88-myscript` runs.

- If you want to make a custom version of a script supplied by SGI, change the script's suffix from the distro name to `.local`. The `.local` suffix indicates that you want the local version to run in place of the one supplied by SGI. This naming convention allows you to customize the scripts provided by SGI while preserving the original, default script supplied by SGI. Scripts that end in `.local` have the highest precedence. In other words, if you had `88-myscript.sles` and `88-myscript.local`, then `88-myscript.local` runs in all cases and any other `88-myscript.`*suffix* scripts never run.

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.

When is the framework called?

- The framework is called when an image is created.
- The framework is also called when the admin node, RLC, or compute 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 an SGI ICE compute node root image to RLCs, the configuration framework executes within the `chroot` of the SGI ICE compute node image after it is pulled from the admin node to the RLC.
- The framework is called when a compute node or an RLC node is installed.

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 the framework is called then, too. The second pass installs the other hundreds of RPMs. The framework is called for the first time before some packages are installed, and the framework is called 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. The `sgi-cluster` package ensures that this setting is correct. Anytime `yum` is installed or updated, it verifies that `plugins=1` is set.

How does yume work?

- `yume`, an oscar wrapper for `yum`, works by creating a temporary `yum` configuration file in `/tmp` and then points `yum` at it. This temporary configuration file needs to have plugins enabled. A tiny patch to `yume` makes this happen. This fixes it for `yume` and also `mksiimage`, which calls `yume` as part of its operation.

Cluster Configuration Repository: Updates on Demand

The SGI ICE X system includes a cluster configuration repository/update framework. This framework generates and distributes configuration updates to admin node, rack leader controller (RLC), and compute nodes in the cluster. Some of the configuration files managed by this framework include C3 conserver, DNS, Ganglia, hosts files, and NTP.

When an event occurs that requires these files to be updated, the framework executes on the admin node. The admin node stores the updated configuration framework in a special cached location and updates the appropriate nodes with their new configuration files.

In addition to the updates happening as required, the configuration file repository is consulted when an admin node, RLC, or compute 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.

Using the SMC Version Control System

The SMC version control system (VCS) formalizes the archiving, tracking, and otherwise management of the various versions of an image you might create. The SMC implementation of VCS uses the `cinstallman` command. To make the process fast and use disk space efficiently, `rsync` is used to store the revisions of images using the `--link-dest` feature.

The following topics describe how to use VCS to manage system images:

When to Use VCS

Node-specific software resides on the admin nodes, and when cluster software is installed and configured, the installer pushes the node-specific software to each node in the cluster. Over time, you might need to modify these software images. For example, you might need to add a workload manager or file system software. Before you add additional software, SGI recommends that you back up the original, default software images. Over time, if you modify the images frequently, your image repository might contain several different version and become more difficult to manage. As an alternative to managing these images manually, you can use VCS.

The following terminology pertains to the image files:

- The *working copy* of an image is the copy that is stored on the admin node in `/var/lib/systemimager/images/`*image_name*. The *image_name* directory contains additional subdirectories and files, all of which comprise the system image. The format for the *image_name* directory's name is one of the following:
  - *os_name*. For example, `rhel6.6`. This is the name of the image that can reside on the flat compute nodes in the cluster.
  - `lead-`*os_name*. For example, `lead-rhel6.6`. This is the name of the image that can reside on the rack leader controllers in the cluster.
  - `ice-`*os_name*. For example, `ice-rhel6.6`. This is the name of the image that can reside on the SGI ICE compute nodes in the cluster.

  When you install cluster software, the installer pushes the working copy image from the admin node to the appropriate nodes in the cluster. This is also where you make changes such as editing files, updating or installing RPMs, etc.
- A *committed copy* of an image is a copy that resides in the VCS repository. It is best to check in (commit) copies of images as you modify them to ensure that modifications are not lost.

VCS Repository

Versions of the image are stored in a special location on the filesystem using `rsync`. When you commit or revert an image, rsync is used to transfer the data as needed to or from the VCS repository.

Note: You should never modify files within the version control system repository. If you were to edit files in the VCS repository, the integrity of VCS would be compromised.

Managing New Images

When you create a new image (See "`cinstallman` Command" on page 124.), it resides in `/var/lib/systemimager/images/`*image_name*. Once that is created, SMC sends a copy is sent to the VCS repository and the revision number is set to 1.

Managing Clones

With VCS, cloning works like creating a new image (described in the preceding section). When cloning, you may optionally specify a revision of the image to be the source for the clone. See the `--rev` option.

Committing the Working Copy

After making changes to the working copy of an image, `/var/lib/systemimager/images/`*image_name*, you can commit your changes in to VCS with the `cinstallman --commit` operation. The commit also requires you to enter a log message. This may be specified with `--msg`, or it will be read from the terminal.

Reverting the Working Copy to a Specified Revision

If you wish to revert the working copy of an image to a specified revision, use `cinstallman --revert`. The working copy of the image will be removed and replaced by a copy of the revision you specify from the VCS repository.

Reviewing Revision History

Each time a commit is done, a log message is added with the associated change. You can use the `cinstallman --history` command to list the revision history of an image. You can optionally specify a revision range.

Reviewing File-Level Changes Between Revisions and the Working Copy

If you wish to see what has changed in the image, you can use `cinstallman --changed`.

- When no revision is specified, the working copy (`/var/lib/systemimager/images/image_name`) will be compared to the highest version checked in to VCS.
- If a single revision is supplied, then the working copy is compared to the specified revision.
- If a revision range is specified, the changes between those two revisions are output.
- By default, a special mode of `rsync` is used to list the changed files. Preceding each file in the list of changed files is an 11–character summary of the differences. See the examples in later section "Adding a Revision and Querying Changes" on page 151. For an explanation of the 11–character summary, see the description of the `itemize-changes` option of the `rsync` man page.

  Using the `--cmp-tool` parameter, you can switch to using `diff` in brief mode to summarize changes instead.

Reviewing File-Content Differences Between Versions and the Working Copy

This is the same concept as described in the preceding section, except `diff` output is printed instead of changed files. You can optionally target a specific file with the `--file` option. You can use a custom `diff` tool in place of `diff` using the `--diff-tool` option. However, the command must behave as `diff` does for argument processing.

Amending a Commit Message

You can adjust the commit message of a committed change using the `cinstallman --commit-msg` operation. If `--msg` is not specified, it will be read from the terminal.

Removing Revisions

The `--del-revisions` option to `cinstallman` will delete all stored revisions but leave the working copy. You can do this if you want to free space used by revisions or wish to start over with the revision history.

The following two commands would free all space used in the revision history and then commit a new first revision:

```
# cinstallman --del-revisions --image myimage
# cinstallman --commit --image myimage --msg "Initial commit"
```

The working copy would remain intact and the two revisions would effectively collapse into one.

Examples

Note: For the examples in this section, you must be logged onto the admin node as the root user. All the examples pertain to a compute node image `sles11sp3`.

Adding a Revision and Querying Changes

The following example shows how to add the file `test_file` to the compute node image `sles11sp3`.

Procedure 3-8 Adding Revision and Querying Changes

1. Type the following command to view the current status of image `sles11sp3` in the VCS repository:

```
icicle:~ # cinstallman --history --image sles11sp3
Revision history for image sles11sp3, revisions 1 through 1
------------------------------------------------------------------------------
Revision: 1, Commit Time: Mon 18 Aug 2014 11:40:24 AM CDT
==============================================================================
Image created using cinstallman.
```

   All images you create using the `cinstallman` command are automatically added to VCS as revision 1.

2. Type the following command to add `test_file` to the working copy of the image:

```
icicle:~ # echo "test file" > /var/lib/systemimager/images/sles11sp3/tmp/test_file
```

3. Type the following command to show the differences between the working copy and revision 1.

   The following command shows that there is one difference from the version that was checked in:

```
icicle:~ # cinstallman --changed --image sles11sp3
icicle: cinstallman: Comparing revision 1 and working copy for image sles11sp3...
cmd: rsync -avHix --dry-run --delete /var/lib/systemimager/images//sles11sp3/ /var/lib/systemimager/vcs/
sending incremental file list
.d..t...... tmp/
>f+++++++++ tmp/test_file

sent 4961524 bytes  received 16926 bytes  1991380.00 bytes/sec total size is 4015834620  speedup is 806.
```

   The preceding output shows the addition of file `test_file`.

4. Type the following command to commit the new image that contains file `test_file`:

```
icicle:~ # cinstallman --commit --image sles11sp3 --msg "Added test_file to /tmp"
icicle: cinstallman: vcs: Using rsync to commit image sles11sp3...
cmd: rsync -aqHx --link-dest=/var/lib/systemimager/vcs/sles11sp3/1 /var/lib/systemimager/images/sles11sp
icicle: cinstallman: image sles11sp3 committed to vcs, rev: 2
```

5. Type the following command to verify that there are no differences between the working copy and the committed copy:

```
icicle:~ # cinstallman --changed --image sles11sp3
icicle: cinstallman: Comparing revision 2 and working copy for image sles11sp3...
cmd: rsync -avHix --dry-run --delete /var/lib/systemimager/images//sles11sp3/ /var/lib/systemimager/vcs/
sending incremental file list

sent 4961516 bytes  received 16918 bytes  1991373.60 bytes/sec total size is 4015834611  speedup is 806.
```

6. Type the following command to retrieve the revision history of the image:

```
icicle:~ # cinstallman --history --image sles11sp3
Revision history for image sles11sp3, revisions 1 through 2
```

```
------------------------------------------------------------------------
Revision: 1, Commit Time: Mon 18 Aug 2014 11:40:24 AM CDT ==============================================
Image created using cinstallman.

Revision: 2, Commit Time: Wed 20 Aug 2014 08:17:08 AM CDT ==============================================
Added test_file to /tmp

Done
```

7. Type the following command to display the list of all files changed between revision 1 and revision 2:

```
icicle:~ # cinstallman --changed --image sles11sp3 --rev 1..2
icicle: cinstallman: Comparing revisions 1 and 2 for image sles11sp3...
cmd: rsync -avHix --dry-run --delete /var/lib/systemimager/vcs/sles11sp3/2/ /var/lib/systemimager/vcs/sl
sending incremental file list
>f.st...... etc/opt/sgi/vcs-log-entry
.d..t...... tmp/
>f+++++++++ tmp/test_file

sent 5135898 bytes  received 16931 bytes  3435219.33 bytes/sec total size is 4015834611  speedup is 779.
```

Notice the presence of the `vcs-log-entry` file, which is always modified upon commits.

Reverting to a Previous Revision

If you revise and check in an image but later decide that you want to revert to a previous image, you can use the `cinstallman` command to preform the revert.

Procedure 3-9 Reverting to a Previous Version

1. Type the following command to declare that you want version 1 of the `sles11sp3` software image to be the working copy:

```
icicle:~ # cinstallman --revert --image sles11sp3 --rev 1
icicle: cinstallman: Removing image work dir: /var/lib/systemimager/images/sles11sp3
icicle: cinstallman: vcs: Syncing revision 1 in to place...
cmd: rsync -aqHx /var/lib/systemimager/vcs/sles11sp3/1/ /var/lib/systemimager/images/sles11sp3/
icicle: cinstallman: Working copy of sles11sp3 now at revision 1
```

2. Type the following command to retrieve the revision history:

```
icicle:~ # cinstallman --history --image sles11sp3
Revision history for image sles11sp3, revisions 1 through 2
--------------------------------------------------------------------------------
Revision: 1, Commit Time: Mon 18 Aug 2014 11:40:24 AM CDT ==========================================
Image created using cinstallman.

Revision: 2, Commit Time: Wed 20 Aug 2014 08:17:08 AM CDT ==========================================
Added test_file to /tmp

Done
```

As the preceding output shows, reverting the working copy does not affect the revision history.

