SGI® ICE™ X Administration Guide

007–5918–001

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

c3:

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

Record of Revision

Version	Description
001	May 2013 Original publication. This revision supports the SGI Management Center 1.7 release.

Contents

Figures

Examples

Procedures

About This Guide

This guide is a reference document for people who administer SGI® ICE™ X systems. It describes how to perform general system operations.

Related Publications

The following additional documentation might be useful to you:

- *SGI ICE X Installation and Configuration Guide*

  This manual explains how to install and configure SGI ICE X systems.

- *SGI Management Center Installation and Configuration*

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

- *SGI Management Center System Administrator's Guide*

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

- *SGI ICE X System Hardware User Guide*

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

- *SGI Performance Suite X.X Start Here*

  This manual lists the current SGI software and hardware manuals.

- Documentation from other sources:

  – Novell documentation for SUSE Linux Enterprise Server 11 (SLES 11)

  – Red Hat documentation for Red Hat Linux Enterprise Server 6 (RHEL 6)

  – Intel compiler documentation

– Intel documentation about Xeon architecture

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at: http://docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- Online versions of the *SGI Performance Suite X.X Start Here*, release notes, which contain the latest information about software and documentation for each SGI Performance Suite product, the list of RPMs distributed with each product can be found in the `/docs` directory on each SGI Performance Suite product media.
- You can view man pages by typing `man` *title* on a command line.

Conventions

The following conventions are used throughout this document:

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

...	Ellipses indicate that a preceding element can be repeated.

Reader Comments

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

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  SGI
  Technical Publications
  46600 Landing Parkway
  Fremont, CA 94538

SGI values your comments and will respond to them promptly.

System Operation

This chapter describes how to operate your SGI ICE system and covers the following topics:

Changing Global Cluster Configuration Settings

This topic explains how to use the cluster configuration tool 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 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.

The following topics explain how to change global cluster configuration settings through this menu or by using commands.

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 1-1 To change the NTP server information

1. From the video graphics array (VGA) screen, or through an `ssh` connection, log into the system admin controller (SAC) 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 1-2 To change the DNS server information

1. From the VGA screen, or through an `ssh` connection, log into the system admin controller (SAC) 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 system admin controller (SAC) provides name services for the SGI ICE X system. When you configure a backup DNS, however, the compute nodes can use a service node as a secondary DNS server if the SAC 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.

If you want to use the cluster configuration tool to enable or disable the backup DNS, see the *SGI ICE X Installation and Configuration Guide*.

You can also use commands to enable or disable this feature. The following examples show how to use the commands:

- 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
  done
  ```

```
sys-admin: update-configs: updating SMC for SGI ICE X configuration files
sys-admin: update-configs: -> dns
...
```

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

```
# /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 for SGI ICE X configuration files
sys-admin: update-configs: -> dns
  ...
```

Configuring the InfiniBand Fabric

The InfiniBand network includes two subnetworks, `ib0` and `ib1`. The following chapter contains information about the InfiniBand networks:

Chapter 2, "InfiniBand Fabric Management" on page 99

Configuring a Redundant Management Network (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 service nodes is 802.3ad link aggregation. The RMN has the following additional characteristics:

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

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

The following procedure explains how to configure an RMN from the cluster configuration tool.

Procedure 1-3 To enable the RMN from the cluster configuration tool

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

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

   Perform this step only if the cluster configuration tool is not running at this time.
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 `mgmt_bonding=TYPE` parameter and with the `cadmin` command's `--set-mgmt-bonding` parameter. If you use the `cadmin` command to change a service node or a leader node, reboot the node to make your changes take effect. The following examples show how to use commands to configure the RMN.

Example 1. The following `discover` command disables the RMN on node `service0`:

```
# discover --service0,xe500,redundant_mgmt_network=no
```

Example 2. The following `cadmin` command enables the RMN on node `service0`:

```
# cadmin --set-redundant-mgmt-network --node service0 yes
```

Example 3. The following `cadmin` command enables the RMN on RLC `r1lead` and shows the required subsequent reboot:

To turn on the redundant management network on an RLC, perform the following command:

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

Configuring MySQL Replication

SGI ICE X systems store cluster information in an internal MySQL database. MySQL replication is enabled by default on all SGI ICE X systems.

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

In some situations, however, you might need to disable MySQL replication, either for the entire system or only for selected nodes.

To verify whether MySQL database replication is working on an RLC or service node, type the following command:

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

For information about how replication is implemented and configured, 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 MySQL replication:

- "Disabling MySQL Replication" on page 6
- "Enabling MyQSL Replication" on page 7

Disabling MySQL Replication

By default, the SGI ICE X system uses an internal MySQL database, and SGI recommends that you keep MySQL database replication enabled. This practice keeps the internal cluster database synchronized. If the system hosts other software that cannot be used when database replication is enabled, you can disable the MySQL database replication on a particular node. When you disable synchronization on a specific node, that node uses the system admin controller (SAC) for database queries.

For example, if the database becomes corrupt, you can disable replication on the entire SGI ICE X system during the debugging session and reenable it later.

The following procedure explains how to disable MySQL database replication.

Procedure 1-4 To disable MySQL database replication on a service node

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

   ```
   admin:~ # cattr set --node service0 my_sql_replication no
   ```

3. Type the following command to confirm that database replication is disabled:

   ```
   admin:~ # ssh service0 /etc/opt/sgi/conf.d/80-update-mysql
   Shutting down service MySQL ..done
   mysql                    0:off  1:off  2:off  3:off  4:off  5:off 6:off
   ```

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

   ```
   admin:~ # cattr set --node service0 ignore_my_sql_replication yes
   ```

   At this point, if you run `80-update-mysql` again, you are returned to the system prompt. Unlike the example in the previous step, the command does not issue any messages.

5. Configure software on the service node.

Procedure 1-5 To disable MySQL replication on an SGI ICE X system

1. From the VGA screen, or through an `ssh` connection, log into the system admin controller (SAC) 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 MySQL Replication (optional)**, and select **OK**.
4. On the pop-up window that appears, select **N no**, and select **OK**.

Enabling MyQSL Replication

The following procedure explains how to enable MySQL database replication.

Procedure 1-6 To enable MySQL database replication from the cluster configuration tool

1. From the VGA screen, or through an `ssh` connection, log into the system admin controller (SAC) 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 MySQL Replication (optional)**, and select **OK**.
4. On the pop-up window that appears, select **Y yes**, and select **OK**.

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

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

Procedure 1-7 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 system admin controller (SAC) 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`**

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

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**.

Configuring the blademond Rescan Interval

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

Procedure 1-8 To configure the blademond rescan interval from the cluster configuration tool

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

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

   Perform this step only if the cluster configuration tool is not running at this time.
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**.
5. On the **Main Menu** screen, select **U Configure Default Max Rack IRU Setting (optional)**, and select **OK**.
6. Visually inspect the pop-up window that appears and verify that the maximum IRU setting is appropriate for your system.

   On SGI ICE X platforms, set to this value to 8.

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

   For more information about this setting, see "Configuring the Default Maximum Rack Individual Rack Unit (IRU) Setting" on page 8.

`discover` Command

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

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

When you use the `discover` command to perform the discovery operation on your SGI ICE X system, you will be prompted with instructions on how to proceed.

When using the `--delrack` and `--delservice` options, the node is not removed completely from the database but it is marked with the administrative status `NOT_EXIST`. When you go to discover a node that previously existed, you now get the same IP allocations you had previously and the node is then marked with the administrative status of `ONLINE`. If you have a service node, for example, `service0`, that has a custom host name of "`myhost`" and you later go to delete `service0` using the `discover --delservice` command, the host name associated with it will still be present. This can cause conflicts if you wish to reuse the custom host name "`myhost`" on a node other than service0 in the future. You can use the `cadmin --db-purge --node service0` command that will remove the node entirely from the database (for more information, see "`cadmin`: SMC for SGI ICE X Administrative Interface" on page 66). You can then reuse the "`myhost`" name.

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

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

- The management switches should be the first hardware discovered in the system.

- The management switches should both start with their power cords unplugged (analogous to how SMC for SGI ICE X discovers RLCs and service nodes).
- The external switches can be given higher numbered service numbered if your site does not want them to take lower numbers.
- You can also elect to give these switches an alternate host name using the `cadmin` command after discovery is complete.
- Examples of using the `discover` command `generic` hardware type are, as follows:

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

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

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

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

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

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

```
Details:
  Any number of management switches, racks, service nodes, or external
  switches can be discovered in one command line.  Rack numbers generally
  start at 1, service nodes, management switches, and infiniband switches
  generally start at 0.  An existing node can be re-discovered by re-running
  the discover command. An easier way to simply re-image a node is by
  using the cinstallman command, see the --next-boot and --assign-image
  options.

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

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

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

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

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

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

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

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

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

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

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

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

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

  Discover a switch used to extend the Tempo management network - a generic
```

```
device.
 # discover --service 99,generic

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

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

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

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

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

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

EXAMPLES

Example 1-1 `discover` Command Examples

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

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

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

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

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

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

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

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

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

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

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

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

Note: You may direct a service node to image itself with a custom image later, without re-discovering it. See "cinstallman Command" on page 26.

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

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

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

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

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

An example command is, as follows:

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

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

To remove a switch, perform the following:

```
admin:~ # discover --delibswitch name
```
where *name* is that of a previously discovered switch.

An example command is, as follows:

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

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

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

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

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

Managing a Multiboot System

The following topics explain how to manage a multiboot system:

- "Managing the Boot Slot and Changing the Boot Slot" on page 16
- "Cloning a Slot" on page 17
- "Customizing the Slot Labels on a Multiboot System" on page 18

Managing the Boot Slot and Changing the Boot Slot

If you configured more than one slot, you can boot from the boot partition in any of the slots. The following procedure explains how to change the system to boot from a different slot.

Procedure 1-9 To change the boot partition and enable the system to boot from a different slot

1. Log in as the root user to the system admin controller (SAC).
2. Type the following command to verify the current boot slot:

   ```
   # cadmin --show_root-labels
   admin node currently booted on slot: 1
   ```

3. (Optional) Change the default slot.

   Perform this step if you know the slot from which you want to boot.

   You can specify the new slot now, or you can specify the new slot during the reboot.

   # **cadmin --set-default-root --slot *num***

For *num*, specify the new boot slot number. *num* can be `1`, `2`, `3`, `4`, or `5`.

For example, to specify a boot from slot 2, type the following:

```
admin:~ # cadmin --set-default-root --slot 2
```

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

   ```
   # cpower --shutdown --system
   ```

5. Type the following command to reboot the SAC:

   ```
   # reboot
   ```

6. Monitor the reboot and, optionally, select the slot from which you want to boot.

   During the reboot, the system displays a screen that shows all the available slots and highlights the current boot slot. If you need to select a different boot slot, use the arrow keys to select a new slot from which to boot and press `Enter`.

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

7. Log in as the root user again.

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

   ```
   # cpower --reboot --system
   ```

   If the IP addresses are configured differently within different slots, the `cpower` command might not be able communicate with the baseboard management controllers (BMC)s immediately after you reboot the SAC. If you have trouble connecting to the RLC and service node BMCs after you change slots, wait up to 15 minutes and issue the `cpower` command again. The wait enables the nodes to obtain new IP addresses.

Cloning a Slot

A script named `/opt/sgi/sbin/clone-slot` is available. This script allows you to clone a source slot to a destination slot. It then handles synchronizing the data and fixing up `grub` and `fstabs` to make the cloned slot a viable booting choice.

The script sanitizes the input values, then calls a worker script in parallel on all managed nodes and the system admin controller (SAC) that does the actual work. The `clone-slot` script waits for all children to complete before exiting.

Important: If the slot you are using as a source is the mounted/active slot, the script will shut down `mysql` on the SAC prior to starting the backup operation and start it when the backup is complete. This ensures there is no data loss.

Customizing the Slot Labels on a Multiboot System

You can use the `cadmin` command to label the slots on a multiboot SGI ICE X system. After an installation, the slot label is `(none)`.

Procedure 1-10 To customize the slot labels

1. Log into the system admin controller (SAC) as the root user.
2. Type the following command to retrieve the current labels:

   ```
   admin:~ # cadmin --show-root-labels
   ```

3. Type the command again, in the following format, to specify the slot and the label:

   `cadmin --show-root-labels --slot` *num* `--label "`*mylabel*`"`

   For *num*, type `1`, `2`, `3`, `4`, or `5` to specify the slot you want to label.

   For *mylabel*, type the label you want to apply to the slot.

   For example:

   ```
   # cadmin --set-root-label --slot 1 --label "SLES"
   # cadmin --show-root-labels
   slot 1: SMC for ICE x.x / sles11: SLES
   slot 2: SMC for ICE x.x / sles11: RHEL
   slot 3: SMC for ICE x.x / sles11: SLES-2
   ```

Software Image Management

This section describes image management operations.

This section describes Linux services turned off on compute nodes by default, how you can customize the software running on compute nodes or service nodes, create a simple clone image of compute node or service node software, how to use the `cimage` command to push images to compute nodes, how to use `crepo` command to manage software image reposistories, and how to use the `cinstallman` command to create compute and service node images. It covers these topics:

Finding Which Distributions (Distros) Are Supported

To find a list of distributions supported on SGI ICE X nodes, perform the following commands from the system admin controller (SAC), as follows:

```
sys-admin:~ # cd /opt/sgi/share/rpmlists/distro/
```

```
sys-admin:/opt/sgi/share/rpmlists/distro # ls
compute-distro-centos5.4.rpmlist  lead-distro-rhel6.1.rpmlist
compute-distro-centos5.5.rpmlist  lead-distro-rhel6.2.rpmlist
compute-distro-centos6.0.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist    service-distro-centos5.5.rpmlist
compute-distro-rhel5.5.rpmlist    service-distro-centos6.0.rpmlist
compute-distro-rhel5.6.rpmlist    service-distro-rhel5.4.rpmlist
compute-distro-rhel5.7.rpmlist    service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist    service-distro-rhel5.6.rpmlist
compute-distro-rhel6.1.rpmlist    service-distro-rhel5.7.rpmlist
compute-distro-rhel6.2.rpmlist    service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist  service-distro-rhel6.1.rpmlist
compute-distro-sles10sp4.rpmlist  service-distro-rhel6.2.rpmlist
compute-distro-sles11sp1.rpmlist  service-distro-sles10sp3.rpmlist
lead-distro-centos6.0.rpmlist     service-distro-sles10sp4.rpmlist
lead-distro-rhel6.0.rpmlist       service-distro-sles11sp1.rpmlist
```

Operating Systems Supported per Node Type

This section describes what operating systems are supported for various SGI ICE X nodes.