3. To make the working copy correspond to the highest revision (the normal order of things), you can use the following command to commit the current working image (same content as revision 1):

```
icicle:~ # cinstallman --commit --image sles11sp3 --msg "Saving good copy of the image that doesn't have
icicle: cinstallman: vcs: Using rsync to commit image sles11sp3...
cmd: rsync -aqHx --link-dest=/var/lib/systemimager/vcs/sles11sp3/2 /var/lib/systemimager/images/sles11sp
icicle: cinstallman: image sles11sp3 committed to vcs, rev: 3
```

4. Type the following command to retrieve the revision history:

```
icicle:~ # cinstallman --history --image sles11sp3
Revision history for image sles11sp3, revisions 1 through 3
--------------------------------------------------------------------------------
Revision: 1, Commit Time: Mon 18 Aug 2014 11:40:24 AM CDT ==========================================
Image created using cinstallman.

Revision: 2, Commit Time: Wed 20 Aug 2014 08:17:08 AM CDT ==========================================
Added test_file to /tmp

Revision: 3, Commit Time: Wed 20 Aug 2014 08:25:39 AM CDT ==========================================
Saving good copy of the image that doesn't have the test_file in /tmp.

Done
```

Cloning an Image

The following example shows how to clone an image based on one of the previous revision images.

Procedure 3-10 Cloning an Image

1. Type the following command to clone an image based on revison 2, which includes `test_file`:

```
icicle:~ # cinstallman --create-image --clone --source sles11sp3 --rev 2 --image sles11sp3+test_file
About to use mksiimage --Copy to clone the image...
icicle: cinstallman: vcs: Syncing revision 2 of sles11sp3 to new sles11sp3+test_file ...
cmd: rsync -aqHx /var/lib/systemimager/vcs/sles11sp3/2/ /var/lib/systemimager/images/sles11sp3+test_file
icicle: cinstallman: Working copy of sles11sp3+test_file now at revision 2 of image sles11sp3 Ran sgi-mk
```

   Note in the `--rev 2` argument in the preceding command, which directs the `cinstallman` command to create the clone from revision 2.

2. Type the following command to verify the images that exist on the admin node after the cloning operation:

```
icicle:~ # cinstallman --show-images
Image Name                              BT VCS Compat_Distro
sles11sp3                               0  1   sles11
        3.0.76-0.11-default
sles11sp3+test_file                     0  1   sles11
        3.0.76-0.11-default
lead-sles11sp3                          0  1   sles11
        3.0.76-0.11-default
ice-sles11sp3                           0  1   sles11
        3.0.76-0.11-default
```

3. Type the following command to retrieve the revision history:

```
icicle:~ # cinstallman --history --image sles11sp3+test_file
Revision history for image sles11sp3+test_file, revisions 1 through 1
--------------------------------------------------------------------------------
Revision: 1, Commit Time: Wed 20 Aug 2014 08:29:07 AM CDT ======================================
Image clone of sles11sp3 by cinstallman
```

```
Done
```

Permanently Delete All Revisions

The following procedure explains how to use the `cinstallman` command to permanently delete all revisions.

Procedure 3-11 Permanently Deleting all Revisions

1. Type the following command to delete all revisions:

```
icicle:~ # cinstallman --del-revisions --image sles11sp3
Removing all revisions of sles11sp3, leaving the working copy...
```

2. Type the following command to retrieve the revision history and verify the deletion:

```
icicle:~ # cinstallman --history --image sles11sp3
icicle: cinstallman: There are no checked in revisions of this image.
Image history failed. See above for error messages
```

3. (Optional) Type the following command to commit the current image to the version control system:

```
icicle:~ # cinstallman --commit --image sles11sp3 --msg "A new beginning :)"
icicle: cinstallman: vcs: First revision, rsync the image work dir to the first revision...
cmd: rsync -aqHx /var/lib/systemimager/images/sles11sp3/ /var/lib/systemimager/vcs/sles11sp3/1/
```

4. (Optional) Type the following command to verify the commit:

```
icicle:~ # cinstallman --history --image sles11sp3
Revision history for image sles11sp3, revisions 1 through 1
------------------------------------------------------------------------
Revision: 1, Commit Time: Wed 20 Aug 2014 08:32:55 AM CDT ==============================================
A new beginning :)

Done
```

InfiniBand Fabric Management

This chapter includes the following topics:

About the InfiniBand Network

The SGI ICE X system topology includes internal InfiniBand switches. These switches are located in the individual rack units (IRUs). The InfiniBand technology facilitates fast communication between the SGI ICE compute nodes within a rack and between SGI ICE compute nodes in separate racks. The InfiniBand network on SGI ICE X systems uses Open Fabrics Enterprise Distribution (OFED) software. The OFED fabric management software monitors and controls the InfiniBand fabric. For information about OFED, see http://www.openfabrics.org.

Your system is configured with one of the following topologies:

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

Each SGI ICE X system is configured with one or two separate InfiniBand *fabrics* or *subnetworks*. The SGI documentation typically refers to these subnetworks as `ib0` and `ib1`. On storage compute nodes, there might be several interfaces called `ib0`, `ib1`, and so on, and all of them might be connected to the same subnetwork.

The SGI ICE X system uses a distributed memory scheme. Parallel processes in an application pass messages, and each process has its own dedicated processor and address space. This differs from the shared memory scheme found in the SGI UV system series. By default, MPI uses only the `ib0` subnetwork, and storage uses the `ib1` subnetwork. Other InfiniBand configurations are possible and can lead to better performance with specific workloads. For example, you can configure SGI's Message

Passing Interface (MPI) library, the SGI Message Passing Toolkit (MPT), to use one or two InfiniBand subnetworks to optimize application performance.

For information about MPI and MPT, see the *SGI MPI and SGI SHMEM User Guide*.

InfiniBand Fabric Management

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

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 162). The InfiniBand fabric connects the compute nodes, rack leader controllers (RLCs), and the SGI ICE compute nodes. It does not connect to the admin node 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, which runs on an RLC. For a system with two or more racks, the subnet manager for each fabric is usually configured to run on different RLCs. In a single rack system, both subnet managers will run on the single RLC. Each subnet manager may also be paired with a standby subnet manager which can take over in the event of the failure of the primary subnet manager. For more information, see "InfiniBand Fabric Failover Mechanism" on page 171.

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 subnet manager instance with a particular port on the RLC. Usually, `ib0` is mapped to port 1 of the InfiniBand host channel adapter (HCA) on the subnet manager node, and `ib1` is mapped to port 2 of the HCA on the subnet manager node. The subnet manager 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 admin node, enter the following command:

```
admin:~ # tempo-configure-fabric
```

The **InfiniBand Management Tool** GUI appears, as shown in Figure 4-1 on page 160.

You can also access the InfiniBand management tools from the cluster configuration tool. To start the cluster configuration tool, type `configure-cluster` at the system prompt and select **Configure Infiniband Fabric**.

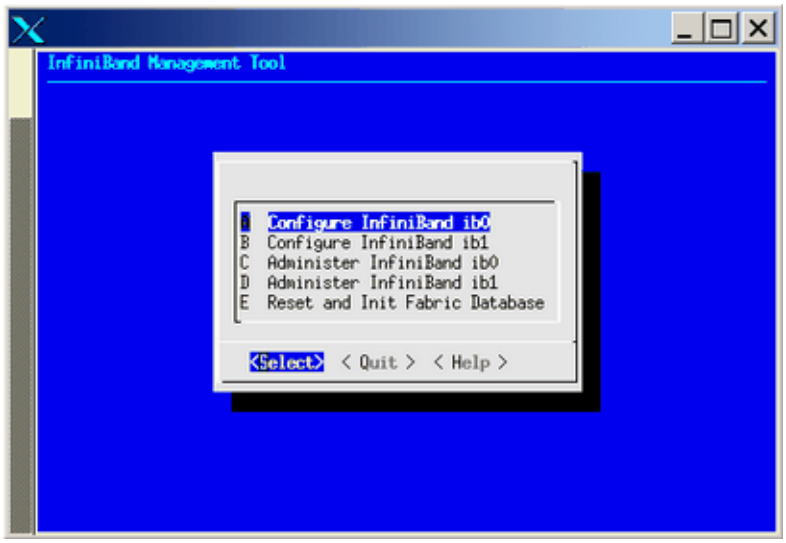

Figure 4-1 InfiniBand Management Tool Screen

Use the **Select** button to select the action you want to perform. A submenu will appear. Use the **Quit** button to return to the previous screen. Use the InfiniBand Management GUI to manage your InfiniBand fabric. You can use the **Help** button to get online help for each of the GUI actions.

If the `tempo-configure-fabric` command fails in a configuration or administrative operation, it suggests that you use the `sgifmcli`(8) command (described in "Fabric Component `sgifmcli` Command" on page 162) to debug the problem. Alternatively, you can use the **Reset and Init Fabric Database** option from the **InfiniBand Management Tool** main menu (see Figure 4-1 on page 160) to start over and completely reconfigure the InfiniBand fabrics.

From the **Configure InfiniBand** screen, make sure you select the **Configure Topology** option to set the topology as shown in Figure 4-2 on page 161. For more information, see "Network Topology" on page 167.

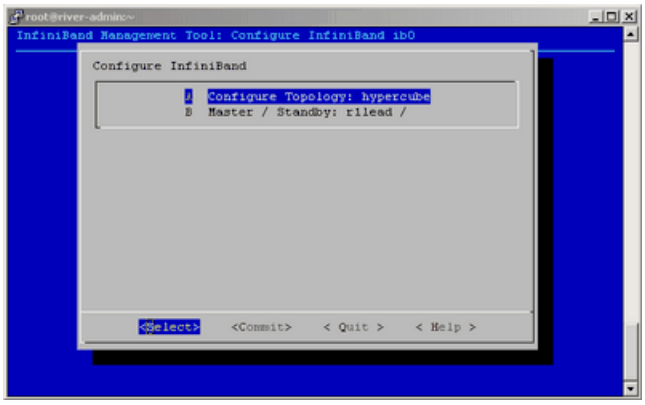

Figure 4-2 Configure Topology Screen

Use 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. You can use this screen to start, stop, restart, or refresh a fabric.

You can verify the status via the **Status** option, as shown in Figure 4-3 on page 161.

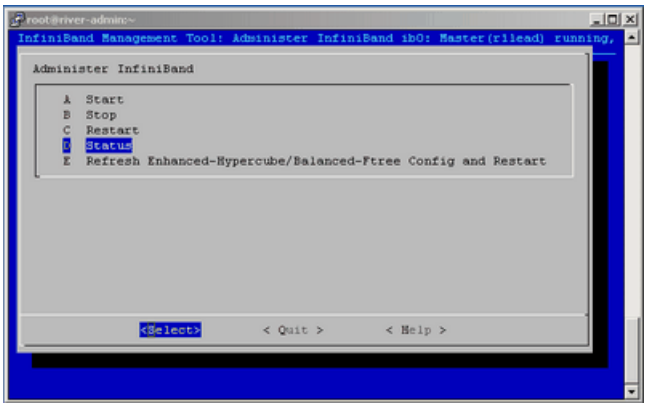

Figure 4-3 Administer InfiniBand Status Option

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 compute 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 manager and the optional standby subnet manager for the specified fabric (see "InfiniBand Fabric Failover Mechanism" on page 171).

Ideally, the refresh action for a fabric should be taken when there are no jobs running in the system. Restarting the subnet manager can have an adverse impact on the running jobs in the system.

Fabric Component `sgifmcli` Command

For the most common fabric management operations, the `tempo-configure-fabric` command (described in "InfiniBand Management Tool Graphical User Interface" on page 159) 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
mellanox5600	Mellanox 5600
mellanox6036	Mellanox 6036

To configure an external InfiniBand switch, cluster-wide InfiniBand connectivity is not required. The only necessity is that the supplied switch host name is resolvable and a working networking connection to the external InfiniBand switch exists. See the sgifmcli(8) man page for more information about adding external InfiniBand switches to your cluster's fabric.

Verify the integrity of an InfiniBand fabric requires that the InfiniBand network is first configured properly. This is most easily done using tempo-configure-fabric (see "InfiniBand Management Tool Graphical User Interface" on page 159). 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 subnet manager 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 admin node:

```
admin:~ # man sgifmcli
```

The `sgifmcli`(8) fabric administration utilities man page appears.

`sgifmdb` Fabric Management Database Command

The fabric component maintains a database (DB) of the objects it manages (managed objects). The database version is automatically set during cluster install. You do not need to set it. Most likely, this database will change over time. To manage multiple database versions and also to aid in field support, SGI has added another command line tool that currently reports the managed objects database version.

The `sgifmdb` command is, as follows:

```
sgifmdb [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]
```

It accepts the following general options:

General Option	Description
`-g, --get`	Reads the database version object from the database
`-d, --dump`	Dumps the database. This option allows the you to see what fabric objects are currently stored in the fabric database.
`-v, --version`	Prints version
`-r, --reset`	Resets the database and starts clean
`-h, --help`	-h, –help

Example 4-1 Getting `sgifmdb`(8) Command Help

For a `sgifmdb` command usage statement, perform the following from the admin node:

```
admin:~ # sgifmdb -h
SGI Fabric Component DB tool
Usage: db_version [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]

        -g, --get       Read DB version object from DB
```

```
-d, --dump      Dump the DB
-v, --version   Print version
-r, --reset     Reset the database and start clean
-h, --help      Show this text
```

InfiniBand Fabric Management Configuration and Operation Overview

Each subnet manager performs a light sweep of the fabric it is managing, every 10 seconds by default. The time interval is set by setting the `sweep_interval` variable in the `/opt/sgi/var/sgifmcli/opensm-ib0.conf.templ` file and then doing a **Commit** operation in the `tempo-configure-fabric` GUI. Alternately, the `sgifmcli` command has a `--arglist` option to set various subnet manager configuration parameters including the sweep interval.

Note: If your cluster is larger than 256 nodes, SGI highly recommends increasing this variable to 90 seconds or even larger value.

If a subnet manager 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 admin node.

Each `opensm` instance (one for each fabric) associates itself with a particular globally unique identifier (GUID) for a port on the node where `opensm` runs (see). 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.

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 subnet manager master.

Configuring the InfiniBand Fabric

This section describes how to configure and administer the InfiniBand fabric using the `sgifmcli`(8) command.

Note: SGI highly recommends that you use the `tempo-configure-fabric` GUI to configure and administer the fabric (see "InfiniBand Management Tool Graphical User Interface" on page 159).

Procedure 4-1 Configure the Master Subnet Manager

When configuring the subnet manager master, the following rules apply:

- Each InfiniBand fabric needs to have a subnet manager master.
- There can be at most one subnet manager 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 subnet manager master manager is launched. The `fabric` option associates the subnet manager master manager with either `ib0` or `ib1`. The `topology` option refers to the InfiniBand topology, which can be either hypercube, enhanced hypercube, or fat tree.

To configure a master for the fabric `ib0` on a hypercube cluster, perform the following steps:

1. From the admin node to configure a subnet manager master, perform the following:

```
# sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube
```

   This creates an subnet manager 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 subnet manager master is, as follows:

   ```
   sgifmcli --start --id identifier
   ```

   To start the `master_ib0` subnet manager 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 subnet manager master is, as follows:

   ```
   sgifmcli --stop --id identifier
   ```

   To stop the `master_ib0` subnet manager master, perform the following:

   ```
   # sgifmcli --stop --id master_ib0
   ```

   The subnet manager 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 subnet manager master is, as follows:

   ```
   sgifmcli --status --id identifier
   ```

To check the status of the `master_ib0` subnet manager 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 subnet manager 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 subnet manager master is, as follows:

   ```
   sgifmcli --remove --id identifier
   ```

   To remove the `master_ib0` subnet manager master, first stop it and then perform the **`-remove`** option, as follows:

   ```
   # sgifmcli --stop --id master_ib0

   # sgifmcli --remove --id master_ib0
   ```

   The subnet manager 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 subnet manager 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 has a failover mechanism. If the master subnet manager fails, the standby subnet manager 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 4-2 Enabling the InfiniBand Failover Mechanism

When enabling the InfiniBand failover mechanism, the following rules apply:

- Each InfiniBand fabric can optionally have exactly one standby.
- A standby subnet manager 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 subnet manager master and subnet manager standby for a particular fabric can not coexist on the same node.

SGI highly recommends that you use the `tempo-configure-fabric` GUI to configure the failover mechanism. If it is necessary to use `sgifmcli`(8) to enable the InfiniBand failover mechanism, perform the following steps:

1. If a subnet manager master is defined and running, stop it, as follows:

   # **sgifmcli --stop --id master_ib0**

   If the subnet manager master has not been defined, define it, as follows:

sgifmcli --mastersm --init --id master_ib0 --hostname r1lead --fabric ib0 --topology hypercube

2. Define the subnet manager standby, as follows:

```
# sgifmcli --standbysm --init --id standby_ib0 --hostname r2lead --fabric ib0
```

3. Start the subnet manager master, as follows:

   ```
   # sgifmcli --start --id master_ib0
   ```

   This automatically starts the subnet manager master and the subnet manager 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` subnet manager standby, first stop its master and then perform the **`remove`** option, as follows:

   ```
   # sgifmcli --stop --id master_ib0
   # sgifmcli --remove --id standby_ib0
   ```

   The subnet manager 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 162.

InfiniBand switches are generally classified as being of two types: edge switches and core or spine switches. Edge switches are used to connect to SGI ICE 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 `discover` command (see "`discover` Command" on page 49) 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 `discover` 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 4-3 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 admin node, initialize the switch. The syntax to initialize the switch is, as follows:

```
sgifmcli --init --ibswitch --model   --id  --switchtype [leaf | spine]
```

An example command is, as follows:

```
# sgifmcli --init --ibswitch --model voltaire-isr-2004  --id isr2004 --switchtype spine
```

This configures a Voltaire switch ISR2004 with hostname `isr2004` as a spine switch. `isr2004` refers to the admin port of the switch and needs to be configured previously to allow for switch access. The switch is now initialized and the root GUID from the spine switches have been downloaded.

4. From the admin node, add the switch to the fabric. The syntax to add the switch is, as follows:

   ```
   sgifmcli --add --id <fabric> --switch <hostname>
   ```

   An example command is, as follows:

   ```
   # sgifmcli --add --id ib0 --switch isr2004
   ```

   In this example, ISR2004 is connected to the `ib0` fabric.

5. For the new switch to be activated, the subnet manager master and the optional subnet manager standby need to be (re)started.

   ```
   # sgifmcli --start --id master_ib0
   ```

   If the subnet manager 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 subnet manager master failure the subnet manager 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 4-4 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 49. The switch model to be used is "mellanox5030". The `discover` command supports external switches in a manner similar to racks and compute nodes, except that switches do not have BMCs and there is no software to install.
3. Discover all external switches.
4. Use `tempo-configure-fabric` to configure the fabric, as described in "InfiniBand Management Tool Graphical User Interface" on page 159.

   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 159, 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 4-5 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 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.

Utilities and Diagnostics

The InfiniBand diagnostics package on your SGI ICE X system contains tools and diagnostic software for the Open Fabrics Enterprise Distribution (OFED) software. These tools reside on the rack leader controllers (RLCs) in the `/usr/sbin` directory. In addition, the `opensm`(8) man page describes options that control logging and debugging.

For information about the InfiniBand fabric diagnostics, see the following topics:

Retrieving Information About InfiniBand Diagnostic Tools

This topic explains how to find the complete list of diagnostic tools that are available on SGI ICE X systems. Later topics explain some of the individual tools in more detail.

To see a full list of diagnostics, complete the following procedure.

Procedure 4-6 To retrieve information about OFED tools and diagnostics

1. Log into the admin node as the root user.
2. Type the following command to retrieve the identifiers for the RLCs:

   ```
   # cnodes --all
   ```

3. Use the `ssh`(1) command to log into one of the RLCs.
4. Retrieve the name of the diagnostic package.

   The following example shows the command to use and typical output:

   ```
   # rpm -qa  | grep infiniband
   infiniband-diags-1.5.7-0.3.2
   ```

5. Retrieve information about the utilities in the diagnostic package.

   The following example shows the command to use and typical output:

   ```
   # rpm -ql infiniband-diags-1.5.7-0.3.2 | grep sbin
   /usr/sbin/check_lft_balance.pl
   /usr/sbin/dump_lfts.sh
   /usr/sbin/dump_mfts.sh
   /usr/sbin/ibaddr
   ```

```
/usr/sbin/ibcacheedit
/usr/sbin/ibcheckerrors
/usr/sbin/ibcheckerrs
/usr/sbin/ibchecknet
/usr/sbin/ibchecknode
/usr/sbin/ibcheckport
/usr/sbin/ibcheckportstate
/usr/sbin/ibcheckportwidth
/usr/sbin/ibcheckstate
/usr/sbin/ibcheckwidth
/usr/sbin/ibclearcounters
/usr/sbin/ibclearerrors
/usr/sbin/ibdatacounters
/usr/sbin/ibdatacounts
/usr/sbin/ibdiscover.pl
/usr/sbin/ibfindnodesusing.pl
/usr/sbin/ibhosts
/usr/sbin/ibidsverify.pl
/usr/sbin/iblinkinfo
/usr/sbin/iblinkinfo.pl
/usr/sbin/ibnetdiscover
/usr/sbin/ibnodes
/usr/sbin/ibping
/usr/sbin/ibportstate
/usr/sbin/ibprintca.pl
/usr/sbin/ibprintrt.pl
/usr/sbin/ibprintswitch.pl
/usr/sbin/ibqueryerrors
/usr/sbin/ibqueryerrors.pl
/usr/sbin/ibroute
/usr/sbin/ibrouters
/usr/sbin/ibstat
/usr/sbin/ibstatus
/usr/sbin/ibswitches
/usr/sbin/ibswportwatch.pl
/usr/sbin/ibsysstat
/usr/sbin/ibtracert
/usr/sbin/perfquery
/usr/sbin/saquery
/usr/sbin/set_nodedesc.sh
/usr/sbin/sminfo
```

```
/usr/sbin/smpdump
/usr/sbin/smpquery
/usr/sbin/vendstat
```

`ibstat`(8) and `ibstatus`(8) Commands

You can use the `ibstat`(8) command to see the current status of the host channel adapters (HCAs) in your InfiniBand fabric. The status includes the HCAs on the rack leader controllers (RLCs).

Example 1. The following output was obtained **before** starting the fabric management software.

```
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
```

Example 2. The following output was obtained from the `ibstat`(8) command **after** the fabric management software was 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
```

Example 3. You can use the `ibstatus`(8) command to show the link rate. The `ibstatus`(8) command is less verbose than the `ibstat` command.

```
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`(8) Command

The `perfquery`(8) command is useful for finding errors on one or more host channel adaptors (HCAs) and errors on switch ports. You can also use `perfquery`(8) command to reset HCA and switch port counters.

Example 1. The following example shows how to retrieve the usage statement for the `perfquery`(8) command.

```
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
```

Example 2. The following example shows `perfquery`(8) command output.

```
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`(8) Command

The `ibnetdiscover`(8) command enables you to discover the InfiniBand fabric.

Example 1. The following example retrieves the usage statement for the `ibnetdiscover`(8) command. The output has been truncated for inclusion in this documentation.

```
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
```

Example 2. The following example shows sample `ibnetdiscover`(8) output.

```
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`(1) Command

The `ibdiagnet`(1) command scans the fabric and extracts information about connectivity and devices.

Example 1. The following example retrieves the usage statement for the `ibdiagnet`(1) command.

```
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
```

Example 2. The following example output contains no errors, which means that 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.
```

Example 3. The following example shows how to use `ibdiagnet` to load the fabric for testing.