System Admin Controller (SAC)

The SAC supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Rack Leader Controller (RLC)

The RLC supports the following operating systems: SLES 11 SP1, RHEL 6.0, CENTOS 6.0, RHEL 6.1 and RHEL 6.2.

Service Nodes

Service nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5

Compute Nodes

Compute nodes support the following operating systems, as follows:

- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4
- SLES 11 SP1, SLES 10 SP4, SLES 10 SP3, RHEL 6.2, RHEL 6.1, RHEL 6.0, RHEL 5.7, RHEL 5.6, RHEL 5.5, RHEL 5.4, CENTOS 6.0, and CENTOS 5.5, and CENTOS 5.4

Compute Node Services Turned Off by Default

To improve the performance of applications running MPI jobs on compute nodes, most services are disabled by default in compute node images. To see what adjustments are being made, view the `/etc/opt/sgi/conf.d/80-compute-distro-services` script.

If you wish to change anything in this script, SGI suggests that you copy the existing script to `.local` and adjust it there. Perform the following commands:

```
# cd /var/lib/systemimager/images/compute-image-name
# cp etc/opt/sgi/conf.d/80-compute-distro-services 80-compute-distro-services.local
# vi etc/opt/sgi/conf.d/80-compute-distro-services.local
```

At this point, the configuration framework will execute the `.local` version, and skip the other. For more information on making adjustments to configuration framework files, see "SGI ICE X System Configuration Framework" on page 50.

Use the `cimage` command to push the changed image out to the rack leader controllers (RLCs).

`crepo` Command

You can use the `crepo` command to manage software repositories such as SGI Foundation, SMC, SGI Performance Suite, and the Linux distribution(s) you are using on your system. You also use the `crepo` command to manage any custom repositories you create yourself.

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

Each repository has associated with it a name, directory, update URL, selection status, and suggested package lists. The update URL is used by the `sync-repo-updates` command. 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, and is located in one of these locations, as follows:

Repository	Description
`/tftpboot/sgi/*`	For SGI media
`/tftpboot/other/*`	For any media that is not from SGI
`/tftpboot/distro/*`	For Linux distribution repositories such as SLES or RHEL

`/tftpboot/`*x*

Customer-supplied repositories

The repository information is determined from the media itself when adding media supplied by SGI, Linux distribution media (SLES, RHEL, and so on.), and any other YaST-compatible media. For customer-supplied repositories, the information must be provided to the `crepo` command when adding the repository.

Repositories can be selected and deselected. Usually, SMC 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-10-SP3
Updating: /etc/opt/sgi/rpmlists/generated-compute-sles10sp3.rpmlist
Updating: /etc/opt/sgi/rpmlists/generated-service-sles10sp3.rpmlist
```

When generating the RPM lists, the `crepo` command combines the a list of distribution RPMs with suggested RPMs from every other selected repository. The distribution RPM lists are usually read from the `/opt/sgi/share/rpmlists/distro` directory. For example, the compute node RPM list for `sles11sp1` is `/opt/sgi/share/rpmlists/distro/compute-distro-sles11sp1.rpmlist`. The suggested RPMs for non-distribution repositories are read from the `/var/opt/sgi/sgi-repodata` directory. For example, the `rpmlist` for SLES 11 SP1 compute nodes is read from `/var/opt/sgi/sgi-repodata/SMC-for-ICE 1.5-for-Linux-sles11/smc-ice-compute.rpmlist`.

The suggested `rpmlist`s can be overridden by creating an override `rpmlist` in the `/etc/opt/sgi/rpmlists/override/` directory. For example, to change the default SMC for ICE 1.5 suggested RPM list, a file `/etc/opt/sgi/rpmlists/override/SMC-for-ICE-1.5 -for-Linux-sles11/smc-ice-compute.rpmlist` can be created.

The following example shows the contents of the `/etc/opt/sgi/rpmlists` directory after the `crepo` command has created the suggested RPM lists. Change

directory (cd) to the /etc/opt/sgi/rpmlists directory. Use the ls command to see a list of RPMs, as follows:

```
admin distro]# ls
compute-distro-centos5.4.rpmlist  lead-distro-sles11sp1.rpmlist
compute-distro-rhel5.4.rpmlist    service-distro-rhel5.4.rpmlist
compute-distro-rhel5.5.rpmlist    service-distro-rhel5.5.rpmlist
compute-distro-rhel6.0.rpmlist    service-distro-rhel6.0.rpmlist
compute-distro-sles10sp3.rpmlist  service-distro-sles10sp3.rpmlist
compute-distro-sles11sp1.rpmlist  service-distro-sles11sp1.rpmlist
lead-distro-rhel6.0.rpmlist
```

Specifically, SMC software looks for /etc/opt/sgi/rpmlists/generate-*.rpmlist and creates an image for each rpmlist that matches.

It also determines the default image to use for each node type by hard-coding $nodeType-$distro as the type, where distro is the system admin controller's (SAC's) distro and nodeType is compute, service, rack leader controller (RLC), and so on. The default image can be overridden by specifying a global cattr attribute named image_default_$nodeType; for example, image_default_service. Use cattr --h, for information about the cattr command.

The following example shows the contents of the /etc/opt/sgi/rpmlists directory after the crepo command has created the suggested RPM lists. The files with -distro- in the name are the base Linux distro RPMs that SGI recommends.

Use the cd(1) to change to the /etc/opt/sgi/rpmlists directory. Use the ls command to see a list of RPMs, as follows:

```
admin:/etc/opt/sgi/rpmlists # ls
compute-minimal-sles11sp1.rpmlist  generated-lead-rhel6.2.rpmlist
generated-compute-rhel6.2.rpmlist  generated-service-rhel6.2.rpmlist
```

For more information on rpmlist customization information, see "Creating Compute and Service Node Images Using the cinstallman Command" on page 46.

You can use the crepo --show command to show the available repositories on the SAC, as follows:

```
sys-admin:~ # crepo --show
* SGI-Management-Center-1.5-rhel6 : /tftpboot/sgi/SGI-Management-Center-1.5-rhel6
* SGI-Foundation-Software-2.5-rhel6 : /tftpboot/sgi/SGI-Foundation-Software-2.5-rhel6
* SGI-XFS-XVM-2.5-for-RHEL-rhel6 : /tftpboot/sgi/SGI-XFS-XVM-2.5-for-RHEL-rhel6
```

```
* SGI-Accelerate-1.3-rhel6 : /tftpboot/sgi/SGI-Accelerate-1.3-rhel6
* SGI-Tempo-2.5-rhel6 : /tftpboot/sgi/SGI-Tempo-2.5-rhel6
* SGI-MPI-1.3-rhel6 : /tftpboot/sgi/SGI-MPI-1.3-rhel6
* Red-Hat-Enterprise-Linux-6.2 : /tftpboot/distro/rhel6.2
```

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

```
admin:~ # crepo --h
crepo Usage:
Operations:
--help                : print his usage message

--add {path/URL}      : add SGI/SMC media to the system repositories
       --custom {name}: Optional.Use with -add to add custom repo under
                        /tftpboot Repo must pre-exist for this case.

--del {product}       : delete an add-on product and associated /tftpboot repo

--select {product}    : mark the product as selected

--show                : show available add-on products

--show-distro         : like show, but only reports distro media like sles10sp2

--show-updateurls     : Show the update sources associated add-on products

--reexport            : re-export all repositories with yume.  Use if there
                        was a yume export problem previously.

--unselect {product}  : mark the product as not selected


Flags:

Note for --add: If the pathname is local to the machine, it can be an
ISO file or mounted media.  If a network path is used -- such as an nfs
path or a URL -- the path must point to an ISO file.  The argument to
--add may be a comma delimited list to specify multiple source media.

Use --add for SGI/SMC media, to make the repos and rpms available. If the
supplied SGI/SMC media has suggested rpms from SMC node types, those
suggested rpms will be integrated with the default rpmlists for leader,
```

```
service, and compute nodes.  You can use create-default-sgi-images to
re-create the default images including new suggested packages or you can
just browse the updated versions in /etc/opt/sgi/rpmlists.

Use --add with --custom to register your own custom repository.  This will
ensure that, by default, the custom repository is available to yume and
mksiimage commands.  It is assumed you will maintain your own default package
lists, perhaps using the sgi default package lists in /etc/opt/sgi/rpmlists
or /opt/sgi/share/rpmlists as a starting point.  The directory and rpms within
must pre-exist.  This script will create the yum metadata for it.
Example:
crepo --add /tftpboot/myrepo --custom my-custom-name
```

cinstallman Command

The cinstallman command is a wrapper tool for several SMC operations that previously ran separately. You can use the cinstallman command to perform the following:

- Create an image from scratch
- Clone an existing image
- Recreate an image (so that any nodes associated with said image prior to the command are also associated after)
- Use existing images that may have been created by some other means
- Delete images
- Show available images
- Update or manage images (via yume)
- Update or manage nodes (via yume)
- Assign images to nodes
- Choose what a node should do next time it reboots (image itself or boot from its disk)
- Refresh the bittorrent tarball and torrent file for a compute node image after making changes to the expanded image

When compute images are created for the first time, a `bittorrent` tarball is also created. When images are pushed to rack leader controllers (RLCs) for the first time, `bittorrent` is used to transport the tarball snapshot of the image. However, as you make adjustments to your compute image, those changes do not automatically generate a new `bittorrent` tarball. We handle that situation by always doing a follow-up `rsync` of the compute image after transporting the tarball. However, as your compute image begins to diverge from the `bittorrent` tarball snapshot, it becomes less and less efficient to transport a given compute node image that is new to a given RLC.

You no longer need to use `yum`, `yume`, or `mksiimage` commands directly for most common operations. Compute images are automatically configured in such a way as to make them available to the `cimage` command.

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

```
admin:~ # cinstallman --help
Usage: blademond [OPTION] ...

Discover CMCs and blades managed by CMCs.

Note: This daemon normally takes no arguments.
  --help      Print this usage and exit.
  --debug     Enable debug mode (also can be enabled by setting CM_DEBUG)
  --fakecmc   Development only: Discover fake CMCs instead of real ones
  --scan-once Initialize, scan for blades, set blades up. Do not daemonize.
              Do not keep looping - do one pass and exit.

[root@r1lead ~]# exit
logout
Connection to r1lead closed.
[root@river-admin ~]# clear
[root@river-admin ~]# pwd
/root
[root@river-admin ~]# cinstallman --h
cinstallman Usage:

cinstallman is a tool that manages:
 - image creation (as a wrapper to mksiimage)
 - node package updates (as a wrapper to yume)
 - image package updates (yume within a chroot to the image)
```

```
This is a convenience tool and not all operations for the commands that are
wrapped are provided.  The most common operations are collected here for
ease of use.

For operations that take the --node parameter, the node can be an aggregation
of nodes like cimage and cpower can take.  Depending on the situation,
non-managed or offline nodes are skipped.

The tool retrieves the registered repositories from crepo so that they
need not be specified on the command line.

Operations:
--help                  : print his usage message
--create-image          : create a new systemimager image
                          By default, requires --rpmlist and --image
                          Optional flags  below:
       --clone          : Clone existing image, requires --source, --image.
                          Doesn't require --rpmlist.
       --recreate       : Like --del-image then --add-image, but preserves any
                          node associations.
                          Requires --image and --rpmlist
       --repos {list}   : A comma-seperated list of repositories to use.
       --use-existing   : register an already existing image, doesn't
                          require --rpmlist
       --image {image}  : Specify the image to operate on
       --rpmlist {path} : Provide the rpmlist to use when creating images
       --source {image} : Specify a source image to operate on (for clone)

--del-image             : delete the image, may use with --del-nodes
       --image {image}  : Specify the image to operate on

--show-images           : List images, BT 1 if root tarballs are desired

--show-nodes            : Show non-compute nodes (similar to mksimachine -L)

--update-image          : update packages in image to latest packages available
                          in repos, Requires --image
       --image {image}  : Specify the image to operate on

--refresh-image         : Refresh the given image to include all packages
                          in the supplied rpmlist.  Use after registering
```

```
                          new media with crepo that has new suggested rpms.
       --image {image}  : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packets to be sure are included

--yum-image             : Perform yum operations to supplied image, via yume
                          Requires --image, trailing arguments passed to yume
       --image {image}  : Specify the image to operate on

--update-node           : Update supplied node to latest pkgs avail in
                          repos, requires --node
       --node {node}    : Specify the node or nodes to operate on

--refresh-node          : Refresh the given node to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --node {node}    : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packets to be sure are included

--yum-node              : Perform yum operations to nodes, via yume.  Requires
                          --node.  Trailing arguments passed to yume
       --node {node}    : Specify the node or nodes to operate on

--assign-image          : Assign image to node.  Requires --node, --image
       --node {node}    : Specify the node or nodes to operate on
       --image {image}  : Specify the image to operate on

--next-boot             : Action to perform when the service node or leader
                          node next boots.
        {image|bt|disk}:  disk: The node should boot from disk
                          image: re-install the node the standard way
                          bt: re-install the node, make use of bt, requires
                          assgined image to be set up with bittorrent, see
                          --add-to-bt.
                          Requires --node
       --node {node}    : Specify the node or nodes to operate on

--add-to-bt             : Start creating root BT tarballs for this image
                          Note: Compute nodes images are added by default
       --image {image}  : Specify the image to operate on

--del-from-bt           : No longer update BT root tarballs for this image.
```

```
       --image {image}  : Specify the image to operate on

--refresh-bt            : Refresh the bittorrent tarball and torrent file
                          Requires --image
       --image {image}  : Specify the image to operate on

[root@river-admin ~]# clear
[root@river-admin ~]# cinstallman --help
cinstallman Usage:

cinstallman is a tool that manages:
 - image creation (as a wrapper to mksiimage)
 - node package updates (as a wrapper to yume)
 - image package updates (yume within a chroot to the image)

This is a convenience tool and not all operations for the commands that are
wrapped are provided.  The most common operations are collected here for
ease of use.