```
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.
```

OpenSM Logging and Debugging Options

OpenSM is the InfiniBand subnet manager. The `opensm`(8) man page describes the ranges for the debugging and logging options. When you start a troubleshooting session, SGI recommends that you set the following parameters:

- `-D 0x7`, which sets a reasonable log verbosity level.
- `-d 2`, which clears the logs immediately after each log message.

For more information about the OpenSM utility, log into one of the RLCs and see the `opensm`(8) man page.

System Maintenance, Monitoring, and Debugging

This chapter includes the following topics:

Hardware Maintenance Procedures

This section describes some common maintenance procedures, as follows:

Taking a Node Offline for Maintenance Temporarily

This section describes how to temporarily take a node offline for maintenance.

Procedure 5-1 Temporarily Take a Node Offline for Maintenance

To temporarily Take a node offline for maintenance, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

   # **cpower node off 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 node on r1i0n0**

7. Enable the node in the batch scheduler (depends on your batch scheduler).

Replacing 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 5-2 Permanently Replace a Failed Blade

To permanently replace a failed blade (compute node), perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

   # **cpower node off 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.

6. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 125 for your options), as follows:

   # **cimage --set mycomputeimage mykernel r1i0n0**

7. Power up the node, as follows:

   # **cpower node on r1i0n0**

8. Enable the node in the batch scheduler (depends on your batch scheduler).

Removing a Blade Permanently

This section describes how to permanently remove a blade from your SGI ICE X system.

Procedure 5-3 Permanently Remove a Blade

To permanently remove a blade from your system, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).

2. Power off the node, as follows:

   # **cpower node off 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.

Adding a New Blade

This section describes how to add a new blade to an SGI ICE X system.

Procedure 5-4 Add a New Blade

To add a new blade to your system, perform the following steps:

1. Physically insert the new blade
2. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon.
3. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 125 for your options), as follows:

   # **`cimage --set mycomputeimage mykernel r1i0n0`**

4. Power up the node, as follows:

   # **`cpower node on r1i0n0`**

5. Enable the node in the batch scheduler (depends on your batch scheduler).

Replacing a Switch

During the initial installation and configuration of the SGI ICE X system, you saved your switch configurations to one or more files in the `/tftpboot` directory on the admin node. When you replace a switch, you can push the saved configuration file from the admin node to the new switch. The following procedure explains how to replace a switch and used the saved configuration file to configure the new switch.

Procedure 5-5 To configure a new switch

1. Use the switch manufacturer's instructions to physically replace the old switch with the new switch.

   Make sure that the cabling is identical to the way the old switch cabling was configured.

2. Log into the admin node as the root user, and type the following command to push the configuration file to the new switch:

   `switchconfig push_switch_config -s` *switch_ID* `-f` *file* `[--debug]`

   For *switch_ID*, specify the name of the new switch.

For *file*, specify the name of the file that contains the saved switch configuration information. The command copies the file from `/tftpboot/`*file*`.cfg` on the admin node. It is not necessary to specify the `.cfg` extension when you use this command.

The `--debug` parameter is optional.

For example, the following command copies the configuration file for `mgmtsw0` from file `/tftpboot/mgmtsw0_startup1.cfg` to the new switch:

```
switchconfig push_switch_config -s mgmtsw0 -f mgmtsw0_startup1 --debug
```

3. (Optional) Type the following command to view debugging information and logging information:

```
tail -100 /var/log/switchconfig.log
```

Node Replacement Procedure for Cold Spare Admin Node, Rack Leader Controller (RLC), or Compute Nodes

This section describe how to install and configure a spare admin node, RLC, or managed compute node. The cold spare can be a shelf spare or a factory-installed cold spare that ships with your system. For more information on cold spare requirements and tools needed to do this procedure, see "Cold Spare Admin Node or Rack Leader Controler (RLC) Availability" on page 192.

It covers the following topics:

Note: When ordering shelf spare systems from SGI, it is important to order spare nodes appropriate to or in conjunction with your SGI ICE X system. This is because the serial number is programmed into the admin node itself. If you try to migrate the admin node to a shelf spare system that does not have the correct system serial number programmed into it, parts of the system 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 admin node and RLC are generally installed within (or in some cases on top of) the system rack. The replacement of a failed admin node 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 193, "Transfer Disks from Existing Server to the Cold Spare" on page 196, "Migrating to a Cold Spare: Importing the Disk Volumes" on page 197 and "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 199.

Note: If you are using multiple root slots repeat the procedures described in this section for each slot.

Cold Spare Admin Node or Rack Leader Controler (RLC) Availability

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

If the admin node or RLC node should fail, the cold spare can be swapped in to position to take over the duties of the failed node.

If you wish to make use of cold spare nodes, SGI suggests that you have both a admin node 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 admin node and an RLC are different. For example, a admin node 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. An admin node may not be set up this way.
- Given the first two items in this list, if you try to use a shelf-spare admin node as an RLC, the RLC is not discovered properly.

Shelf Spare Hardware Limitations

Currently, the hardware replacement procedure described in this section only supports admin nodes, rack leader controllers (RLCs), and managed compute nodes supplied by SGI.

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 admin node 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 5-1 on page 194.

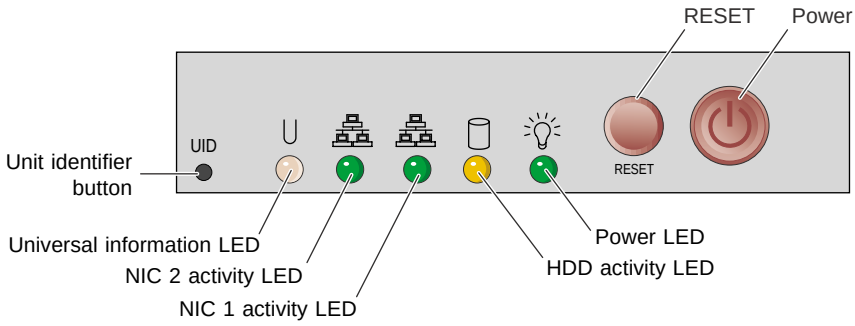

Figure 5-1 Admin/RLC Server Front Panel Controls and Indicator LEDs

The universal information LED (left side of the panel) shows two types of failure that can bring the server down. This multi-color LED blinks red quickly to indicate a fan failure and blinks red slowly for a power failure. A continuous solid red LED indicates a CPU is overheating.

If the unit's power supply has failed or been disconnected, the power LED (far right) will be dark. Check both ends of the power cable for a firm connection prior to switching over to the cold spare.

If you find that an admin node or 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 admin node stores the system-wide serial number. The admin node shelf spares must be ordered from the factory as admin node shelf spares so that the proper serial number can be stored within.

Procedure 5-6 Replacing a Node with a Cold Spare: Installing the Hardware

To replace an admin node or RLC that has failed, perform the following steps:

1. Power down the failed node (if possible).
2. Disconnect both power cables, see Figure 5-2 on page 196 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 197.

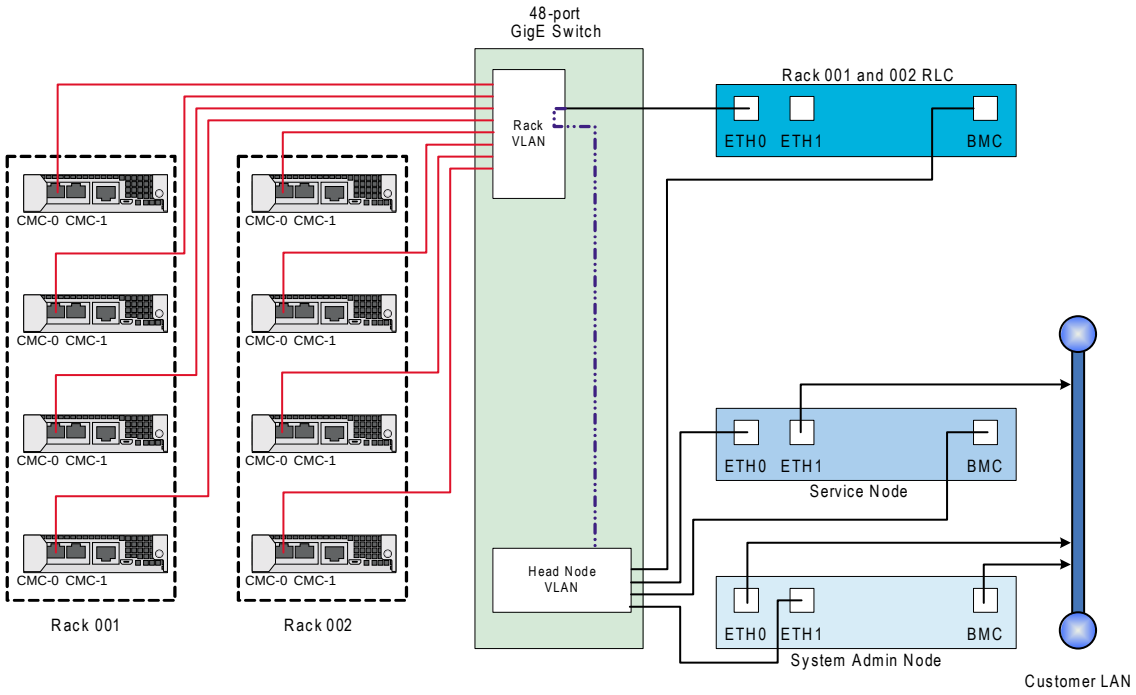

Figure 5-2 Simple CMC LAN (VLAN) Cable Examples

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 5-3 on page 197.

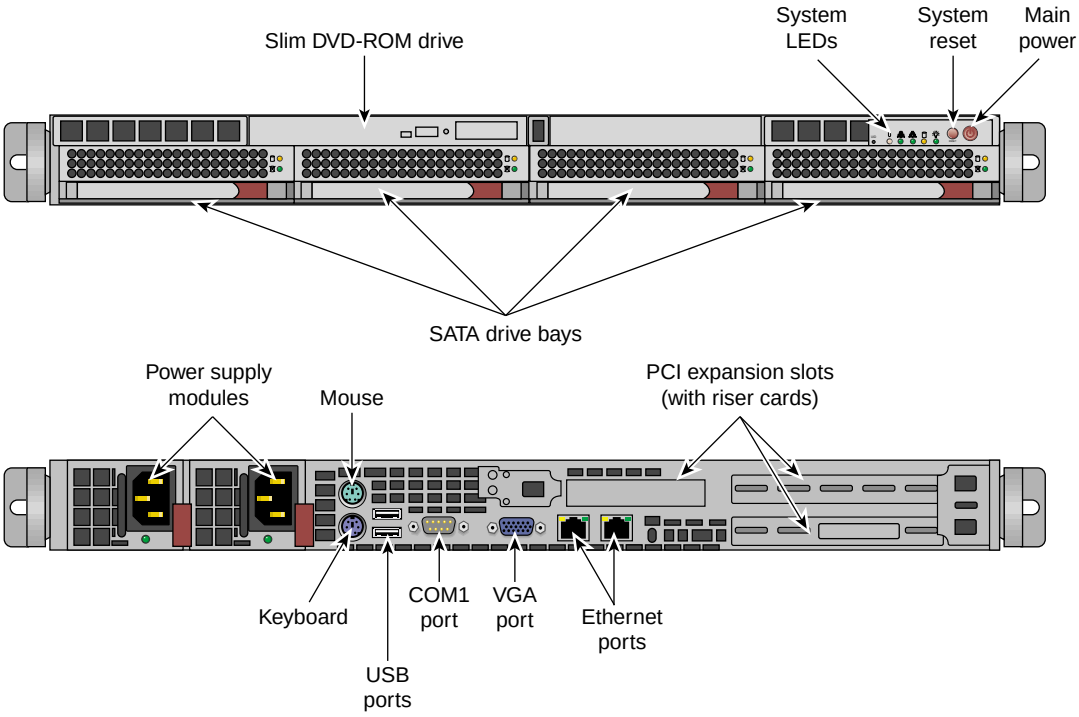

Figure 5-3 Admin Node and RLC Server Front Features and Rear Connector Locations

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 193. 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 5-7 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 199.

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

This section provides information on booting the system for the first time on a replacement node.

Note: Important: If your site is using cascading dual boot, only the currently used slot will be updated or repaired. Therefore, if the admin node is booted to slot 2, the fix up operations documented in these sections only apply to slot 2. The instructions need to be done for each slot you wish to fix up.

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 compute 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 admin node, a script is run by hand to integrate the repaired admin node 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, when doing `discovery` operations, you may find it convenient to disable it.

For the case of the admin node, you will need to ensure your console output goes to the VGA screen and not serial-over-lan (SOL). For managed compute 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?

- admin nodes, all cases: Procedure 5-8, page 200.
- Managed compute nodes and RLCs in a non-cascading dual boot cluster: Procedure 5-8, page 200.
- Managed service, RLCs in a cascading dual boot cluster: Procedure 5-9, page 201.

Procedure 5-8 Migrating to a Cold Spare in a Non-cascading Dual Boot Cluster Node

This section describes how to boot the admin node or RLC or compute node in non-cascading dual boot clusters.

Note: This section applies to admin nodes and sites that are **not** making use of cascading dual boot. Cascading dual boot is set up by default in newer SMC software releases. If you are using cascading dual boot, follow these instructions **only** for the admin node.

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

1. Ensure that the VGA console is powered on.
2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 197).

   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, highlighted could be **SUSE Linux Enterprise Server 11 SP3**. Enter **e** to edit the boot parameters for this boot only.

3. Enter **e** to edit the kernel parameters.
4. Arrow down once so that the line starting **kernel** is highlighted.
5. Look at the settings. If no serial console is defined, you do not need to change anything. If a serial console is defined, append `console=tty0` to the end of the parameter list. This will ensure that console output goes to the VGA screen for this boot.

   Note: By default, the admin node goes to the VGA screen. Therefore, this adjustment does not need to be made. RLCs and compute 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 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 the SMC software.

   ---

10. Reboot the node and let it boot normally.

Procedure 5-9 Migrating to a Cold Spare: Compute Node or RLC Using Cascading Dual Boot

This section describes what to do for managed compute nodes and RLCs in a cluster making use of cascading dual boot. It does **not** apply to admin nodes. For admin nodes, see Procedure 5-8, page 200.

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 197).

   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 197). 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 compute nodes and rack leader controllers (RLCs) to automatically make the necessary adjustments for both the node setup itself and the 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 compute 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 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

The cluster allocates four IP addresses for auto-recovery operations. It allocates these IP addresses as available and the addresses might not be the same from one cluster to the next.

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 49.

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.

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 the *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*.

In addition, you need to add or remove the RLC from the `/var/yp/ypservers` file on NIS Master compute node. Remember to use the `-ib1` name for the RLC, as compute nodes cannot resolve `r2lead` style names. For example, use `r2lead-ib1`.

```
# cd /var/yp && make
```

Out-of-Memory Occurrences on SLES11 and PBS Professional Batch Scheduler

SGI ICE compute nodes are diskless blade servers typically configured with `nfs` root and a small (50 MB) swap space that is served via `iscsi`. A maximum of 288 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. 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 the *SGI MPI and SGI SHMEM User 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 daemon and each arrayd children. The default is "-17,0",
hence resulting in the arrayd daemon never being selected as a
candidate by the oom kernel killer thread and children selected as
```

normal candidates. 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. 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.

- 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. 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

System Monitoring

This section describes the following system monitoring tools:

Ganglia

Ganglia is a scalable, distributed monitoring system. It displays web browser-based, real-time (on demand) histograms of system metrics. Figure 5-4 on page 207 shows an example display.

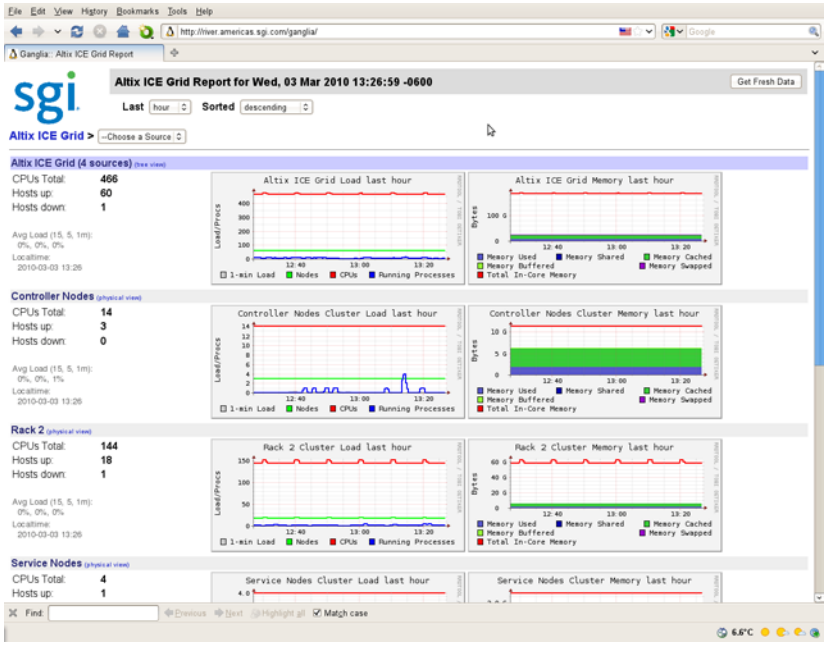

Figure 5-4 Ganglia System Monitor

Each SGI ICE compute node (blade) is a single monitoring source that sends its statistics to the rack leader controller (RLC). After collecting the data, the RLC forwards aggregated rack statistics to the admin node. The RLC also sends its own statistics to the admin node. The admin node is the meta-aggregator for the entire SGI 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 5-5 on page 208 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 useful when trying to locate a blade that is down.

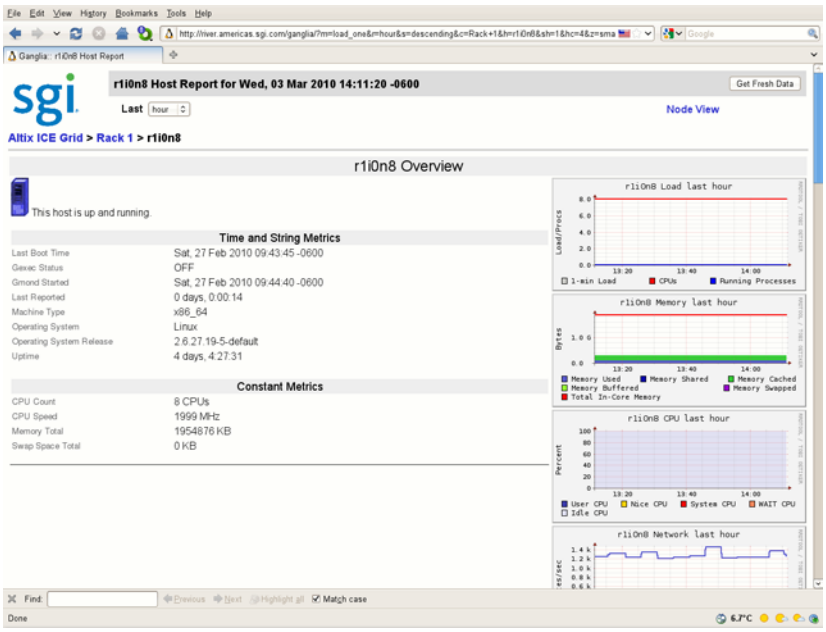

Figure 5-5 Ganglia System Monitoring Node View

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

Accessing the Ganglia System Monitor

To access the Ganglia system monitor, point your browser to the following location: http://*admin_domain_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, compute 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 compute node. The **Get Fresh Data** button allows you to see current data performance.

Default Admin Node Metrics

By default, SMC has configured Ganglia to gather the following categories of metrics for admin nodes:

cpu
disk
diskstat
load
memory
memory_vm
network
process
procstat
ssl
tcp
tcpext
udp

Default RLC Metrics

By default, SMC has configured Ganglia to gather the following categories of metrics for RLCs:

no_group
cpu
disk
diskstat
load
memory
memory_vm
network
process
procstat
ssl
tcp
tcpext
udp

Note: The no_group metrics include the temperature-related metrics.

Default Compute Node Metrics

By default, SMC has configured Ganglia to gather the following categories of metrics for compute nodes:

cpu
disk
load
memory
network
process

SEL/Hardware Event Logs

All server nodes in a cluster have a specialized controller called the board management controller (BMC). For instance, in an SGI ICE cluster, the admin node, rack leader controllers (RLCs), the compute nodes, the chassis management controllers (CMCs), and all the SGI ICE compute nodes (blades) all have BMCs. These units provide a broad set of functions as described in the IPMI 2.0 standard. SMC 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 SGI ICE 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*> `TEMPO:`<*node*> `EVENT:`<*event*> `APP:`<*app*> `Date:`<*date*> `VERSION:`<*version*> `TEXT` <*text*>

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*event*>	`BMC_CMC`
<*app*>	SEL-LOGGER
<*date*>	date / time of the event
<*version*>	1.0
<*text*>	Exact copy of the hardware event description from the BMC

After reading the events from the BMCs, the BMC event logs are cleared on the controller to avoid duplicate events.

Heartbeat Daemon

The availability of each node in an SGI ICE X system is monitored by a lightweight daemon called `tempohbc`. Each managed compute node, rack leader controller (RLC), and SGI ICE compute node runs this daemon and reports its status to the server which monitors it. The server daemon, which runs on the admin node and 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*> `TEMPO:`<*node*> `EVENT:HEARTBEAT APP:TEMPOHBD Date:`<*date*> `VERSION:1.0 TEXT` <*text*>

The HEARTBEAT event is created when nodes fail or recover, described by the `TEXT` field.

The following fields are all of the type string:

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

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

Nagios

Nagios is a feature-rich, web-based system monitoring tool for networks and clusters. Among its features are the following:

- Ability to monitor the health of specified cluster nodes and services
- Ability to notify a specified audience by email or SMS if a critical event occurs
- Ability to gather and display statistics about specified nodes and services
- Highly customizable

Accessing Nagios

Nagios is installed on all admin nodes and also on rack leaders on SGI ICE X clusters. To monitor the entire cluster, access Nagios on the admin node. To only monitor the nodes subordinate to a rack leader, access Nagios on that rack leader.

- Accessing Nagios on the admin node

  `http://`*admin-public-domain-name*`/nagios/`

- Accessing Nagios on a leader node

  `http://`*admin-public-domain-name/leader-name*`/`

In both cases, the default username/password is `nagiosadmin/sgisgi`.

Procedure 5-10 Changing the password from SLES

To change the password from SLES, do the following:

1. Enter the following command:

   # **`htpasswd2 -c /usr/local/nagios/etc/htpasswd.users nagiosadmin`**

2. At the prompt, supply default password `sgisgi`.
3. After changing the password, restart Apache services.

   Use one of the following commands:

   - On RHEL 7 and SLES 12 platforms, type the following command:

     # **`systemctl status httpd`**

   - On RHEL 6 and SLES 11 platforms, type the following command:

     # **`service apache2 restart`**

Procedure 5-11 Changing the password from RHEL

To change the password from RHEL, do the following:

1. Enter the following command:

   # **`htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin`**

2. At the prompt, supply default password `sgisgi`.
3. After changing the password, restart HTTP services.

   Use one of the following commands:

   - On RHEL 7 platforms, type the following command:

     # **`service restart httpd`**

   - On RHEL 6 platforms, type the following command:

     # **`service httpd restart`**

Configuring Nagios

By default, SMC has Nagios configured to monitor the following services:

- Pings of all hosts
- `check_ganglia` parameters—Checks Ganglia parameters on five-minute and one-minute load averages. An alert is set for a value of 10 for both five-minute and one-minute load averages.
- `check_log`—Checks for memlog failures on the admin and leader nodes.

On the admin node and rack leaders, the following additional parameters are monitored:

Current load
Current users
HTTP
Root partition
SSH
Swap usage
Total processors

Modifying the Configuration Files

Nagios is installed in the `/opt/sgi/nagios` directory of admin nodes and also in the same directory of RLCs in SGI ICE X systems. The main configuration file, `nagios.cfg`, is located in the `/opt/sgi/nagios/etc` directory. This file contains the directives that control how Nagio monitors the system. The directives specify object defintion files that target hosts, services, hostgroups, contacts, contact groups, commands, etc. There is where you define all of the objects that you want to monitor and how you want to monitor them.

You use the `cfg_file` and/or the `cfg_dir` directives to specify the following object definition files in the main configuration file:

- `hosts.cfg`—Contains all of the hosts associated with the node where Nagios is installed.
- `services.cfg`—Contains all of the services to be executed on the hosts or host groups defined in the `services.cfg` templates.
- `commands.cfg`—Contains all of commands that are the Perl, shell, or Python scripts to be executed by `services.cfg`.
- `contacts.cfg`—Contains the contacts or contact groups to be notified by services.

Directory `/opt/sgi/nagios/libexec` contains the available plugins (commands/services).