For operations that take the --node parameter, the node can be an aggregation
of nodes like cimage and cpower can take.  Depending on the situation,
non-managed or offline nodes are skipped.

The tool retrieves the registered repositories from crepo so that they
need not be specified on the command line.

Operations:
--help                  : print his usage message
--create-image          : create a new systemimager image
                          By default, requires --rpmlist and --image
                          Optional flags  below:
       --clone          : Clone existing image, requires --source, --image.
                          Doesn't require --rpmlist.
       --recreate       : Like --del-image then --add-image, but preserves any
                          node associations.
                          Requires --image and --rpmlist
       --repos {list}   : A comma-seperated list of repositories to use.
       --use-existing   : register an already existing image, doesn't
                          require --rpmlist
       --image {image}  : Specify the image to operate on
       --rpmlist {path} : Provide the rpmlist to use when creating images
```

```
       --source {image} : Specify a source image to operate on (for clone)

--del-image             : delete the image, may use with --del-nodes
       --image {image}  : Specify the image to operate on

--show-images           : List images, BT 1 if root tarballs are desired

--show-nodes            : Show non-compute nodes (similar to mksimachine -L)

--update-image          : update packages in image to latest packages available
                          in repos, Requires --image
       --image {image}  : Specify the image to operate on

--refresh-image         : Refresh the given image to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --image {image}  : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packets to be sure are included

--yum-image             : Perform yum operations to supplied image, via yume
                          Requires --image, trailing arguments passed to yume
       --image {image}  : Specify the image to operate on

--update-node           : Update supplied node to latest pkgs avail in
                          repos, requires --node
       --node {node}    : Specify the node or nodes to operate on

--refresh-node          : Refresh the given node to include all packages
                          in the supplied rpmlist.  Use after registering
                          new media with crepo that has new suggested rpms.
       --node {node}    : Specify the node or nodes to operate on
       --rpmlist {path} : rpmlist containing packets to be sure are included

--yum-node              : Perform yum operations to nodes, via yume.  Requires
                          --node.  Trailing arguments passed to yume
       --node {node}    : Specify the node or nodes to operate on

--assign-image          : Assign image to node.  Requires --node, --image
       --node {node}    : Specify the node or nodes to operate on
       --image {image}  : Specify the image to operate on
```

```
--next-boot             : Action to perform when the service node or leader
                          node next boots.
        {image|bt|disk}:  disk: The node should boot from disk
                          image: re-install the node the standard way
                          bt: re-install the node, make use of bt, requires
                          assgined image to be set up with bittorrent, see
                          --add-to-bt.
                          Requires --node
       --node {node}    : Specify the node or nodes to operate on

--add-to-bt             : Start creating root BT tarballs for this image
                          Note: Compute nodes images are added by default
       --image {image}  : Specify the image to operate on

--del-from-bt           : No longer update BT root tarballs for this image.
       --image {image}  : Specify the image to operate on

--refresh-bt            : Refresh the bittorrent tarball and torrent file
                          Requires --image
       --image {image}  : Specify the image to operate on
```

In the following example, the `--refresh-node` operation is used to ensure the online managed service nodes include all the packages in the list. You could use this if you updated your `rpmlist` to include new packages or if you recently added new media with the `crepo` command and want running nodes to have the newly updated packages. A similar `--refresh-image` operation exists for images.

```
# cinstallman --refresh-node --node service\* --rpmlist
/etc/opt/sgi/rpmlists/service-sles11.rpmlist
```

Customizing Software On Your SGI ICE X System

This section discusses how to manage various nodes on your SGI ICE X system. It describes how to configure the various nodes, including the compute and service nodes. It describes how to augment software packages. Many tasks having to do with package management have multiple valid methods to use.

Creating Compute Node Custom Images

You can add per-host compute node customization to the compute node images. You do this by adding scripts either to the `/opt/sgi/share/per-host-customization/global/` directory or the `/opt/sgi/share/per-host-customization/mynewimage/` directory on the system admin controller (SAC).

Note: When creating custom images for compute nodes, make sure you clone the original SGI images. This provides the original images intact that you can fall back to if necessary. The following example is based on SLES.

Scripts in the global directory apply to all compute nodes images. Scripts under the image name apply only to the image in question. The scripts are cycled through once per host when being installed on the rack leader controllers (RLCs). They receive one input argument, which is the full path (on the RLC) to the per-host base directory, for example, `/var/lib/sgi/mynewimage/i2n11`. There is a `README` file at `/opt/sgi/share/per-host-customization/README` on the SAC, as follows:

```
This directory contains compute node image customization scripts which are
executed as part of the install-image operations on the leader nodes when
pulling over a new compute node image.

After the image has been pulled over, and the per-host-customization dir has
been rsynced, the per-host /etc and /var directories are populated, then the
scripts in this directory are cycled through once per-host.  This allows the
scripts to source the node specific network and cluster management settings,
and set node specific settings.

Scripts in the global directory are iterated through first, then if a
directory exists that matches the image name, those scripts are iterated
through next.

You can use the scripts in the global directory as examples.
```

An example global script, `/opt/sgi/share/per-host-customization/global/sgi-fstab` is, as follows:

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
```

```
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation; either version 2 of the License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307 USA
#
# Set up the compute node's /etc/fstab file.
#
# Modify per your sites requirements.
#
# This script is excecuted once per-host as part of the install-image
operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

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

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n '/^compat
/s/^compat.*: //p') )
compatdistro=${os[0]}${os[1]}

if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then

 #
 # SLES 10 compatible
 #
 cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
```

```
tmpfs           /tmp            tmpfs   size=150m       0       0
EOF

elif [ ${compatdistro} = "rhel5"  ]; then

 #
 # RHEL 5 compatible
 #

 #
 # RHEL expects several subsys directories to be present under
/var/run
 # and /var/lock, hence no tmpfs mounts for them
 #
 cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
devpts          /dev/pts        devpts  gid=5,mode=620  0       0
EOF

else

 echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi
```

Modify Compute Image Kernel Boot Options

You can use the `cattr` command to set extra kernel boot parameters for compute nodes on a per-image basis. For example to append `cgroup_disable=memory` to kernel boot parameters for any node booting the `compute-sles11sp1` image, perform a command similar to the following:

```
% cattr set kernel_extra_params-compute-sles11sp1 cgroup_disable=memory
```

Push the image, as follows:

```
# cimage --push-rack mynewimage r1
```

Compute Node Per-Host Customization for Additional Network Interfaces

Note: The following example is only for systems running SLES.

Per compute-node customization may be useful for configuring additional network interfaces that are on some, but not all, compute nodes. An example of how to configure network interfaces on individual compute nodes is the `/opt/sgi/share/per-host-customization/mynewimage/mycustomization` script, that follows:

```
#Copyright (c) 2008 Silicon Graphics, Inc.
# All rights reserved.
#
# do node specific setup
#
# This script is excecuted once per-host as part of the install-image operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $ARGV[0],
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

use lib "/usr/lib/systemconfig","/opt/sgi/share/per-host-customization/global";
use sanity;

sanity_checks();

$blade_path = $node = $ARGV[0];
$node =~ s/.*\///;

sub i0n4 {
        my $ifcfg="etc/sysconfig/network/ifcfg-eth2";
        open(IFCFG, ">$blade_path/$ifcfg") or
                die "$0: can't open $blade_path/$ifcfg";
        print IFCFG<<EOF
BOOTPROTO='static'
IPADDR='10.20.0.1'
NETMASK='255.255.0.0'
STARTMODE='onboot'
WIRELESS='no'
EOF
        ;
```

```
        close(IFCFG);
}

@nodes = ("i0n4");

foreach $n (@nodes) {
        if ( $n eq $node ) {
                eval $n;
        }
}
```

Pushing `mynewimage` to rack 1 causes the eth2 interface of compute node `r1i0n4` to be configured with IP address 10.20.0.1 when the node is brought up with `mynewimage`. Push the image, as follows:

```
# cimage --push-rack mynewimage r1
```

Customizing Software Images

Note: Procedures in this section describe how to work with service node and compute node images. Always use a cloned image. If you are adjusting an RPM list, use your own copy of the RPM list.

The service and compute node images are created during the `configure-cluster` operation (or during your upgrade from a prior release). This process uses an RPM list to generate a root on the fly.

You can clone a compute node image, or create a new one based on an RPM list. For service nodes, SGI does not support a clone operation. For compute images, you can either clone the image and work on a copy or you can always make a new compute node image from the SGI supplied default RPM list.

Procedure 1-11 Creating a Simple Compute Node Image Clone

Note: Always work from a clone image, see "Customizing Software Images" on page 37.

To create a simple compute node image clone from the system admin controller (SAC), perform the following steps:

1. To clone the compute node image, perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image compute-sles11-new
```

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

```
# cimage --list-images
image: compute-sles11
       kernel: 2.6.27.19-5-smp

image: compute-sles11-new
       kernel: 2.6.27.19-5-smp
```

3. To push the compute node image out to the rack, perform the following:

```
# cimage --push-rack compute-sles11-new r\*
```

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

```
# cimage --set compute-sles11-new 2.6.27.19-5-smp "r*i*n*"
```

Procedure 1-12 Manually Adding a Package to a Compute Node Image

To manually add a package to a compute node image, perform the steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "`cinstallman` Command" on page 26.

---

1. Make a clone of the compute node image, as described in "Customizing Software Images" on page 37.
2.

Note: This example shows SLES11.

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

```
# cimage --list-images
  image: compute-sles11
         kernel: 2.6.27.19-5-smp

  image: compute-sles11-new
         kernel: 2.6.27.19-5-smp
```

3. From the system admin controller (SAC), change directory to the images directory, as follows:

   ```
   # cd /var/lib/systemimager/images/
   ```

4. From the SAC, copy the RPMs you wish to add, as follows, where `compute-sles11-new` is your own compute node image, as follows:

   ```
   # cp /tmp/newrpm.rpm compute-sles11-new/tmp
   ```

5. The new RPMs now reside in `/tmp` directory in the image named `compute-sles11-new`. To install them into your new compute node image, perform the following commands:

   ```
   # chroot compute-sles11-new bash
   ```

   And then perform the following:

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

   At this point, the image has been updated with the RPM.

6. The image on the SAC is updated. However, you still need to push the changes out. Ensure there are no nodes currently using the image and then run this command:

   ```
   # cimage --push-rack compute-sles11-new r\*
   ```

   This will push the updates to the rack lead controllers and the changes will be seen by the compute nodes the next time they start up. For information on how to ensure the image is associated with a given node, see the `cimage --set` command and the example in Procedure 1-11, page 37.

Procedure 1-13 Manually Adding a Package to the Service Node Image

To manually add a package to the service node image, perform the following steps:

Note: Use the `cinstallman` command to install packages into images when the package you are adding is in a repository. This example shows a quick way to manually add a package for compute nodes when you do **not** want the package to be in a custom repository. For information on the `cinstallman` command, see "`cinstallman` Command" on page 26.

1. Use the `cinstallman` command to create your own version of the service node image. See "`cinstallman` Command" on page 26.
2. Change directory to the `images` directory, as follows:

   # **`cd /var/lib/systemimager/images/`**

3. From the SAC, copy the RPMs you wish to add, as follows, where `my-service-image` is your own service node image:

   # **`cp /tmp/newrpm.rpm my-service-image/tmp`**

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

   # **`chroot my-service-image bash`**

   And then perform the following:

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

   At this point, the image has been updated with the RPM. Please note, that unlike compute node images, changes made to a service node image will not be seen by service nodes until they are reinstalled with the image. If you wish to install the package on running systems, you can copy the RPM to the running system and use the RPM from there.

`cimage` Command

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

For a help statement, perform the following command:

```
admin:~ # cimage --help
cimage is a program for managing compute node root images in SMC for ICE.

Usage: cimage OPTION ...

Options
 --help                                 Usage and help text.
 --debug                                Output additional debug information.
 --list-images                          List images and their kernels.
 --list-nodes NODE                      List node(s) and what they are set to.
 --set [OPTION] IMAGE KERNEL NODE       Set node(s) to image and kernel.
    --nfs                               Use NFS roots (default).
    --tmpfs                             Use tmpfs roots.
 --set-default [OPTION] IMAGE KERNEL    Set default image, kernel, rootfs type.
    --nfs                               Use NFS roots (default).
    --tmpfs                             Use tmpfs roots.
 --show-default                         Show default image, kernel, rootfs type.
 --add-db IMAGE                         Add image and its kernels to the db.
 --del-db IMAGE                         Delete image and its kernels from db.
 --update-db IMAGE                      Short-cut for --del-db, then --add-db.
 --push-rack [OPTIONS] IMAGE RACK       Push or update image on rack(s).
    --force                             Bypass the booted nodes check, deletes.
    --update-only                       Skip files newer in dest, no delete.
    --quiet                             Turn off diagnostic information.
 --del-rack IMAGE RACK                  Delete an image from rack(s).
 --clone-image OIMAGE NIMAGE            Clone an existing image to a new image.
 --del-image [OPTIONS] IMAGE            Delete an existing image entirely.
    --quiet                             Turn off diagnostic information.