Validating Changes and Reloading Nagios

After you change one of the object definition files, you should validate the change using the following command:

```
# /opt/sgi/nagios/bin/nagios -v /opt/sgi/nagios/etc/nagios.cfg
```

After you successfully validate your changes, reload Nagios. Use one of the following commands:

- On RHEL 7 and SLES 12 platforms, type the following command:

  ```
  # systemctl reload nagios
  ```

- On RHEL 6 and SLES 11 platforms, type the following command:

  ```
  # service nagios reload
  ```

Performance Co-Pilot

A wealth of system metrics are also available through the Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The Performance Co-Pilot collection daemon (PMCD) runs on the admin node, rack leader controllers (RLCs), and managed compute nodes. A performance metrics domain agent (PMDA) is running on the RLCs, which collects metrics from the SGI ICE 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 on SGI ICE X systems, in order to minimize load on the SGI ICE compute nodes. The following metrics are available for each SGI ICE 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 SGI ICE 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_monitoring 1
```

The default list of metrics that are collected by each SGI ICE 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 an SGI ICE 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.nice \
cluster.kernel.percpu.cpu.sys cluster.kernel.percpu.cpu.idle \
cluster.kernel.percpu.cpu.intr cluster.kernel.percpu.cpu.wait.total
```

Monitoring SDR Metrics

The sensor data repository (SDR) metrics are available through Performance Co-Pilot. The SDR provides temperature, voltage, and fan speed information for all compute nodes, rack leader controllers (RLCs), SGI ICE compute nodes, and CMCs. This information is collected from service and SGI ICE compute nodes through their BMC interface, so it is out-of-band and does not impact the performance of the node.

For information about Performance Co-Pilot, see the *Linux Application Tuning Guide for SGI X86-64 Based Systems.*

The following metrics are available through the PMCD:

```
admin:~ # pminfo -h r1lead sensor
sensor.value.fan
sensor.value.voltage
sensor.value.temperature
```

Each sensor will have a separate instance within the domain, with the instance of the form:

```
<nodeName>:<nodeType>:<metricName>

nodeName ::= Tempo 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

The SGI ICE cluster monitors itself and shuts down components if the temperature is too high. This feature is enabled by default as a safety mechanism. The following procedure describes how to turn it off.

Procedure 5-12 To turn off the `temperature.pmie` feature

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. Restart the `pmie` daemon.

   Use one of the following commands:

- On RHEL 7 and SLES 12 platforms, type the following command:

  # **systemctl restart pmie**

- On RHEL 6 and SLES 11 platforms, type the following command:

  # **/etc/init.d/pmie restart**

If you just want to adjust `temperature.pmie` values, see "Adjusting `temperature.pmie` Values" on page 220.

This has to be done on the admin node 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

You can adjust the warning or shutdown temperature values manually on the admin node 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 are preserved between reboots.

The following procedure explains how to change the values.

Procedure 5-13 To adjust `temperature.pmie` values

1. Open the `/opt/sgi/lib/temperature.pmie` in a text editor and edit accordingly.

   For example:

   ```
   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. Type the following command to verify that you updated the script correctly:

   # **pmie -C /opt/sgi/lib/temperature.pmie**

If there are no errors, the preceding command returns with no message.

3. Restart the `pmie` service.

   Use one of the following commands:

   - On RHEL 7 and SLES 12 platforms, type the following command:

     # **systemd restart pmie**

   - On RHEL 6 and SLES 11 platforms, type one of the following commands:

     # **service pmie restart**

     or

     # **/etc/init.d/pmie restart**

To turn off the `temperature.pmie` value, see "Turning Off the `temperature.pmie` Feature" on page 219.

Cluster Performance Monitor

You can use the Cluster Performance Monitor to monitor your SGI ICE X system. Log into the admin node using the `ssh -X` command. Execute the pmice command and the **pmice - Cluster Performance Monitor** appears, as follows:

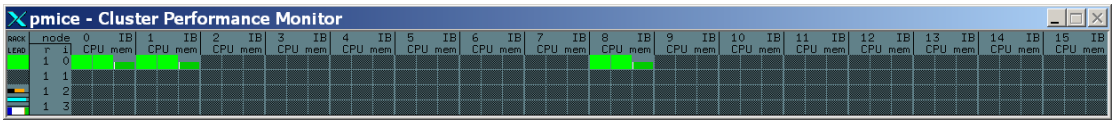

Figure 5-6 pmice- Cluster Performance Monitor

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 IRU Power Up and Automatic Power Down Problems

The following topics describe troubleshooting tactics you can employ when there are problems with powering up and powering down your SGI ICE X system:

About SGI ICE X Power Supplies

On SGI ICE X systems, the power shelves and their associated power supplies are external to the individual rack units (IRUs). Two to four power shelves power either one or two IRUs. The number of IRUs associated with a set of power shelves depends on your specific system configuration.

A *power domain* controls the power for a set of IRUs. The power domain includes the IRUs, their associated power shelves, and the fan controllers. When you power on or power off a set of IRUs, the power domain coordinates the action for the IRUs it controls.

Similarly, a *cooling domain* controls the cooling for a set of IRUs.

Relative to the IRU, the power and cooling systems are not internal to the IRU. The CMCs communicate to the power shelves over the CANbus cable, which is a physical cable in the back of the rack. Collectively, the CMC, the power shelves, the fan controllers, and the CANbus cable are called the *CAN*.

Your SGI ICE X system has one of the following power domain and cooling domain configurations:

- D-Racks with single node blades

  In this configuration, the power and cooling domains are the same.

  There are two IRUs, which constitute an *IRU pair*. There is one CMC in each IRU. The pair of IRUs share two power shelves with three power supplies in each shelf. There is one fan controller for the 12 fans.

- M-Racks with single node, air-cooled blades

  In this configuration, the power doman and cooling domains are distinct.

The power domain consists of two IRUs. There is one CMC in each IRU. There are four power shelves, with three power supplies, in each each IRU.

The cooling domain consists of two racks (or four IRU pairs) with external air cooling from a cooling rack.

- M-Racks with single-node or twin-node water-cooled blades (early version)

  In this configuration, the power doman and cooling domains are distinct.

  The power domain consists of two IRUs. There can be one or two CMCs in each IRU. There are four power shelves, with three power supplies, in each IRU.

  The cooling domain consists of four racks (or eight IRU pairs).

- M-Racks with single-node or twin-node water-cooled blades (late version)

  In this configuration, the power doman and cooling domains are distinct.

  The power domain consists of one IRU. There can be one or two CMCs in each IRU. There are four power shelves, with three power supplies, in each IRU.

  The cooling domain consists of four racks (or sixteen IRUs).

About the Power On Process

When you issue a power on, the first chassis management controller (CMC) in a power domain is the CMC that actually performs the power on. When you issue a power off, the last CMC in the power domain is the CMC that actually performs the power off.

The power-on and the power-off processes occur in phases. When you use the `cpower` command to power-up and power-down, the command handles the process for you.

When you type the `cpower node on "r*i*n*"` command on an admin node to power-up the SGI ICE compute nodes, the following occurs:

1. The each CMC turns on the power supplies.

   At this point the BMCs on the compute blades have power, are booted, and are running.

2. The CMC enables the fans and waits until it determines that air is moving through the IRU.

3. The CMC send an IPMI command to the BMCs that tells the BMCs to enable power to the compute blades.
4. The BMCs enable power to the compute blades.

CMC Monitoring

During typical operation, the CMC monitors several aspects of the power supply.

On an SGI ICE X system with M-Racks, which uses external cooling, the CMC verifies the following:

- That communication between the CMC and its associated CRC and CDU is open. The CMC monitors the CRC and CDU for error conditions and can power off the IRU if needed. The rack number of the CMC determines the CRC and CDU that the CMC monitors.
- That the correct number of power shelves can be detected.

On an SGI ICE X system with D-Racks, the CMC verifies the following:

- That a certain number of fans are present and spinning. Environmental software on the CMC controls the fan speed and reports failures.
- That the correct number of power shelves can be detected.

Power Cycling the IRUs

The chassis management controllers (CMCs) enable power to the IRUs. If an SGI ICE X system loses power abruptly, your first step is to power cycle the IRUs. If power supplies are turned off, the LEDs on the power supplies are flashing green. If one of the power supplies has a fault, the LED is solid amber. Depending on how the software in the system detected the power off, there can be log entries that provide more information. For information about the log entries, see "Log Files" on page 237.

In most cases, if you can power cycle the IRUs, the CMCs can restore power.

The following procedure explains how to power cycle the IRUs.

Procedure 5-14 To power cycle the IRUs

1. Log into the admin node as the root user.

2. (Conditional) Use the cnodes command to retrieve a list of RLCs and CMCs in your system.

   Perform this step if you are unsure of the system ID for the RLC and CMC that is affected.

   Type the following commands:

   ```
   # cnodes --leader
   r1lead
   r2lead
   # cnodes --cmc
   r1i0c
   r2i1c
   ```

   The preceding commands show the IDs for the RLCs and CMCs on a 2–rack system.

3. Use the ssh(1) command to connect to the rack leader controller (RLC) in which the problem CMC resides.

   For example, if the CMC you need to power cycle is in rack 1, type the following command:

   ```
   # ssh r1lead
   ```

4. Use the cpower command to retrieve information about the CMCs in the rack.

   For example:

   ```
   # cpower iru status "r1i*"
   xxxxx
   xxxxx
   .
   .
   .
   r1i0c: power is On
   r2i1c: power is On
   ```

5. Use the ssh(1) command to connect to the problem CMC that you want to power cycle.

   For example:

   ```
   # ssh r1i0c
   ```

6. Type the following command to affirm that the power is on for the CMC:

```
> power status
Power is ON
```

If the `power status` command does not return `Power is ON`, then the CMC and one or more of components connected to the CMC are still down. In this case, the output from the `power status` command is one or more of the following:

Message	Additional Information

`ERROR: CAN bus lock failure`
`ERROR: power shelf query failure`
`ERROR: CAN transaction failure`
`ERROR: CAN bus unlock failure`
	Indicates a problem with the CAN. Power cycle the CMCs, and if the problem persists, contact SGI technical support.

`ERROR: unrecognized CMC/power shelf configuration` `ERROR: unknown fan configuration`	Indicates that the CMC could not determine if it was in a D-Rack or in an M-Rack. Reboot. If the problem persists, contact SGI technical support.
`ERROR: power shelf offline`	Indicates that the CMC did not detect the correct number of power shelves. This could be one of several things. For information, see "Troubleshooting a Missing Power Shelf" on page 234.
`ERROR: too many power supplies offline` `WARNING: power shelf supply offline`	Not currently implemented.
`ERROR: power state save failure`	This rare CMC message can indicate that the file system is full.

```
ERROR: fan query failure
ERROR: no fans available
ERROR: fan start failure
WARNING: fan power supply offline
```

Indicates a problem with the fan controller in the D-Racks.

```
ERROR: no CRC available
ERROR: CRC query failure
ERROR: CRC communication failure
ERROR: CRC temperature failure
ERROR: CRC blower failure
WARNING: CRC temperature alert
```

Indicates a problem with the CRC. This could be a communication error.

```
ERROR: no CDU available
ERROR: CDU communication failure
ERROR: CDU is off
ERROR: CDU valves not auto
ERROR: CDU pump off
WARNING: CDU blower offline
```

Indicates a problem with the CDU. Any of these could indicate a communication error.

7. Type the following command to power down the IRUs:

   > **power off**

8. Type the following command to power up the IRUs:

   > **power on**

9. Use the `pfctl status` command to retrieve the power status of the CMC and the components connected to the CMC.

   Depending on your system configuration, you see output similar to one of the following:

   - On a powered-on SGI ICE X system with M-Racks, the command includes the IP addresses of the cooling racks at the end of the output, as follows:

     > **pfctl status**
     r001i0c: state: ON, demand: MAX watts

```
r001p0s0s0: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s0s1: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s0s2: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s0: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s1: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s2: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s2s0: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s2s1: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s2s2: status: ON PWROK, errors: 0, capacity: UNKNOWN
CRC@172.26.128.1: temp bot (1): GOOD
CRC@172.26.128.1: temp mid (1): GOOD
CRC@172.26.128.1: temp top (1): GOOD
CRC@172.26.128.1: temp bot (2): GOOD
CRC@172.26.128.1: temp mid (2): GOOD
CRC@172.26.128.1: temp top (2): GOOD
CRC@172.26.128.1: blower bot:   GOOD
CRC@172.26.128.1: blower mid:   GOOD
CRC@172.26.128.1: blower top:   GOOD
CDU@172.26.144.1: status: ON
CDU@172.26.144.1: valves: AUTO
CDU@172.26.144.1: pump1:  ON
CDU@172.26.144.1: pump2:  OFF
CDU@172.26.144.1: temp:   24C
```

- On a powered-on SGI ICE X system with D-Racks, the command includes the following information:

```
> pfctl status
r001i0c: state: ON, demand: MAX watts
r002i1c: state: ON, demand: MAX watts
r001p0s0s0: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s0s1: status: ON NOAC, errors: 0, capacity: UNKNOWN
r001p0s0s2: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s0: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s1: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001p0s1s2: status: ON PWROK, errors: 0, capacity: UNKNOWN
r001f0c0: state: ON PWROK RPMOK, errors: 0
r001f0c1: state: ON PWROK RPMOK, errors: 0
r001f0f05:  3583     r001f0f11:  3646
r001f0f04:  3663     r001f0f10:  3646
r001f0f03:  3684     r001f0f09:  3647
r001f0f02:  3627     r001f0f08:  3685
```

```
r001f0f01:  3637     r001f0f07:  3652
r001f0f00:  3663     r001f0f06:  3640
```

10. (Conditional) Issue `ping`(8) commands from the CMC to the IP addresses that appear in the `pfctl status` command output with a status of `NOT FOUND`.

    Complete this step if you are on an SGI ICE X systems with M-Racks and the `pfctl status` command output shows one or more CRCs or CDUs with a status of `NOT FOUND`.

    The CMC calculates the IP addresses of the CRC and CDU based on the CMC's rack and U-position as recorded in the `/etc/sysconfig/`*module_id* file. For each CRC or CDU that appears in the `pfctl status` command output as `NOT FOUND`, issue a `ping`(8) command to the CRC or CDU. Evaluate the output as follows:

    - If the `ping`(8) to the address fails, there could be a physical connection problem between the CRC or CDU and the ethernet switch.
    - If the CMC can ping the CRC or CDU but the `pfctl status` command reports `NOT FOUND`, then the CRC or CDU might not be configured properly to respond to the SNMP requests that the CMC is makes. Either the IP address is not correct or the CRC or CDU is not configured correctly. You should contact SGI technical support.

Power Supplies and the Watchdog Timer

If all the CMCs in the power domain detect a fault condition, then the watchdog timer expires, and the system powers off all the power supplies. When the system is operating as expected, and CMCs detect no faults, the CMCs send a watchdog reset every 10 seconds.

The power shelves must receive a watchdog reset once every 45 seconds from each CMC in the power domain. If the watchdog timer expires, the power shelf controller disables the power supplies on that shelf and sets the WDOG status bit.

The following conditions can prevent the CMC from sending the watchdog reset to the power shelves:

- The CMC cannot confirm that a minimum number of fans are spinning. This pertains to SGI ICE X systems with D-Racks.

- The CMC cannot communicate with the external CRC and/or CDU. The CMC detects a fault condition reported from the CRC and/or the CDU. These conditions pertain to SGI ICE X systems with M-Racks.

The output from the CMC `pfctl status` command shows the status of the WDOG status bit. You can type the `pfctl status` from any CMC in the power domain. The command reports power shelf and supply status and reports fan or CRC/CDU status, depending on rack type.

Interpreting the Power Supply LEDs

The following information explains how to read the status indicators on the power supply LEDs:

If the light is ...	Meaning
Solid green	Power supply is on and OK.
Blinking green	Power supply has AC, but it is not on.
	If all of the power supply LEDs are blinking green, then the power was turned off. This situation could result from the watchdog timer firing or from a power down issued by all CMCs because of a cooling problem, due to either the fan controller on a D-Rack system or the CRC/CDU unit on an M-Rack system.
Solid amber	The power supply has failed.
Blinking amber	This is a power supply warning. The supply is still operating.
	There is no AC to the power supply, but the power supply is plugged into the system.
	There is no AC input (under voltage).

Troubleshooting the Devices on the CANbus Interface

The CANbus is the interface that connects all the CMCs, power shelves, and (in D-Racks) fan controllers. You can use the `pfctl ping` command to retrieve the status of each device and then take corrective action. To use this command, log into the CMC and type the command at the system prompt.

If the `pfctl ping` command reports missing power shelves, see "Troubleshooting a Missing Power Shelf" on page 234.

The following examples show typical output:

Example 1. The following output was obtained on an SGI ICE X system M-Rack, with two IRUs in a power domain:

```
> pfctl ping
PWR-UPPER-CMC1: r1i5c
PWR-UPPER-CMC0: r1i1c
PWR_SHELF3:     -
PWR_SHELF2:     PRESENT
PWR_SHELF1:     PRESENT
PWR_SHELF0:     PRESENT
PWR-LOWER-CMC1: r1i4c
PWR-LOWER-CMC0: r1i0c

EXTERNAL FANS
```

Example 2. The following output was obtained on an SGI ICE X system M-Rack, with one IRU in a power domain and twin node blades:

```
> pfctl ping
PWR-UPPER-CMC1: -
PWR-UPPER-CMC0: r1i4c
PWR_SHELF3:     -
PWR_SHELF2:     PRESENT
PWR_SHELF1:     PRESENT
PWR_SHELF0:     PRESENT
PWR-LOWER-CMC1: -
PWR-LOWER-CMC0: r1i0c
EXTERNAL FANS
```

Example 3. The output in this example is from an SGI ICE X system D-Rack. The fan controller hosts two programmable system on a chip (PSOC) units, and the fan controller controls 12 fans. The following output shows the fan controllers that appear in the `FAN-CONTROL` lines as `PRESENT`, which is correct for a system that is operating properly:

```
> pfctl ping
PWR-UPPER-CMC1: -
PWR-UPPER-CMC0: r1i1c
```

```
PWR_SHELF3:     -
PWR_SHELF2:     -
PWR_SHELF1:     PRESENT
PWR_SHELF0:     PRESENT
PWR-LOWER-CMC1: -
PWR-LOWER-CMC0: r1i0c

FAN-UPPER-CMC1: -
FAN-UPPER-CMC0: r1i1c
FAN-CONTROL1    PRESENT
FAN-CONTROL0    PRESENT
FAN-LOWER-CMC1: -
FAN-LOWER-CMC0: r1i0c
```

Flashing the Firmware on a Power Shelf or Fan Controller

In very rare situations, the power shelf firmware or fan controller firmware can become corrupted. In this situation, the power shelf or the fan controller becomes completely broken or remains perpetually in bootloader mode. If in bootloader mode, the fan controller's fimware can respond to the firmware flashing utility.

The following procedure explains how to flash the firmware.

Procedure 5-15 To flash the firmware

1. Log in to the power shelf or the fan controller.

   For a power shelf, log into the lowest CMC in the power domain.

   For information about how to log into a CMC, see "Power Cycling the IRUs" on page 225.

2. Type the following command to change to the directory that contains the firmware images:

   ```
   > cd /usr/local/firmware/psoc
   ```

3. Use the following command to flash the firmware:

   `flashcan -f` *image* `-p` *target* `-r`

   This variables in this command are as follows:

- For *image*, specify one of the firmware images from the `/usr/local/firmware/psoc` directory.
- For *target*, specify the one of the following:

target	**Hardware to be flashed**
`0`	Power shelf 0
`1`	Power shelf 1
`2`	Power shelf 2
`3`	Power shelf 3
`4`	Fan controller 0
`5`	Fan controller 1

Troubleshooting a Missing Power Shelf

If a power shelf is physically present but does not appear in the `pfctl ping` command output, use the information in the following topics:

- "Booting a Power Shelf Manually" on page 234
- "Fixing Problems Related to a Newly Installed Power Shelf" on page 235

Booting a Power Shelf Manually

Occasionally, when the AC power breakers are enabled, the power shelf controller or the fan controller might not boot properly. The power shelf is said to be *wedged* in this situation. In this case, complete the following procedure to power cycle the AC power to all the power supplies in the power domain or cooling domain.

Procedure 5-16 To boot a power shelf

1. Power cycle the system again.

   Use the procedure in "Power Cycling the IRUs" on page 225.

2. Manually flip the power breakers on the power distribution unit (PDU) at the top of the rack.

3. Type the following command from the CMC:

   ```
   > pfctl ping
   ```

4. Examine the output.

   The output should show the correct number of power shelves as present. The following examples show correct output for their specific systems.

   Example 1. The following output was obtained on an SGI ICE X system D-Rack:

```
> pfctl ping
PWR-UPPER-CMC1: -
PWR-UPPER-CMC0: r1i1c
PWR_SHELF3:     -
PWR_SHELF2:     -
PWR_SHELF1:     PRESENT
PWR_SHELF0:     PRESENT
PWR-LOWER-CMC1: -
PWR-LOWER-CMC0: r1i0c

FAN-UPPER-CMC1: -
FAN-UPPER-CMC0: r1i1c
FAN-CONTROL1    PRESENT
FAN-CONTROL0    PRESENT
FAN-LOWER-CMC1: -
FAN-LOWER-CMC0: r1i0c
```

   Example 2. The following output was obtained on an SGI ICE X system M-Rack, with one IRU in a power domain and single node blades:

```
> pfctl ping
PWR-UPPER-CMC1: -
PWR-UPPER-CMC0: -
PWR_SHELF3:     -
PWR_SHELF2:     PRESENT
PWR_SHELF1:     PRESENT
PWR_SHELF0:     PRESENT
PWR-LOWER-CMC1: -
PWR-LOWER-CMC0: r1i0c

EXTERNAL FANS
```

Fixing Problems Related to a Newly Installed Power Shelf

If you recently added or replaced a power shelf, it is possible that the firmware on the new power shelf was not flashed or there could be a problem with the CANbus

connection. The following procedure explains how to troubleshoot a new power shelf that is not integrated properly.

Procedure 5-17 To integrate a power shelf

1. Log into the lower CMC in the IRU (for systems with M-Racks) or the lower CMC in the pair (for systems with D-Racks).

   For information about how to log into the CMC, see "Power Cycling the IRUs" on page 225.

2. Type the following command to change to the directory that contains the firmware images:

   > **cd /usr/local/firmware/psoc**

3. Use the following command to flash the firmware:

   `flashcan -f` *image* `-p` *controller* `-r`

   This variables in this command are as follows:

   - For *image*, specify one of the firmware images from the `/usr/local/firmware/psoc` directory.
   - For *controller*, specify one of the following:

controller	Target
0	Power shelf 0
1	Power shelf 1
2	Power shelf 2
3	Power shelf 3
4	Fan controller 0
5	Fan controller 1

4. Type the `pfctl ping` command to retrieve the status of the power shelf.

   If the status returns `PRESENT` for the problem power shelf, you are finished. If the `pfctl ping` command does not return `PRESENT`, continue with this procedure to troubleshoot other causes of the problem.

5. Perform one or more of the following remedies:

   - Re-seat the power shelf.

- Visually inspect the connector on the shelf and make sure that it is correct.

  Inspecting the blind connector at the rear of the power shelf slot can be difficult to do.

- Visually inspect the LED lights.

  If there were power supplies in the shelf with the fail LED lite, there is a remote chance that the failed supply did some damage to the power shelf. If a power supply turns on immediately when the AC power is applied, the shelf itself might be damaged. If the power supplies in all of the other shelves are off (flashing green) and the supplies in the missing shelf turn solid green as soon as the breaker is enabled, then the power shelf is probably bad. Find the sticker on the power shelf, and see if it is discolored.

- Inspect the CANBus cable in the back on the rack.

Log Files

The following log files contain information that can help you troubleshoot a power problem:

- The `/tmp/pfctld.log` file contains entries from the power and fan control daemon, `pfctld`. When the `pfctld` daemon powers down an IRU, it records a log entry in the log file. The entry includes the reason for the power down.
- The `/tmp/eric.log` file contains output from an enviornmental software monitoring appliation, called ERIC, that runs on the CMC. ERIC's actions are written to this log file. ERIC monitors blade temperatures and adjusts fans speeds appropriately. ERIC also monitors the CMC inlet air temperature and powers down the IRU when appropriate. That is, ERIC powers down the blades associated with that CMC.

  If your SGI ICE X system has D-Racks, and the following conditions are all present, the problem might be related to the CMC air inlet remperature:

  – Only one IRU's blades are powered down

  – The blades in the other IRU are still on

  – Power supply LEDs are solid green

  In an M-Rack configuration, ERIC could power off only the upper or lower board in the blade if there is a problem with the CMC air inlet temperature.