RACK arguments take the format 'rX'
NODE arguments take the format 'rXiYnZ'
ROOTFS argument can be either 'nfs' or 'tmpfs'

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

EXAMPLES

Example 1-2 `cimage` Command Examples

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

To list the available images and their associated kernels, perform the following:

```
# cimage --list-images
image: compute-sles11
        kernel: 2.6.27.19-5-carlsbad
        kernel: 2.6.27.19-5-default
image: compute-sles11-1_7
        kernel: 2.6.27.19-5-default
```

To list the compute nodes in rack 1 and the image and kernel they are set to boot, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-default nfs
r1i0n8: compute-sles11 2.6.27.19-5-default nfs
```

The `cimage` command also shows the root filesystem type (NFS or tmpfs).

To set the `r1i0n0` compute node to boot the `2.6.27.19-5-smp` kernel from the `compute-sles11` image, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i0n0
```

To list the nodes in rack 1 to see the changes set in the example above, perform the following:

```
# cimage --list-nodes r1
r1i0n0: compute-sles11 2.6.27.19-5-smp
r1i0n1: compute-sles11 2.6.27.19-5-smp
r1i0n2: compute-sles11 2.6.27.19-5-smp
[...snip...]
```

To set all nodes in all racks to boot the `2.6.27.19-5-smp` kernel from the `compute-sles11` image, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r*i*n*
```

To set two ranges of nodes to boot the 2.6.27.19-5-smp kernel, perform the following:

```
# cimage --set compute-sles11 2.6.27.19-5-smp r1i[0-2]n[5-6] r1i[2-3]n[0-4]
```

To clone the compute-sles11 image to a new image (so that you can modify it) , perform the following:

```
# cinstallman --create-image --clone --source compute-sles11 --image mynewimage
Cloning compute-sles11 to mynewimage ... done
```

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

Note: If you have made changes to the compute node image and are pushing that image out to rack leader controllers (RLCs), it is a good practice to use the `cinstallman --refresh-bt --image {`*image*`}` command to refresh the `bittorrent` tarball and `torrent` file for a compute node image. This avoids duplication by `rsync` when the image is pushed out to the RLCs. For more information, see the `cinstallman --h` usage statement or "`cinstallman` Command" on page 26.

To change to the cloned image created in the example, above, copy the needed RPMs into the `/var/lib/systemimager/images/mynewimage/tmp` directory, use the `chroot` command to enter the directory and then install the RPMs, perform the following:

```
# cp *.rpm /var/lib/systemimager/images/mynewimage/tmp
# chroot /var/lib/systemimager/images/mynewimage/ bash
# rpm -Uvh /tmp/*.rpm
```

If you make changes to the kernels in the image, you need to refresh the kernel database entries for your image, To do this, perform the following:

```
# cimage --update-db mynewimage
```

If you did not make changes to the kernels in the cloned image created in the example above, you can omit this step.

To push new software images out to the compute blades in a rack or set of racks, perform the following:

```
# cimage --push-rack mynewimage r*
r1lead: install-image: mynewimage
r1lead: install-image: mynewimage done.
```

To list images in the database the kernels they contain, perform the following:

```
# cimage --list-images

image: compute-sles11
        kernel: 2.6.16.60-0.7-carlsbad
        kernel: 2.6.16.60-0.7-smp

image: mynewimage
        kernel: 2.6.16.60-0.7-carlsbad
        kernel: 2.6.16.60-0.7-smp
```

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

```
# cimage --set mynewimage 2.6.16.60-0.7-smp r1i3n*
```

You need to reboot the compute nodes to run the new images.

Completely remove an image you no longer use, both from system admin controller (SAC) and all compute nodes in all racks, perform the following:

```
# cimage --del-image mynewimage
r1lead: delete-image: mynewimage
r1lead: delete-image: mynewimage done.
```

Using `cinstallman` to Install Packages into Software Images

The packages that make up SMC for SGI ICE X, SGI Foundation, and the Linux distribution media, and any other media or custom repositories you have added reside in repositories. The `cinstallman` command looks up the list of all repositories and provides that list to the commands it calls out for its operation such as `yume`.

Note: Always work with copies of software images.

The `cinstallman` command can update packages within `systemimager` images. You may also use `cinstallman` to install a single package within an image.

However, `cinstallman` and the commands it calls only works with the configured repositories. So if you are installing your own RPM, you will need that package to be part of an existing repository. You may use the `crepo` command to create a custom repository into which you can collect custom packages.

Note: The `yum` command maintains a cache of the package metadata. If you just recently changed the repositories, `yum` caches for the nodes or images you are working with may be out of date. In that case, you can issue the `yum` command "`clean all`" with `--yum-node` and `--yum-image`. The `cinstallman` command `--update-node` and `--update-image` options do this for you.

The following example shows how to install the `zlib-devel` package in to the service node image so that the next time you image or install a service node, it will have this new package.

```
# cinstallman --yum-image --image my-service-sles11 install zlib-devel
```

You can perform a similar operation for compute node images. Note the following:

- If you update a compute node image on the system admin controller (SAC), you have to use the `cimage` command to push the changes. For more information on the `cimage` command, see "`cimage` Command" on page 40.
- If you update a service node image on the SAC, that service node needs to be reinstalled and/or reimaged to get the change. The `discover` command can be given an alternate image or you may use the `cinstallman --assign-image` command followed by the `cinstallman --next-boot` command to direct the service node to reimage itself with a specified image the next time it boots.

Using yum to Install Packages on Running Service or Rack Leader Controllers (RLCs)

Note: These instructions only apply to managed service nodes and RLCs. They do not apply to compute nodes.

You can use the `yum` command to install a package on a service node. From the system admin controller (SAC), you can issue a command similar to the following:

```
# cinstallman --yum-node --node service0 install zlib-devel
```

Note: To get all service nodes, replace `service0` with `service\*`.

For more information on the `cinstallman` command, see "`cinstallman` Command" on page 26.

Creating Compute and Service Node Images Using the `cinstallman` Command

You can create service node and compute node images using the `cinstallman` command. This generates a root directory for images, automatically.

Fresh installations of SMC for SGI ICE X create these images during the `configure-cluster` installation step.

The RPM lists that drive which packages get installed in the images are listed in files located in `/etc/opt/sgi/rpmlists`. For example, `/etc/opt/sgi/rpmlists/compute-sles11.rpmlist` (see "`crepo` Command" on page 22). 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 1-14 Using the `cinstallman` Command to Create a Service Node Image:

To create a service node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the example service node image RPM list and work on the copy, as follows:

```
# cp /etc/opt/sgi/rpmlists/service-sles11.rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.

3. Use the `cinstallman` command with the `--create-image` option to create the images root directory, as follows:

```
# cinstallman --create-image --image my-service-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-service-node.rpmlist
```

This example uses `my-service-node-image` as the home/name of the image.

Output is logged to `/var/log/cinstallman` on the system admin controller (SAC).

4. After the `cinstallman` comand finishes, the image is ready to be used with service nodes. You can supply this image as an optional image name to the `discover` command, or you may assign an existing service node to this image using the `cinstallman --assign-image` command. You can tell a service node to image itself next reboot by using `the cinstallman --next-boot` option.

Procedure 1-15 Use the `cinstallman` Command to Create a Compute Node Image

To create a compute node image using the `cinstallman` command, perform the following steps:

1. Make a copy of the compute node image RPM list and work on the copy, as follows:

   # **cp /etc/opt/sgi/rpmlists/compute-sles11.rpmlist /etc/opt/sgi/rpmlists/my-compute-node.rpmlist**

2. Add or remove any packages from the RPM list. Keep in mind that needed dependencies are pulled in automatically.

3. Run the `cinstallman` command to create the root, as follows:

```
# cinstallman --create-image --image my-compute-node-image --rpmlist
/etc/opt/sgi/rpmlists/my-compute-node.rpmlist
```

   This example uses the name `my-compute-node-image` as the name.

   Output is logged to `/var/log/cinstallman` on the SAC.

   The `cinstallman` command makes the new image available to the `cimage` command.

4. For information on how to use the `cimage` command to push this new image to rack leader controllers (RLCs), see "`cimage` Command" on page 40.

Installing a Service Node with a Non-default Image

If you have a non-default service node image you wish to install on a service node, you have two choices, as follows:

- Specify the image name when you first discover the node with the `discover` command.
- Use the `cinstallman` command to associate an image with a service node, then set up the node to reinstall itself the next time it boots.

The following example shows how to associate a custom image at discover time:

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

The next example shows how to reinstall an already discovered service node with a new image:

```
# cinstallman --assign-image --node service2 --image my-service-node-image
# cinstallman --next-boot image --node service2
```

When you reboot the node, it will reinstall itself.

For more information on the `discover` command, see "`discover` Command" on page 10. For more information on the `cinstallman` command, see "`cinstallman` Command" on page 26.

Retrieving a Service Node Image from a Running Service Node

To retrieve a service node image from a running service node, perform the following steps:

1. As **root user**, log into the service node from which you wish to retrieve an image. You can use the `si_prepareclient`(8) program to extract an image. Type the following command to start the program:

```
service0:~ # si_prepareclient --server admin --no-uyok
Welcome to the SystemImager si_prepareclient command. This command may modify
the following files to prepare your golden client for having its image
retrieved by the imageserver.  It will also create the /etc/systemimager
directory and fill it with information about your golden client. All modified
files will be backed up with the .before_systemimager-3.8.0 extension.

 /etc/services:
   This file defines the port numbers used by certain software on your system.
   Entries for rsync will be added if necessary.

 /tmp/filetlOeP5:
   This is a temporary configuration file that rsync needs on your golden client
   in order to make your filesystem available to your SystemImager server.

 inetd configuration:
   SystemImager needs to run rsync as a standalone daemon on your golden client
   until its image is retrieved by your SystemImager server.  If rsyncd is
```

```
   configured to run as a service started by inetd, it will be temporarily
   disabled, and any running rsync daemons or commands will be stopped.  Then,
   an rsync daemon will be started using the temporary configuration file
   mentioned above.

See "si_prepareclient --help" for command line options.

Continue? (y/[n]):
```

Enter **y** to continue. After a few moments, you are returned to the command prompt. You are now ready to retrieve the image from the system admin controller (SAC).

2. Exit the **service0** node, and as **root user** on the SAC, perform the following command: (Replace the image name and service node name, as needed.)

   ```
   admin # mksiimage --Get --client service0 --name myimage
   ```

   It now retrieves the image. No progress information is provided. It takes several minutes depending on the size of the image on the service node.

3. Use the `cinstallman` command to register the newly collected image:

   ```
   admin # cinstallman --create --use-existing --image myimage
   ```

4. If you want to discover a node using this image directly, you can use the `discover` command, as follows:

   ```
   admin # discover --service 0,image=myimage
   ```

5. If you want to re-image an already discovered node with your new image, run the following commands:

   ```
   # cinstallman --assign-image --node service0 --image myimag
   # cinstallman --next-boot image --node service0
   ```

6. Reboot the service node.

Using a Custom Repository for Site Packages

This section describes how to maintain packages specific to your site and have them available to the `crepo` command (see "`crepo` Command" on page 22).

SGI suggests putting site-specific packages in a separate location. They should not reside in the same location as SGI or Novell supplied packages.

Procedure 1-16 Setting Up a Custom Repository for Site Packages

To set up a custom repository for your custom packages, perform the following steps:

1. Create directory for your site-specific packages on the system admin controller (SAC), as follows:

   # **mkdir -p /tftpboot/site-local/sles-10-x86_64**

2. Copy your site packages in to the new directory, as follows:

   # **cp my-package-1.5.x86_64.rpm /tftpboot/site-local/sles-10-x86_64**

3. Register your custom repository using the `crepo` command. This command will ensure your repository is consulted when the `cinstallman` command performs its operations. This command also creates the necessary `yum`/`repomd` metadata.

   # **crepo --add /tftpboot/site-local/sles-10-x86_64 --custom my-repo**

   Your new repository may be consulted by `cinstallman` command operations going forward including updating images, nodes, and creating images.

4. If you wish this repository to be used by `cinstallman` by default, you need to select it. Use the following command:

   # **crepo --select my-repo**

5. If you use `cinstallman` to create an image, you will want to add your custom package to the rpmlist you use with the cinstallman command (see "Using `cinstallman` to Install Packages into Software Images" on page 44).

SGI ICE X System Configuration Framework

All node types that are part of an SGI ICE X system can have configuration settings adjusted by the configuration framework. There is some overlap between the per-host customization instructions and the configuration framework instructions. Each approach plays a role in configuring your system. The major differences between the two methods are, as follows:

- Per-host customization runs at the time an image is pushed to the rack leader controllers (RLCs).

- Per-host customization only applies to compute node images.
- The SGI ICE system configuration framework can be used with all node types.
- The system configuration framework is run when a new root is created, when `SuSEconfig` command is run for some other reason, as part of a `yum` operation, or when new compute images are pushed with the `cimage` command.

This framework exists to make it easy to adjust configuration items. There are SGI-supplied scripts already present. You can add more scripts as you wish. You can also exclude scripts from running without purging the script if you decide a certain script should not be run. The following set of questions in bold and bulleted answers describes how to use the system configuration framework.

How does the system configuration framework operate?

These files could be added, for example, to a running service node, or to an already created service or compute image. Remember that images destined for compute nodes need to be pushed with the `cimage` command after being altered. For more information, see "`cimage` Command" on page 40.

- 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 only run if the node in question is running that `distro`. For example, `/etc/opt/sgi/conf.d/99-foo.sles` would only run if the node was running `sles`. This example shows the precedence of operations.

  If you had `88-myscript.sles10`, `88-myscript.sles`, and `88-myscript`:

  – On a `sles10` system, `88-myscript.sles10` would execute

  – On a `sles` system that is not `sles10`, `88-myscript.sles` would execute

  – On all other `distro`s, `88-myscript` would execute

- If you wish to make a custom version of an script supplied by SGI, you may simply name it with `.local` and the local version will run in place of the one supplied by SGI. This allows for customization without modifying scripts supplied by SGI. Scripts ending in `.local` have the highest precedence. In other words, if you had `88-myscript.sles`, and `88-myscript.local`, then `88-myscript.local` would execute in all cases and the other `88-myscript` scripts would never execute.

From where is the framework called?

- The callout for `/opt/sgi/lib/cluster-configuration` is implemented as a `yum` plugin that executes after packages have been installed and cleaned.
- On SLES only, there is also a SUSE configuration script in the `/sbin/conf.d` directory, called `SuSEconfig.00cluster-configuration`, that calls the framework. This is in case of you are using YaST to install or upgrade packages.
- On SLES only, one of the scripts called by the framework calls `SuSEconfig`. A check is made to avoid a callout loop.
- The framework is also called when the system admin conroller (SAC), RLC, or service nodes start up. The call is made just after networking is configured. As a site administrator, you could create custom scripts here that check on or perform certain configuration operations.
- When using the `cimage` command to push a compute node root image to RLCs, the configuration framework executes within the `chroot` of the compute node image after it is pulled from the SAC to the RLC.

How do I adjust my system configuration?

- Create a small script in `/etc/opt/sgi/conf.d` to do the adjustment.

  Be sure that you test for existence of files and do not assume they are there (see "Why do scripts need to tolerate files that do not exist but should?" below).

Why do scripts need to tolerate files that do not exist but should?

- This is because the `mksiimage` command runs `yume` and `yum` in two steps. The first step only installs 40 or so RPMs but our framework is called then too. The second pass installs the other "hundreds" of RPMs. So the framework is called once before many packages are installed, and again after everything is in place. So not all files you expect might be available when your small script is called.

How does the yum plugin work?

- In order for the yum plugin to work, the /etc/yum.conf file has to have plugins=1 set in its configuration file. SMC for SGI ICE X software ensures that is in place by way of a trigger in the sgi-cluster package. Anytime yum is installed or updated, it verify plugins=1 is set.

How does yume work?

- yume, an oscar wrapper for yum, works by creating a temporary yum configuration file in /tmp and then points yum at it. This temporary configuration file needs to have plugins enabled. A tiny patch to yume makes this happen. This fixes it for yume and also mksiimage, which calls yume as part of its operation.

Cluster Configuration Repository: Updates on Demand

SMC for ICE X contains a cluster configuration repository / update framework. This framework generates and distributes configuration updates to system admin controller (SAC), rack leader controller (RLC), and service nodes in the cluster. Some of the configuration files managed by this framework include C3 conserver, DNS, Ganglia, hosts files, and NTP.

When an event occurs that requires these files to be updated, the framework executes on the SAC. The SAC stores the updated configuration framework in a special cached location and updates the appropriate nodes with their new configuration files.

In addition to the updates happening as required, the configuration file repository is consulted when a SAC, RLC, or service node boots. This happens shortly after networking is started. Any configuration files that are new or updated are transferred at this early stage so that the node is fully configured by the time the node is fully operational.

There are no hooks for customer configuration in the configuration repository at this time.

This update framework is tied in with the /etc/opt/sgi/conf.d configuration framework to provide a full configuration solution. As mentioned earlier, customers are encouraged to create /etc/opt/sgi/conf.d scripts to do cluster configuration.

cnodes Command

The cnodes command provides information about the types of nodes in your system. For help information, perform the following:

```
[admin ~]# cnodes --help

Usage: cnodes [OPTIONS]

Options:
 --all                 all compute, leader and service nodes, and switches
 --compute             all compute nodes
 --leader              all leader nodes
 --service             all service nodes
 --ibswitch            all ib switch nodes
 --mgmtswitch          all cluster management switches
 --switch-blade        all switch blade nodes
 --cmc                 all CMCs
 --online              modifier: nodes marked online
 --offline             modifier: nodes marked offline
 --managed             modifier: managed nodes
 --unmanaged           modifier: unmanaged nodes
 --temponames          modifier: return Tempo node names instead of hostnames
 --rack=RACK           modifier: only match nodes related to RACK

Note: default modifiers are 'online' and 'managed' unless otherwise specified.
```

EXAMPLES

Example 1-3 cnodes Example

The following examples walk you through some typical cnodes command operations.

To see a list of all nodes in your system, perform the following:

```
[admin ~]# cnodes --all
r1i0n0
r1i0n1
r1lead
service0
```

To see a list of all compute nodes, perform the following:

```
[admin ~]# cnodes --compute
r1i0n0
```

```
r1i0n1
```

To see a list of service nodes, perform the following:

```
[admin ~]#  cnodes --service
service0
```

Power Management Commands

The `cpower` command allows you to power up, power down, reset, and show the power status of system components.

`cpower` Command

The `cpower` command is, as follows:

`cpower [<`*option*`> ...] [<`*target_type*`>] [<`*action*`>] <`*target*`>`

The <*option*> argument can be one or more of the following:

Option	Description
`--noleader`	Do not include rack leader controllers (RLCs) (valid with rack and system domains only).
`--noservice`	Do not include service nodes (valid with system domain only).
`--force`	When using wildcards in the target, disable all “safety” checks. Make sure you really want to use this command.
`-n, --noexec`	Displays, but does not execute, commands that affect power.
`-v, --verbose`	Print additional information on command progress

The <*target*> argument is one of the following:

`--node`	Applies the action to nodes. Nodes are compute nodes, RLCs, system admin controllers (SACs), and service nodes. [default]
`--iru`	Applies the action at the IRU level (now referred to as a blade enclosure pair).

`--rack`	Applies the action at the rack level.
`--system`	Applies the action to the system. You must **not** specify a target with this type.

The <*action*> argument is one of the following:

`--status`	Show the power status of the target, including whether it is booted or not. [default]
`--up | --on`	Powers up the target.
`--down | --off`	Powers down the target.
`--reset`	Performs a hard reset on the target.
`--cycle`	Power cycles the target.
`--boot`	Boots up the target, unless it is already booted. Waits for all targets to boot.
`--reboot`	Reboots the target, even if already booted. Wait for all targets to boot.
`--halt`	Halts and then powers off the target.
`--shutdown`	Shuts down the target, but does not power it off. Waits for targets to shut down.
`--identify` <*interval*>	Turns on the identifying LED for the specified interval in seconds. Uses an interval of 0 to turn off immediately.
`-h, --help`	Shows help usage statement.

The target must always be specified except when the `--system` option is used. Wildcards may be used, but be careful **not** to accidentally power off or reboot the RLCs. If wildcard use affects any RLC, the command fails with an error.

Operations on Nodes

The default for the `cpower` command is to operate on system nodes, such as compute nodes, rack leader controllers (RLCs), or service nodes. If you do **not** specify the `--iru`, `--rack`, or `--system` option, the command defaults to operating as if you had specified `--node`. Individual rack units are now called blade enclosure pairs but the command syntax is the same.

Here are examples of node target names:

- `r1i3n10`

  Compute node at rack 1, IRU 3, slot 10

- `service0`

  Service node 0

- `r3lead`

  RLC for rack 3

- `r1i*n*`

  Wildcards let you specify ranges of nodes, for example, `r1i*n*` all compute nodes in all IRUs on rack 1

IPMI-style Commands

The default operation for the `cpower` command is to operate on nodes and to provide you the status of these nodes, as follows:

```
# cpower r1i*n*
```

This command is equivalent to the following:

```
# cpower --node --status r1i*n*
```

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

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

The default is to apply to a node.

The following commands behave exactly as you would expect as if you were using `ipmitool`, and have no special extra logic for ordering:

```
# cpower --up r1i*n*
```

```
# cpower --reset r1i*n*
```

```
# cpower --cycle r1i*n*
```

```
# cpower --identify 5 r1i*n*
```

Note: `--up` is a synonym for `--on` and `--down` is a synonym for `--off`.

IRU, Rack, and System Domains

The `cpower` command contains more logic when you go up to higher levels of abstraction, for example, when using the `--iru`, `--rack`, and `--system` options. These higher level domain specifiers tell the command to be smart about how to order various of the actions that you give on the command line.

Note: Individual rack units (IRUs) are now called blade enclosure pairs. The command syntax works the same, as in previous releases.

The `--iru` option tells the command to use correct ordering with IRU power commands. In this case, it firsts connect to the CMC on each IRU in rack 1 to issue the `power on` command, which turns on power to the IRU chassis (this is not the equivalent `ipmitool` command). Then it powers up the compute nodes in the IRU. Powering things down is the opposite, with the power to the IRU being turned off after power to the blades. IRU targets are specified as follows:`r3i2` for rack 3, IRU 2.

```
# cpower --iru --up r1i*
```

The `--rack` option ensures power commands to the rack leader controller (RLC) are down in the correct order relative to compute nodes within a rack. First, it powers up the RLC and waits for it to boot up (if it is not already up). Then it will do the functional equivalent of a `cpower --iru --up r4i*` on each of the IRUs contained in the rack, including applying power to each IRU chassis. Using the `--down` option is the opposite, and also turns off the RLC (after doing a shutdown) after all the IRUs are powered down. To avoid including RLCs in a power command for a rack, use the `--noleader` option. Rack targets are specified, as follows: `r4` for rack 4. Here is an example:

```
# cpower --rack --up r4
```

Commands with the **`--system`** option ensures that power up commands are applied first to service nodes, then to RLCs, then to IRUs and compute blades, in just the same way. Likewise, compute blades are powered down before IRUs, RLCs, and service nodes, in that order. To avoid including service nodes in a system-domain command, use the **`--noservice`** option. Note that you must not specify a target with `--system` option, since it applies to the SGI ICE system.

Shutting Down and Booting

Note: The `--shutdown --off` combination of actions were deprecated in a previous release. Use the `--halt` option in its place.

It is useful to be able to shutdown a machine before turning off the power, in most cases. The following `cpower` options to enable you to do this: `--halt`, `--boot`, and `--reboot`. The `--halt` option allows you to shut down a node. The `--reboot` option ensures that a system is always rebooted, whereas `--boot` will only boot up a system if it is not already booted. Thus, `--boot` is useful for booting up compute blades that have failed to start.

You need to configure the order in which service nodes are booted up and shut down as part of the overall system power management process. This is done by setting a `boot_order` for each service node. Use the `cadmin` command to set the boot order for a service node, for example:

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

The `cpower --system --boot` command boots up service nodes with a lower boot order, first. It then boots up service nodes with a higher boot order. The reverse is true when shutting down the system with `cpower`. For example, if service1 has a boot order of 3 and service2 has a boot order of 5, service1 is booted completely, and then service2 is booted, afterwards. During shutdown, service2 is shut down completely before service1 is shutdown.

There is a special meaning to a service node having a boot order of zero. This value causes the `cpower --system` command to skip that service node completely for both start up and shutdown (although not for status queries). Negative values for the service node boot order setting are not permitted.

Note: The IPMI power commands necessary to enable a system to boot (either with a power reset, or a power on) may be sent to a node. The `--halt` option, halts the target node and then powers it off.

The `--halt` options works on node, IRU, or rack domain levels. It will shut down nodes (in the correct order if you use the `--iru` or `--rack` options), and then just leave them as they are, power still applied. Using both these actions results in nodes being halted, then powered off. This is particularly useful when powering off a rack,

since otherwise, the rack leader controllers (RLCs) may be shut down before there is a chance to power off the compute blades. Here is an example:

```
# cpower --halt --rack r1
```

To boot up systems that have not already been booted, perform the following:

```
# cpower --boot  r1i2n*
```

Again, the command boots up nodes in the right orders if you specify the `--iru` or `--rack` options and the appropriate target. Otherwise, there is no guarantee that, for example, the command will attempt to power on the RLC before compute nodes in the same rack.

To reboot all of the nodes specified, or boot them if they are already shut down, perform the following:

```
# cpower --reboot --iru r3i3
```

The `--iru` or `--rack` options ensure proper ordering if you use them. In this case, the command will make sure that power is supplied to the chassis for rack 3, IRU 3, and then the all the compute nodes in that IRU will be rebooted.

EXAMPLES

Example 1-4 `cpower` Command Examples

To boot compute blade `r1i0n8`, perform the following:

```
# cpower --boot r1i0n8
```

To boot a number of compute blades at the same time, perform the following:

```
# cpower --boot --rack r1
```

Note: The `--boot` option will only boot those nodes that have not already booted.

To shut down service node 0, perform the following:

```
# cpower --halt service0
```

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

```
# cpower --halt --rack r3
```

Note: This command will shutdown and then power off all of the computer nodes in parallel, then shutdown and power off the RLC. Use the `--noleader` option if you want the RLC to remain booted up.

To shutdown the entire system, including all service nodes and all RLCs, but not the system admin controller (SAC), and not turn the power off to anything, perform the following:

```
# cpower --halt --system
```

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

```
# cpower --halt --system --noleader --noservice
```

Note: The only way to shut down the SAC is to perform the operation manually.

Cluster Command and Control (C3) Commands

This section describes the cluster command and control (C3) tool suite for cluster administration and application support. For more information about how to run commands on multiple nodes, see the `pdsh` and `pdcp` utilities described in "`pdsh` and `pdcp` Utilities" on page 66.

The C3 commands are as follows:

C3 Utilities	Description
`cexec`(s)	Executes a given command string on each node of a cluster
`cget`	Retrieves a specified file from each node of a cluster and places it into the specified target directory
`ckill`	Runs `kill` on each node of a cluster for a specified process name
`clist`	Lists the names and types of clusters in the cluster configuration file

`cnum`	Returns the node names specified by the range specified on the command line
`cname`	Returns the node positions specified by the node name given on the command line
`cpush`	Pushes files from the local machine to the nodes in your cluster

`cexec` is the most useful C3 utility. Use the `cpower` command rather than `cshutdown` (see "Power Management Commands" on page 55).