Retrieving Information About the Power Supplies

After you log into the CMC, you can use the `pmbus_drack` and `pmbus_mrack` scripts to retrieve information about the power supplies. These scripts dump some of the PMBus data that is available.

The following example shows output from the `pmbus_mrack` script:

```
> pmbus_mrack
Shelf0 PS0 Vout: 12         Iout: 3.5       Temp: 29        Status: 0x0000
Shelf0 PS1 Vout: 11.6875    Iout: 0         Temp: 29.5      Status: 0x0000
Shelf0 PS2 Vout: 12.0312    Iout: 1         Temp: 28.5      Status: 0x0000
Shelf1 PS0 Vout: 11.625     Iout: 0         Temp: 29        Status: 0x0000
Shelf1 PS1 Vout: 11.6562    Iout: 0         Temp: 29.5      Status: 0x0000
Shelf1 PS2 Vout: 11.6875    Iout: 0         Temp: 28.5      Status: 0x0000
Shelf2 PS0 Vout: 11.6875    Iout: 0         Temp: 28.5      Status: 0x0000
Shelf2 PS1 Vout: 11.625     Iout: 0         Temp: 28        Status: 0x0000
Shelf2 PS2 Vout: 11.6562    Iout: 0         Temp: 29        Status: 0x0000
Shelf3 PS0 Vout:            Iout:           Temp:           Status: not available
Shelf3 PS1 Vout:            Iout:           Temp:           Status: not available
Shelf3 PS2 Vout:            Iout:           Temp:           Status: not available
```

The output shows the output voltage (`Vout`), the output current (`Iout`), and temperature from each of the power supplies.

You can use the `Iout` values to determine whether the power supply load sharing is working correctly within the power domain. Generally all power supplies should be +/- 10% of the average. The status bits record warnings and faults when they occur. All bits are decoded if present.

The following status messages can appear in the `pmbus_mrack` output:

Message	Meaning
`VOUT`	Output voltage warning or fault.
`IOUT`	Output current warning or fault.
`INPUT`	Input fault.
`MFR`	Manufacturer fault. Generally related to the 3.3v auxialliary supply used to power the power shelf controllers, fan shelf controllers, and the CMCs.
`PWRGOOD`	Power output is good (active low).

FANS	Indicates an internal fan failure.
OTHER	Another warning or fault not indicated by other status flags.
UNKNOWN	An internal power supply controller condition was detected.
OFF	Power supply is off.
VOUT_OV	Output voltage over limit.
IOUT_OC	Output current over limit.
VIN_UC	Input voltage under limit.
TEMP	Temperature warning or fault.
CML	Communication error. Can be ignored.
NOTA	None of the above.

Power supplies shut down on any fault condition and remain off unless the fault is a temperature fault. After the power supply has cooled, it reenables itself. Generally, if you cycle the AC power to the faulted power supply, it resets all status flags. Hard failures should reoccur. If the system is under heavy load and a power supply fails, the other supplies pick up the load. If yet another supply fails, this can cause an overcurrent across all supplies, which in turn, powers down all compute blades in the power domain.

Retrieving Information About the PMBus Registers

After you log into the CMC, you can use the `pfctl pmbus dump` command to retrieve information about the PMBus registers. This command queries all power supplies in the power domain. In the command's output, look for nonzero readings to locate possible problems.

The following example shows output from the `pfctl pmbus dump` command:

```
> pfctl pmbus dump
PWR s0s0                VIN:   213.00
PWR s0s0                IIN:     0.78
PWR s0s0               VOUT:    11.97
PWR s0s0               IOUT:     9.00
PWR s0s0          3.3 VOUT:     3.34
PWR s0s0          3.3 IOUT:     1.34
PWR s0s0               TEMP:    23.00
PWR s0s0          FAN1 RPM: 6320
PWR s0s0          FAN2 RPM: 6320
```

```
PWR s0s0        STATUS_BYTE: 00
PWR s0s0        STATUS_WORD: 0000
PWR s0s0        STATUS_VOUT: 00
PWR s0s0        STATUS_IOUT: 00
PWR s0s0       STATUS_INPUT: 00
PWR s0s0 STATUS_TEMPERATURE: 00
PWR s0s0         STATUS_CML: 00
PWR s0s0         STATUS_3V3: 00
PWR s0s0    STATUS_FANS_1_2: 00
PWR s0s0         SMB ALERT : 00 NO
PWR s0s0 SOFTWARE REVISION : pri 167 app 169 boot 2
PWR s0s0              PMBUS: I 1 II 1
PWR s0s0                 ID: DELTA
PWR s0s0              MODEL: AHF-2DC-2837W-12V-240V
PWR s0s0           REVISION: 1 6 167 169
PWR s0s0           LOCATION: DES
PWR s0s0               DATE: 23/10 (13:58:00 06/08/10)
PWR s0s0             SERIAL: A000379
```

Troubleshooting

The following topics explain how to troubleshoot some known problems:

- "dbdump Command" on page 240
- "system_info_gather Command" on page 242
- "cminfo Command" on page 243

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 admin node.

- 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 5-1 `dbdump` Command Examples

To dump the entire database, perform the following:

```
admin:~ # dbdump
0 is { cluster=oscar ifname=service0-bmc dev=bmc0 ip=172.24.0.3 net=head-bmc node=service0
  nodetype=oscar_service mac=00:30:48:8e:
1 is { cluster=oscar ifname=service0 dev=eth0 ip=172.23.0.3 net=head node=service0
  nodetype=oscar_service mac=00:30:48:33:53:2e }
2 is { cluster=oscar ifname=service0-ib0 dev=ib0 ip=10.148.0.2 net=ib-0 node=service0
  nodetype=oscar_service }
3 is { cluster=oscar ifname=service0-ib1 dev=ib1 ip=10.149.0.2 net=ib-1 node=service0
  nodetype=oscar_service }
4 is { cluster=oscar dev=eth0 ip=128.162.244.86 net=public node=oscar_server
  nodetype=oscar_server mac=00:30:48:34:2B:E0 }
...
```

Note: Some of the sample output in this section has been modified to fit the format of this manual.

To dump just the 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 }
...
```

`system_info_gather` Command

The `system_info_gather` command collects system data that you can use to troubleshoot problems. The command collects information about the following:

- Digital media `dminfo` files, system logs, Dynamic Host Configuration Protocol (DHCP), network file system (NFS)
- MySQL / MariaDB cluster database dump
- Network service configuration files, for example, C3, Ganglia, DHCP, domain name service (DNS) configuration files
- Installed system images

- Log files in `/var/log/messages`
- Chassis management control (CMC) slot table information 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 `system_info_gather` command, type **`system_info_gather -h`**.

`cminfo` Command

The `cminfo` command is used internally by many 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 5-2 `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
```

About the kdump Utility

You can download the kernel RPMs, for use with the crash package, from either RHEL or SLES. For RHEL, you can download the Debuginfo package from Red Hat Network (RHN) and incorporate the package into your node images and running nodes. On SLES, the kdump facility is enabled and working by default. For information about how to add additional packages to your RPM lists and your softawre images, see the following:

"Using a Custom Repository for Site Packages" on page 140

The kdump utility is a kexec-based crash dumping mechanism for the Linux operating system. By default, the kdump crash dump capability is enabled on SGI ICE X systems after installation.

The following topics provide more information about kdump:

Obtaining a Traceback or System Dump

You can obtain a system dump from an SGI ICE compute node, a rack leader controller (RLC) node, or a compute node.

For the admin node, this information is located on the admin node in the following locations:

- Traceback information is in `/net/r1lead/var/log/consoles`.
- System dump information is in `/net/r1lead/var/log/dumps/r1i0n0`.

For an SGI ICE compute node, RLC, or compute node, log into the admin node, bring up a console to the node from which you want to obtain a crash dump, and type the following:

```
^e c l 1 8
^e c l 1 t       #traceback
^e c l 1 c       #dump
```

For example, to obtain information from an RLC, type the following:

```
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.

Retrieving the current kdump Setting

The following procedure explains how to retrieve the current `kdump` setting for a specific system image.

Procedure 5-18 To retrieve the current `kdump` setting

1. Log into the admin node as the root user.
2. Use the `cadmin` command in the following format to retrieve the current `kdump` memory allocation:

   `cadmin --show-crashkernel --image` *image_name*

   For *image_name*, specify the name of one of the SGI ICE compute node, rack leader controller (RLC) node, or compute node operating system images.

Disabling kdump

When kdump is enabled, the system reserves some memory for crash dumps. If you want to make this memory available to user programs, you can disable the kdump facility. You can also reduce the size of the memory used for the kdump facility.

The following procedure explains how to disable kdump.

Procedure 5-19 To disable kdump

1. Log into the admin node as the root user.
2. Use the cadmin command in the following format to disable the kdump facility:

   `cadmin --set-crashkernel --image` *image_name* `""`

   For *image_name*, specify the name of one of the SGI ICE compute node, rack leader controller (RLC) node, or compute node operating system images.

   Type two quotation mark characters at the end of the command to represent the empty string.
3. (Conditional) Push the changes to the system.

   Complete this step if you specified an SGI ICE compute node image for *image_name* in the preceding step.

   Type the following command:

   `cadmin --push-rack` *image_name*

   For *image_name*, specify the same *image_name* you specified in the preceding `cadmin --set-crashkernel` command.

For example:

```
# cadmin --set-crashkernel --image ice-sles11sp3 ""
# cadmin --push-rack ice-sles11sp3
```

Setting a Site-specific kdump Value

The following procedure explains how to specify the amount of memory you want to devote to kdump.

Procedure 5-20 To specify the amount of kdump memory

1. Log into the admin node as the root user.
2. Use the cadmin command in the following format to specify the amount of memory to use for the kdump facility:

   `cadmin --set-crashkernel --image` *image_name* "*mem_size*"

   For *image_name*, specify the name of one of the SGI ICE compute node, rack leader controller (RLC) node, or compute node operating system images.

   For *mem_size*, specify an amount of memory.
3. (Conditional) Push the changes to the system.

   Complete this step if you specified an SGI ICE compute node image for *image_name* in the preceding step.

   Type the following command:

   `cadmin --push-rack` *image_name*

   For *image_name*, specify the same *image_name* you specified in the preceding `cadmin --set-crashkernel` command.

For example:

```
# cadmin --set-crashkernel --image sles11sp3 "512M"
```

For more information about setting the `--set-crashkernel` boot parameter, see the kdump(7) man page.

Resetting the kdump Value to the System Default

If you disable kdump or reset the amount of memory for kdump, you can reset the value to the system default value. To retrieve the system default value specify the following command:

The following procedure resets the kdump value to the system defaults.

Procedure 5-21 To reset the kdump memory reservation to the system default value

1. Log into the admin node as the root user.
2. (Optional) Display a list of images and display the default crashkernel value.

Type the following commands:

```
cimage --list-images
cadmin --show-crashkernel --image image_name
```

For *image_name*, specify the one of the images that the `cimage` command displayed.

For example:

```
# cimage --list-images
image: ice-rhel6.5
        kernel: 2.6.32-431.el6.x86_64
# cadmin --show-crashkernel --image ice-rhel6.5
crashkernel=256M
```

3. Use the `cadmin` command in the following format to specify the amount of memory to use for the `kdump` facility:

   ```
   cadmin --set-crashkernel --image image_name
   ```

   For *image_name*, specify the name of one of the SGI ICE compute node, rack leader controller (RLC) node, or compute node operating system images.

   Note that in this format, you do not specify the empty string, nor do you specify a string that contains a memory size.

4. (Conditional) Push the changes to the system.

   Complete this step if you specified an SGI ICE compute node image for *image_name* in the preceding step.

   Type the following command:

   ```
   cadmin --push-rack image_name
   ```

   For *image_name*, specify the same *image_name* you specified in the preceding `cadmin --set-crashkernel` command.

For example:

```
# cadmin --set-crashkernel --image sles11sp3
```

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 admin node, the following command gets the BMC firmware revision for `r1i0n0`:

```
# ipmiwrapper r1i0n0 bmc info | grep 'Firmware Revision'
```

CMC Version Interrogation

The CMC firmware version can 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 admin node 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
```

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

I

K

L

M

N

O

P

S