EXAMPLES

Example 1-5 C3 Command General Examples

The following examples walk you through some typical C3 command operations.

You can use the `cname` and `cnum` commands to map names to locations and vice versa, as follows:

```
# cname rack_1:0-2
local name for cluster:  rack_1
nodes from cluster:  rack_1
cluster:  rack_1 ; node name:  r1i0n0
cluster:  rack_1 ; node name:  r1i0n1
cluster:  rack_1 ; node name:  r1i0n10

# cnum rack_1: r1i0n0
local name for cluster:  rack_1
nodes from cluster:  rack_1
r1i0n0 is at index 0 in cluster rack_1

# cnum rack_1: r1i0n1
local name for cluster:  rack_1
nodes from cluster:  rack_1
```

You can use the `clist` command to retrieve the number of racks, as follows:

```
# clist
cluster  rack_1  is an indirect remote cluster
cluster  rack_2  is an indirect remote cluster
cluster  rack_3  is an indirect remote cluster
cluster  rack_4  is an indirect remote cluster
```

You can use the cexec command to view the addressing scheme of the C3 utility, as follows:

```
# cexec rack_1:1 hostname
************************* rack_1 *************************
************************* rack_1 *************************
--------- r1i0n1---------
r1i0n1

# cexec rack_1:2-3 rack_4:0-3,10 hostname
************************* rack_1 *************************
************************* rack_1 *************************
--------- r1i0n10---------
r1i0n10
--------- r1i0n11---------
r1i0n11
************************* rack_4 *************************
************************* rack_4 *************************
--------- r4i0n0---------
r4i0n0
--------- r4i0n1---------
r4i0n1
--------- r4i0n10---------
r4i0n10
--------- r4i0n11---------
r4i0n11
--------- r4i0n4---------
r4i0n4
```

The following set of command shows how to use the C3 commands to transverse the different levels of hierarchy in your SGI ICE X system.

To execute a C3 command on all blades within the default SGI ICE X system, for example, rack 1, perform the following:

```
# cexec hostname
************************* rack_1 *************************
************************* rack_1 *************************
--------- r1i0n0---------
r1i0n0
--------- r1i0n1---------
```

```
r1i0n1
--------- r1i0n10---------
r1i0n10
--------- r1i0n11---------
r1i0n11

...
```

To run a C3 command on all compute nodes across an SGI ICE X system, perform the following:

```
# cexec --all hostname
************************ rack_1 ************************
************************ rack_1 ************************
--------- r1i0n0---------
r1i0n0
--------- r1i0n1---------
r1i0n1
...
--------- r2i0n10---------
r2i0n10
...
--------- r3i0n11---------
r3i0n11
...
```

To run a C3 command against the first rack leader controller (RLC), in the first rack, perform the following:

```
# cexec --head hostname
************************ rack_1 ************************
--------- rack_1---------
r1lead
```

To run a C3 command against all RLCs, across all racks, perform the following:

```
# cexec --head --all hostname
************************ rack_1 ************************
--------- rack_1---------
r1lead
************************ rack_2 ************************
```

```
--------- rack_2---------
r2lead
************************ rack_3 ************************
--------- rack_3---------
r3lead
************************ rack_4 ************************
--------- rack_4---------
r4lead
```

The following set of examples shows some specific case uses for the C3 commands that you are likely to employ.

Example 1-6 C3 Command Specific Use Examples

From the system admin controller (SAC), run command on rack 1 without including the RLC, as follows:

cexec rack_1: <*cmd*>

Run a command on all service nodes only, as follows:

cexec -f /etc/c3svc.conf <*cmd*>

Run a command on all compute nodes in the system, as follows:

cexec --all <*cmd*>

Run a command on all RLCs, as follows:

cexec --all --head <*cmd*>

Run a command on blade 42 (compute node 42) in rack 2, as follows:

cexec rack_2:42 <*cmd*>

From a **service node** over the InfiniBand Fabric, run a command on all blades (compute nodes) in the system, as follows:

cexec --all <*cmd*>

Run a command on blade 42 (compute node 42), as follows:

cexec blades:42 <*cmd*>

pdsh and pdcp Utilities

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

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

On the service node, software populates the compute group which contains all the online compute nodes in the whole system.

For more information, see the pdsh(1) and pdcp(1) man pages.

EXAMPLES

From the SAC, to run the hostname command on all the RLCs, perform the following:

pdsh -g leader hostname

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

pdsh -g leader pdsh -g compute hostname

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

pdsh -w r1lead,r2lead hostname

cadmin: SMC for SGI ICE X Administrative Interface

The cadmin command allows you to change certain administrative parameters in the cluster such as the boot order of service nodes, the administrative status of nodes, and the adding, changing, and removal of IP addresses associated with service nodes.

To get the cadmin usage statement, perform the following:

```
[sys-admin ~]# cadmin --h
cadmin: SGI Tempo Administrative Interface
Help:
```

```
In general, these commands operate on {node}.  {node} is the Tempo style
node name.  For example, service0, r1lead, r1i0n0.  Even when the host name
for a service node is changed, the Tempo name for that node may still be used
for {node} below.  The node name can either be the tempo unique node name
or a customer-supplied host name associated with a tempo unique node name.

--version : Display current release information
--set-admin-status --node {node} {value}  : Set Administrative Status
--show-admin-status --node {node} : Show Administrative Status
--set-boot-order --node {node} [value] : Set boot order [1]
--show-boot-order --node {node} : Show boot order [1]
--set-ip --node {node} --net {net} {hostname}={ip} : Change an allocated ip [1]
--del-ip --node {node} --net {net} {hostname}={ip} : Delete an ip [1]
--add-ip --node {node} --net {net} {hostname}={ip} : allocate a new ip [1]
--show-ips --node {node} : Show all allocated IPs associated with node
--set-hostname --node {node} {new-hostname} : change the host name [5]
--show-hostname --node {node} : show the current host name for ice node {node}
--set-subdomain {domain} : Set the cluster subdomain [3]
--show-subdomain : Show the cluster subdomain
--set-admin-domain {domain} : Set the admin node house network domain
--show-admin-domain : Show the admin node house network domain
--db-purge --node {node} : Purge service or lead node (incl entire rack) from DB
--set-external-dns --ip {ip} : Set IP addr(s) of external DNS master(s) [4]
--show-external-dns : Show the IP addr(s) of the external DNS master(s)
--del-external-dns : Delete the configuration of external DNS master(s)
--show-root-labels : Show grub root labels if multiple roots are in use
--set-root-label --slot {#} --label {label} : Set changeable part of root label
--show-default-root : Show default root if multiple roots are in use
--set-default-root --slot {#} : Set the default slot if multiple roots in use
--show-current-root : Show current root slot
--enable-auto-recovery : Enable ability for nodes to recover themselves [6]
--disable-auto-recovery : Disable auto recovery [6]
--show-auto-recovery : Show the current state of node auto recovery [6]
--enable-redundant-mgmt-network --node {node}: Enable network
    management redundancy
--disable-redundant-mgmt-network --node {node}: Disable management network
    redundancy
--show-redundant-mgmt-network --node {node}:  Show current value.
--show-dhcp-option: Show admin dhcp option code used to distinguish mgmt network
--set-dhcp-option {value}: Set admin dhcp option code
--enable-switch-mgmt-network --node {node}: Enable switch management network
```

```
    for a node that is connected to managed top level switches. Not for ICE
    8200/8400 nodes. [7]
--disable-switch-mgmt-network --node {node}: Disable switch management network
    for the specified node.
--show-switch-mgmt-network --node {node}:  Show current value.
--enable-replication --node {node}: Enable MySQL Replication on the specified
  admin, leader, or service node. [8]
--disable-replication --node {node}: Disable MySQL Replication on the specified
  admin, leader, or service node.
--show-replication-status --node {node}:  Show current value.
--set-mgmt-bonding --node {node} {value}: "802.3ad" or "active-backup"
   Must be "active-backup" ICE 8200/8400 systems. Depends on mgmt switch.
--show-mgmt-bonding --node {node}:  Show current value.
--set-max-rack-irus --node {node} {value}: Max IRUs/CMCs for this leader node.
   Saves push-rack time by not doing expanisons for nodes that do not exist.
--show-max-rack-irus --node {node}:  Show current value
--set-blademond-rescan --node {node} {value}: rescan interval for this leader.
   Configures how many seconds blademond waits between checks for blade changes.
--show-blademond-rescan --node {node}:  Show current value

Node-attribute options:
--add-attribute [--string-data "{string}"] [--int-data {int}] {attribute-name}
--is-attribute {attribute-name}
--delete-attribute {attribute-name}
--set-attribute-data [--string-data "{string}"] [--int-data {int}]
  {attribute-name}
--get-attribute-data {attribute-name}
--search-attributes [--string-data "{string|regex}"] [--int-data {int}]
--add-node-attribute [--string-data "{string}"] [--int-data {int}]
  --node {node} --attribute {attribute-name}
--is-node-attribute --node {node} --attribute {attribute-name}
--delete-node-attribute --node {node} --attribute {attribute-name}
--set-node-attribute-data [--string-data "{string}"] [--int-data {int}]
  --node {node} --attribute {attribute-name}
--get-node-attribute-data --node {node} --attribute {attribute-name}
--search-node-attributes [--node {node}] [--attribute {attribute-name}]
  [--string-data "{string|regex}"] [--int-data {int}]

Descriptions of Selected Values:
{hostname}={ip} means specify the host name associated with the specified
ip address.
```

```
{net} is the tempo network to change such as ib-0, ib-1, head, gbe, bmc, etc
{node} is a tempo-style node name such as r1lead, service0, or r1i0n0.
[1] Only applies to service/ibswitch/mgmtswitch nodes
[2] This operation may require the cluster to be fully shut down and AC power
    to be removed.  IPs will have to be re-allocated to fit in the new range.
[3] All cluster nodes will have to be reset. Compute images need to be pushed
    again with the cimage command.
[4] Use quoted, semi-colon separated list if more than one master
[5] Only applies to admin and service/ibswitch/mgmtswitch nodes
[6] Auto recovery will allow service and leader nodes to boot in to a special
    recovery mode if the cluster doesn't recognize them.  This is enabled by
    default and would be used, for example, if a node's main board was replaced
    but the original system disks were imported from the original system.
[7] The global value for this is automatically detected and adjustable using
    configure-cluster. This per-node value is used for systems that mix the
    older cascaded CMC management network found in ICE 8200/8400 with the newer
    top level switch management network. In a mixed system, all leader and
    managed service nodes in the 8200/8400 part of the system should have this
    set to no. This can also be specified using the "discover" command.
[8] Node should be admin, leader, or a service node. Use configure-cluster to
    globally configure MySQL Replication on the whole system. If MySQL
    Replication is disabled on the admin node, replication will be off for the
    whole system, even if some nodes may have the replication attribute set to
    "yes". Use this feature with caution, if replication is enabled on a leader
    or service node, but disabled on the admin node, the leader or service node
    will not be able to get the information from the admin node, and the
    database calls will fail or return out of date information. To enable MySQL
    Replication on a limited number of node, disable it globally from within
    configure-cluster, then use cadmin to enable it on the admin node, and then
    on the other nodes.
```

EXAMPLES

Example 1-7 SMC for SGI ICE X Administrative Interface (`cadmin`) Command

Set a node offline, as follows:

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

Set a node online, as follows:

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

Set the boot order for a service node, as follows:

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

Add an IP to an existing service node, as follows:

```
# cadmin --add-ip --node service0 --net ib-0 my-new-ib0-ip=10.148.0.200
```

Change the SMC for SGI ICE X needed `service0-ib0` IP address, as follows:

```
# cadmin --set-ip --node service0 --net head service0=172.23.0.199
```

Show currently allocated IP addresses for `service0`, as follows:

```
# cadmin --show-ips --node service0
IP Address Information for SMC for SGI ICE X node: service0

ifname        ip              Network

myservice-bmc 172.24.0.3      head-bmc
myservice     172.23.0.3      head
myservice-ib0 10.148.0.254    ib-0
myservice-ib1 10.149.0.67     ib-1
myhost        172.24.0.55     head-bmc
myhost2       172.24.0.56     head-bmc
myhost3       172.24.0.57     head-bmc
```

Delete a site-added IP address (you cannot delete SMC for SGI ICE X needed IP addresses), as follows:

```
admin:~ # cadmin --del-ip --node service0 --net ib-0 my-new-ib0-2-ip=10.148.0.201
```

Change the hostname associated with `service0` to be `myservice`, as follows:

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

Change the hostname associated with `admin` to be `newname`, as follows:

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

Set and show the cluster subdomain, as follows:

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

Show the system admin controller (SAC) house network domain, as follows:

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

Show the SMC for SGI ICE X systems DHCP option identifier, as follows:

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

Show the current switch management value for a specified node, as follows:

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

Enable the switch management network for a specificed node that is connected to managed top level switches, as follows:

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

Disable the switch management network for a specificed node that is connected to managed top level switches, as follows:

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

Show MySQL replication status for a specified system admin controller (SAC), rack leader controller (RLC), or service node, as follows:

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

Note: MySQL replication is disabled by default.

Enable MySQL replication on a specified SAC, RLC, or service node, as follows:

```
admin:~ # cadmin --enable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...
mysql                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
Restarting service MySQL
Shutting down service MySQL ..done
Starting service MySQL ..done
```

Disable MySQL replication on a specified SAC, RLC, or service node, as follows:

```
admin:~ # cadmin --disable-replication --node r2lead
Running 'ssh r2lead /etc/opt/sgi/conf.d/80-update-mysql' ...
Shutting down service MySQL ..done
mysql                    0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Console Management

SMC for SGI ICE X management systems software uses the open-source console management package called `conserver`. For detailed information on `consever`, see http://www.conserver.com/

An overview of the conserver package is, as follows:

- Manages the console devices of all managed nodes in an SGI ICE X system
- A `conserver` daemon runs on the system admin controller (SAC) and the rack leader controllers (RLCs). The SAC manages RLC and service node consoles. The RLCs manage blade consoles.
- The `conserver` daemon connects to the consoles using `ipmitool`. Users connect to the daemon to access them. Multiple users can connect but non-primary users are read-only.
- The `conserver` package is configured to allow all consoles to be accessed from the SAC.
- All consoles are logged. These logs can be found at `/var/log/consoles` on the SAC and RLCs. An `autofs` configuration file is created to allow you to access RLC-managed console logs from the SAC, as follows:

  ```
  admin #  cd /net/r1lead/var/log/consoles/
  ```

The `/etc/conserver.cf` file is the configuration file for the conserver daemon. This file is generated for both the SAC and the RLCs from the `/opt/sgi/sbin/generate-conserver-files` script on the SAC. This script is called from `discover-rack` command as part of rack discovery or rediscovery and generates both the `conserver.cf` file for the rack in question and regenerates the `conserver.cf` for the SAC.

Note: The conserver package replaces `cconsole` for access to all consoles (blades, RLCs, managed service nodes)

You may find the following `conserver` man pages useful:

Man Page	Description
`console`(1)	Console server client program
`conserver`(8)	Console server daemon
`conserver.cf`(5)	Console configuration file for `conserver`(8)
`conserver.passwd`(5)	User access information for `conserver`(8)

Procedure 1-17 Using `conserver` Console Manager

To use the `conserver` console manager, perform the following steps:

1. To see the list of available consoles, perform the following:

```
admin:~ #console -x
 service0                on /dev/pts/2                    at  Local
 r2lead                  on /dev/pts/1                    at  Local
 r1lead                  on /dev/pts/0                    at  Local
 r1i0n8                  on /dev/pts/0                    at  Local
 r1i0n0                  on /dev/pts/1                    at  Local
```

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

```
admin:~ # console service0
[Enter '^Ec?' for help]


Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64) - Kernel 2.6.16.60-0.12-smp (ttyS1).


service0 login:
```

3. To connect to the RLC console, perform the following:

```
admin:~ # console r1lead
[Enter '^Ec?' for help]


Welcome to SUSE Linux Enterprise Server 10 sp2 (x86_64)
- Kernel 2.6.16.60-0.12-smp (ttyS1).


r1lead login:
```

4. To trigger system request commands `sysrq` (once connected to a console), perform the following:

```
Ctrl-e c l 1 8               # set log level to 8
Ctrl-e c l 1 <sysrq cmd>            # send sysrq command
```

5. To see the list of `conserver` escape keys, perform the following:

```
Ctrl-e c ?
```

Keeping System Time Synchronized

The SMC for SGI ICE X systems management software uses network time protocol (NTP) as the primary mechanism to keep the nodes in your SGI ICE X system synchronized. This section describes this mechanism operates on the various SGI ICE X components and covers these topics:

System Admin Controller (SAC) NTP

When you used the `configure-cluster` command, it guided you through setting up NTP on the SAC. The NTP client on the SAC should point to the house network time server. The NTP server provides NTP service to system components so that nodes can consult it when they are booted. The SAC sends NTP broadcasts to some networks to keep the nodes in sync after they have booted.

Rack Leader Controller (RLC) NTP

NTP client on the RLC gets time from the system admin controller (SAC) when it is booted and then stays in sync by connecting to the SAC for time. The NTP server on the leader node provides NTP service to SGI ICE components so that compute nodes can sync their time when they are booted. The RLC sends NTP broadcasts to some networks to keep the compute nodes in sync after they have booted.

Managed Service, Compute, and Rack Leader Controller (RLC) BMC Setup with NTP

The BMC controllers on managed service nodes, compute nodes, and RLCs are also kept in sync with NTP. Note that you may need the latest BMC firmware for the BMCs to sync with NTP properly. The NTP server information for BMCs is provided by special options stored in the DHCP server configuration file.

Service Node NTP

The NTP client on *managed* service nodes (for a definition of managed, see "`discover` Command" on page 10) sets its time at initial booting from the system admin controller (SAC). It listens to NTP broadcasts from the SAC to stay in sync. It does not provide any NTP service.

Compute Node NTP

The NTP Client on the compute node sets its time at initial booting from the rack leader controller (RLC). It listens to NTP broadcasts from the RLC to stay in sync.

NTP Work Arounds

Sometime, especially during initial deployment of an SGI ICE X system when system components are being installed and configured for the first time, NTP is not available to serve time to system components.

A non-modified NTP server, running for the first time, takes quite some time before it offers service. This means the rack leader controllers (RLCs) and service nodes may fail to get time from the system admin controller (SAC) as they come online. Compute nodes may also fail to get time from the RLC when they first come up. This situation usually only happens at first deployment. After the `ntp` servers have a chance to create their drift files, `ntp` servers offer time with far less delay on subsequent reboots.

The following work arounds are in place for situations when NTP can not serve the time:

- The SAC and RLCs have the `time` service enabled (`xinetd`).
- All system node types have the `netdate` command.
- A special startup script is on RLC, service, and compute nodes that runs before the NTP startup script.

  This script attempts to get the time using the `ntpdate` command. If the `ntpdate` command fails because the NTP server it is using is not ready yet to offer time service, it uses the `netdate` command to get the clock close.

  The `ntp` startup script starts the NTP service as normal. Since the clock is known to be close, NTP fixes the time when the NTP servers start offering time service.

Changing the Size of `/tmp` on Compute Nodes

This section describes how to change the size of `/tmp` on SGI ICE X compute nodes.

Procedure 1-18 Increasing the `/tmp` Size

To change the size of `/tmp` on your system compute nodes, perform the following steps:

1. From the system admin controller (SAC), use the `cd`(1) command to change to the following directory:

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

2. Open the `sgi-fstab` file and change the `size=` parameter for the `/tmp` mount in both locations that it appears.

```
#!/bin/sh
#
# Copyright (c) 2007,2008 Silicon Graphics, Inc.
# All rights reserved.
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation; either version 2 of the License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307 USA
#
# Set up the compute node's /etc/fstab file.
#
# Modify per your sites requirements.
#
# This script is excecuted once per-host as part of the install-image operation
# run on the leader nodes, which is called from cimage on the admin node.
# The full path to the per-host iru+slot directory is passed in as $1,
# e.g. /var/lib/sgi/per-host/<imagename>/i2n11.
#

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

iruslot=$1
os=( $(/opt/oscar/scripts/distro-query -i ${iruslot} | sed -n '/^compat /s/^compat.*: //p') )

compatdistro=${os[0]}${os[1]}

if [ ${compatdistro} = "sles10" -o ${compatdistro} = "sles11" ]; then
```

```
        #
        # SLES 10 compatible
        #
        cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
EOF

elif [ ${compatdistro} = "rhel5"  ]; then

        #
        # RHEL 5 compatible
        #

        #
        # RHEL expects several subsys directories to be present under /var/run
        # and /var/lock, hence no tmpfs mounts for them
        #
        cat <<EOF >${iruslot}/etc/fstab
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
tmpfs           /tmp            tmpfs   size=150m       0       0
devpts          /dev/pts        devpts  gid=5,mode=620  0       0
EOF

else

        echo -e "\t$(basename ${0}): Unhandled OS.  Doing nothing"

fi
```

3. Push the image out to the racks to pick up the change, as follows:

   # **cimage --push-rack mynewimage r***

   For more information on using the cimage command, see "cimage Command" on page 40.

Enabling or Disabling the Compute Node iSCSI Swap Device

This section describes how to enable or disable the internet small computer system interface (iSCSI) compute node swap device. The iSCSI compute node swap device is turned off by default for new installations. It can cause problems during rack-wide out of memory (OOM) conditions, with both compute nodes and the rack leader controller (RLC) becoming unresponsive during the heavy write-out to the per-node iSCSI swap devices.

Procedure 1-19 Enabling the iSCSI Swap Device

If you wish to enable the iSCSI swap device in a given compute node image, perform the following steps:

1. Change root (`chroot`) into the compute node image on the system admin controller (SAC) and enable the `iscsiswap` service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap on
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Procedure 1-20 Disabling the iSCSI Swap Device

To disable the iSCSI swap device in a compute node image where it is currently enabled, perform the following steps:

1. Disable the service, as follows:

```
# chroot /var/lib/systemimager/images/compute-sles11 chkconfig iscsiswap off
```

2. Then, push the image out to the racks, as follows:

```
# cimage --push-rack compute-sles11 r\*
```

Changing the Size of Per-node Swap Space

This section describes how to change per-node swap space on your SGI ICE system.

Procedure 1-21 Increasing Per-node Swap Space

To increase the default size of the per-blade swap space on your system, perform the following:

1. Shutdown all blades in the affected rack (see "Shutting Down and Booting" on page 59).

2. Log into the rack leader controller (RLC) for the rack in question. (Note that you need to do this on each RLC).
3. Change directory (`cd`) to the `/var/lib/sgi/swapfiles` directory.
4. To adjust the swap space size appropriate for your site, run a script similar to the following:

```
#!/bin/bash

size=262144     # size in KB

for i in $(seq 0 3); do
        for n in $(seq 0 15); do
                dd if=/dev/zero of=i${i}n${n} bs=1k count=${size}
                mkswap i${i}n${n}
        done
done
```

5. Reboot the all blades in the affected rack (see "Shutting Down and Booting" on page 59).
6. From the RLC, use the `cexec --all` command to run the `free`(1) command on the compute blades to view the new swap sizes, as follows:

```
r1lead:~ # cexec --all free
************************ rack_1 ************************
--------- r1i0n0---------
             total       used       free     shared    buffers     cached
Mem:       2060140     206768    1853372          0          4      46256
-/+ buffers/cache:     160508    1899632
Swap:        49144          0      49144
--------- r1i0n1---------
             total       used       free     shared    buffers     cached
Mem:       2060140     137848    1922292          0          4      44200
-/+ buffers/cache:      93644    1966496
Swap:        49144          0      49144
--------- r1i0n8---------
             total       used       free     shared    buffers     cached
Mem:       2060140     138076    1922064          0          4      43172
-/+ buffers/cache:      94900    1965240
Swap:        49144          0      49144
```

If you want change per-node swap space across your entire system, all (new) RLCs as part of discovery, you can edit the `/etc/opt/sgi/conf.d/35-compute-swapfiles` "inside" the `lead-sles11` image on the system admin controller (SAC). The images are in the `/var/lib/systemimager/images` directory. For more information on customizing these images, see "Customizing Software Images" on page 37.

Switching Compute Nodes to a `tmpfs` Root

This section describes how to switch your system compute nodes to a `tmpfs` root.

Procedure 1-22 Switching Compute Nodes to a `tmpfs` Root

To switch your compute nodes to a `tmpfs` root, from the system admin controller (SAC) perform the following steps:

1. To switch compute nodes to a `tmpfs` root, use the optional `--tmpfs` flag to the `cimage --set` command, for example:

```
adminadmin:~ # cimage --set --tmpfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

Note: To use a `/tmpfs` root with the standard compute node image, the compute node needs to have 4GB of memory or above. A standard `/tmpfs` mount has access to half the system memory, and the standard compute node image is just over 1 GB in size.

---

2. You can view the current setting of a compute node, as follows:

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp tmpfs
```

3. To set it back to an NFS root, use the `--nfs` flag to the `cimage --set` command, as follows:

```
admin:~ # cimage --set --nfs compute-sles11 2.6.27.19-5-smp r1i0n0
```

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

```
admin:~ # cimage --list-nodes r1i0n0
r1i0n0: compute-sles11 2.6.27.19-5-smp nfs
```

For help information, use the `cimage --h` option.

Setting up Local Storage Space for Swap and Scratch Disk Space

The SGI ICE X system has the option to support local storage space on compute nodes (also known as blades). Solid state drive (SSD) devices and 2.5" disks are available for this purpose. You can define the size and status for both swap and scratch partitions. The values can be set globally or per node or group of nodes. By default, the disks are partitioned only if blank, the swap is off, and the scratch is set to occupy the whole disk space and be mounted in `/tmp/scratch`.

The `/etc/init.d/set-swap-scratch` script is responsible for auto-configuring the swap and scratch space based on the settings retrieved via the `cattr` command. You can use the `cadmin` to configure settings globally or you can use the `cattr` command to set custom values for specific nodes.

The `/etc/opt/sgi/conf.d/30-set-swap-scratch` script makes sure `/etc/init.d/swapscratch` service is on so that swap/scratch partitions are configured directly after booting. The `swapscratch` service calls the `/opt/sgi/lib/set-swap-scratch` script when the service is started and then it exits.

You can customize the following settings:

- `blade_disk_allow_partitioning`

  The default value is `on`, which means that the `set-swap-scratch` script will repartition and format the local storage disk if needed.

  Note: To protect user data, the script will not re-partition the disk if it is already partitioned. In this case, you need a blank disk before it can be used for swap/scratch.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_allow_partitioning` value for the node on which it is running:

  ```
  # cattr get blade_disk_allow_partitioning -N $compute_node_name --default on
  ```

  You can globally set the value on, as follows:

  ```
  # cadmin --add-attribute --string-data on blade_disk_allow_partitioning
  ```

  You can globally turn if off, as follows:

  ```
  # cadmin --add-attribute --string-data off blade_disk_allow_partitioning
  ```

- `blade_disk_swap_status`

  The default value is `off` which means that the `set-swap-scratch` script will not enable a swap partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_swap_status` value for the node on which it is running:

```
# cattr get blade_disk_swap_status -N $compute_node_name --default off
```

  You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_swap_status
```

  You can globally turn if off, as follows:

```
# cadmin --add-attribute --string-data off blade_disk_swap_status
```

  The `set-swap-scratch` script uses `SGI_SWAP` label when partitioning the disk. It enables the swap only if it finds a partition labeled `SGI_SWAP`.

- `blade_disk_swap_size`

  The default value is 0 which means that the `set-swap-scratch` script will not create a swap partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_swap_size` value for the node on which it is running:

```
attr get blade_disk_swap_size -N $compute_node_name --default 0
```

  You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data 1024 blade_disk_swap_size
```

  The size is specified in megabytes. Allowed values are, as follows: 0, -0 (use all free space when partitioning), 1, 2, ...

- `blade_disk_scratch_status`

  The default value is `off` which means that the `set-swap-scratch` script will not enable the scratch partition on the local storage disk.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_status` value for the node on which it is running:

```
cattr get blade_disk_scratch_status -N $compute_node_name --default off
```

You can globally set the value on, as follows:

```
# cadmin --add-attribute --string-data on blade_disk_scratch_status
```

You can globally turn if off, as follows:

```
cadmin --add-attribute --string-data off blade_disk_scratch_status
```

Note: The `set-swap-scratch` script uses the `SGI_SCRATCH` label when partitioning the disk. It mounts the scratch only on the partition labeled as `SGI_SCRATCH`.

- `blade_disk_scratch_size`

  The default value is `-0` which means that the `set-swap-scratch` script will use all remaining free space when creating the scratch partition.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

  ```
  cattr get blade_disk_scratch_size -N $compute_node_name --default -0
  ```

  You can globally set the value, as follows:

  ```
  cadmin --add-attribute --string-data 10240 blade_disk_scratch_size
  ```

  The size is specified in megabytes. Allowed values are, as follows: 0, -0 (use all free space when partitioning), 1, 2, ...

- `blade_disk_scratch_mount_point`

  The default value is `/tmp/scratch` which means that the `set-swap-scratch` script will mount the scratch partition in `/tmp/scratch`.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_scratch_size` value for the node on which it is running:

```
# cattr get blade_disk_scratch_mount_point -N $compute_node_name --default /tmp/scratch
```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data /tmp/scratch blade_disk_scratch_mount_point
```

You can mount the disk to any mount point you desire. The `set-swap-scratch` script will create that folder if it does not exists (as long as the script has the permission to create it at that path). The root mount point (/) is not writable on the compute nodes. You need to create that folder as part of the compute node image if you want to mount something like `/scratch`.

- `blade_disk_raid_level`

  The default value is `off`. When set to `0`, it allows you to set up RAID0, if you have two disk for swap/scratch. Values of `1`, `...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_raid_level` value for the node on which it is running:

  ```
  attr get blade_disk_raid_level -N $compute_node_name --default 0
  ```

  You can globally set the value, as follows:

  ```
  # cadmin --add-attribute --string-data blade_disk_raid_level
  ```

- `blade_disk_reformat_swap_at_boot`

  The default value is `off`. When set to `0`, it allows you to format the swap every time the node reboots. Values of `1`, `...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_swap_at_boot` value for the node on which it is running:

  ```
  attr get blade_disk_reformat_swap_at_boot -N $compute_node_name --default 0
  ```

  You can globally set the value, as follows:

  ```
  # cadmin --add-attribute --string-data blade_disk_reformat_swap_at_boot
  ```

- `blade_disk_reformat_scratch_at_boot`

  The default value is `off`. When set to `0`, it allows you to format the scratch every time the node reboots. Values of `1`, `...` are currently ignored.

  The `set-swap-scratch` script uses the following command to retrieve the `blade_disk_reformat_scratch_at_boot` value for the node on which it is running:

  ```
  attr get blade_disk_reformat_scratch_at_boot -N $compute_node_name --default 0
  ```

You can globally set the value, as follows:

```
# cadmin --add-attribute --string-data blade_disk_reformat_scratch_at_boot
```

For a `cattr` command help statement, perform the following command:

```
# cattr -h
Usage:
  cattr [--help] COMMAND [ARG]...

Commands:
  exists   check for the existence of an attribute
  get      print the value of an attribute
  list     print a list of attribute values
  set      set the value of an attribute
  unset    delete the value of an attribute

For more detailed help, use 'cattr COMMAND --help'.
```

Viewing the Compute Node Read-Write Quotas

This section describes how to view the per compute node read and write quota.

Procedure 1-23 Viewing the Compute Node Read-Write Quotas

To view the per compute node read and write quota, log onto the rack leader controller (RLC) and perform the following:

```
r1lead:~ # xfs_quota -x -c 'quota -ph 1'
Disk quotas for Project #1 (1)
Filesystem   Blocks  Quota  Limit Warn/Time    Mounted on
/dev/disk/by-label/sgiroot
               64.6M      0     1G  00 [------] /
```

Map the XFS project ID to the quota you are interested in by looking it up in `/etc/projects` file.

If you decided to change the `xfs_quota` values, log back onto the system admin controller (SAC) and edit the `/etc/opt/sgi/cminfo` file **inside** the compute image where you want to change the value, for example,

/var/lib/systemimager/images/*image_name*. Change the value of the PER_BLADE_QUOTA variable and then repush the image with the following command:

```
# cimage --push-rack image_name racks
```

For help information, perform the following:

```
xfs_quota> help
df [-bir] [-hn] [-f file] -- show free and used counts for blocks and inodes
help [command] -- help for one or all commands
print -- list known mount points and projects
quit -- exit the program
quota [-bir] [-gpu] [-hnv] [-f file] [id|name]... -- show usage and limits

Use 'help commandname' for extended help
```

Use help *commandname* for extended help, such as the following:

```
xfs_quota> help quota

quota [-bir] [-gpu] [-hnv] [-f file] [id|name]... -- show usage and limits

 display usage and quota information

 -g -- display group quota information
 -p -- display project quota information
 -u -- display user quota information
 -b -- display number of blocks used
 -i -- display number of inodes used
 -r -- display number of realtime blocks used
 -h -- report in a human-readable format
 -n -- skip identifier-to-name translations, just report IDs
 -N -- suppress the initial header
 -v -- increase verbosity in reporting (also dumps zero values)
 -f -- send output to a file
 The (optional) user/group/project can be specified either by name or by
 number (i.e. uid/gid/projid).

xfs_quota>
```

RAID Utility

The infrastructure nodes on your SGI ICE system have LSI Logic RAID enabled by default from the factory. Prior SGI ICE systems shipped with the `lsiutil` command-line utility. SGI ICE X ships with the LSI Logic MegaRAID command line tool (see "LSI Logic MegaRAID Command-line Utility" on page 91).

LSI Logic `lsiutil` Command-line Utility

The `lsiutil` command-line utility is included with the installation for the system admin controller (SAC), the rack leader controller (RLC), and the service node (when installed from the SGI service node image). This tool allows you to look at the devices connected to the RAID controller and manage them. Some functions, such as, setting up mirrored or striped volumes, can be handled either by the LSI BIOS configuration tool or the `lsiutil` utility.

Note: These instructions only apply to Altix XE250 or Altix XE270 systems with the 1068-based controller. They do not apply to Altix XE250 or Altix XE270 systems that have the LSI Megaraid controller.

Example 1-8 Using the `lsiutil` Utility

The following `lsiutil` command-line utility example shows a sample session, as follows:

Start the `lsiutil` tool, as follows:

```
admin:~ # lsiutil

LSI Logic MPT Configuration Utility, Version 1.54, January 22, 2008

1 MPT Port found

     Port Name         Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
 1.  /proc/mpt/ioc0    LSI Logic SAS1068E B2     105      01140100     0

Select a device:  [1-1 or 0 to quit]
```

Select **1** to show the **MPT Port**, as follows:

```
1 MPT Port found
```

```
     Port Name         Chip Vendor/Type/Rev    MPT Rev  Firmware Rev  IOC
 1.  /proc/mpt/ioc0    LSI Logic SAS1068E B2     105      01140100     0

Select a device:  [1-1 or 0 to quit] 1

 1.  Identify firmware, BIOS, and/or FCode
 2.  Download firmware (update the FLASH)
 4.  Download/erase BIOS and/or FCode (update the FLASH)
 8.  Scan for devices
10.  Change IOC settings (interrupt coalescing)
13.  Change SAS IO Unit settings
16.  Display attached devices
20.  Diagnostics
21.  RAID actions
22.  Reset bus
23.  Reset target
42.  Display operating system names for devices
45.  Concatenate SAS firmware and NVDATA files
60.  Show non-default settings
61.  Restore default settings
69.  Show board manufacturing information
97.  Reset SAS link, HARD RESET
98.  Reset SAS link
99.  Reset port
 e   Enable expert mode in menus
 p   Enable paged mode in menus
 w   Enable logging

Main menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **21. RAID** actions, as follows:

```
Main menu, select an option:  [1-99 or e/p/w or 0 to quit] 21

 1.  Show volumes
 2.  Show physical disks
 3.  Get volume state
 4.  Wait for volume resync to complete
23.  Replace physical disk
26.  Disable drive firmware update mode
27.  Enable drive firmware update mode
30.  Create volume
```

```
31.  Delete volume
32.  Change volume settings
50.  Create hot spare
99.  Reset port
 e   Enable expert mode in menus
 p   Enable paged mode in menus
 w   Enable logging

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **2. Show physical disks**, to show the status of the disks making up the volume, as follows:

```
RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit] 2

1 volume is active, 2 physical disks are active

PhysDisk 0 is Bus 0 Target 1
  PhysDisk State:  online
  PhysDisk Size 238475 MB, Inquiry Data:  ATA      Hitachi HDT72502 A73A

PhysDisk 1 is Bus 0 Target 2
  PhysDisk State:  online
  PhysDisk Size 238475 MB, Inquiry Data:  ATA      Hitachi HDT72502 A73A

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

Choose **1. Show volumes**, to show information about the volume including its health, as follows:

```
RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit] 1

1 volume is active, 2 physical disks are active

Volume 0 is Bus 0 Target 0, Type IM (Integrated Mirroring)
  Volume Name:
  Volume WWID:  09195c6d31688623
  Volume State:  optimal, enabled
  Volume Settings:  write caching disabled, auto configure
  Volume draws from Hot Spare Pools:  0
  Volume Size 237464 MB, 2 Members
  Primary is PhysDisk 1 (Bus 0 Target 2)
```

```
  Secondary is PhysDisk 0 (Bus 0 Target 1)

RAID actions menu, select an option:  [1-99 or e/p/w or 0 to quit]
```

LSI Logic MegaRAID Command-line Utility

This section provides a brief description of the LSI Logic MegaRAID command-line utility. There is also a graphical version available that you can download and install should you choose to.

For an MegaRAID help statement, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -h
```

To show physical disks, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -pdInfo -PhysDrv[252:0] -a0
```

To show logical disk information, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -LdPdInfo -a0
```

To show a MegaRAID summary, perform the following:

```
sys-admin ~]# /opt/MegaRAID/MegaCli/MegaCli64 -ShowSummary -a0
```

Restoring the grub Boot Loader on a Node

When grub(8) boot loader is not written to the rack leader controllers (RLCs) or any of the system service nodes or is not functioning correctly, the grub boot loader will have to be reinstalled on the master boot record (MBR) of the root drive for the node.

To rewrite grub to the MBR of the root drive on a system that is booted, issue the following grub commands:

```
# grub
grub>    root (hd0,0)
grub>    setup (hd0)
grub>    quit
```

If you cannot boot your system (and it is hanging on grub), you need to boot the node in rescue mode and then issue the following commands:

```
# mount /dev/ /system
# mount -o bind /dev /system/dev
# mount -t proc proc /system/proc # optional
# mount -t sysfs sysfs /system/sys # optional
# chroot /system
# grub
grub>    root (hd0,0)
grub>    setup (hd0)
grub>    quit
# reboot
```

Backing up and Restoring the System Database

The SMC for SGI ICE X systems management software captures the relevant data for the managed objects in an SGI ICE X system. The system database is critical to the operation of your SGI ICE X system and you need to back up the database on a regular basis.

Managed objects on an SGI ICE X include the following

- SGI ICE X system

  One SGI ICE X system is modeled as a meta-cluster. This meta-cluster contains the racks each modeled as a sub-cluster.

- Nodes

  System admin controller (SAC), rack leader controllers (RLCs), service nodes, compute nodes (blades) and chassis management control blades (CMCs) are modeled as nodes.

- Networks

  The preconfigured and potentially customized IP networks

- NICs

  The network interfaces for Ethernet and InfiniBand adapters.

- The network interfaces for Ethernet and InfiniBand adapter.

The node images installed on each particular node.

SGI recommends that you keep three backups of your system database at any given time. You should implement a rotating backup procedure following the son-father-grandfather principle.

The following procedures explain how to back up and restore the system database.

Procedure 1-24 To back up the system database

1. Log into the system admin controller (SAC) 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 1-25 To restore the system database

1. Log into the system admin controller (SAC) 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 on SGI ICE X contains code to limit EDNS logging. This

section describes how to delete this code and allow EDNS to work unrestricted and log messages.

Procedure 1-26 Enabling EDNS

To enable EDNS on your SGI ICE X system, perform the following steps:

1. Open the `/opt/sgi/lib/Tempo/Named.pm` file with your favorite editing tool.
2. To remove the limit on the `edns_udp_size` parameter, comment out or remove the following line:

   ```
   $limit_edns_udp_size = "edns-udp-size 512;";"
   ```

3. Remove the following lines so that EDNS logging is no longer disabled:

   ```
   logging {
   category lame-servers {null; };
   category edns-disabled { null; };  };
   ```

Firmware Management

The `fwmgr` tool and its associated libraries form a firmware update framework. This framework makes managing the various firmware types in a cluster easier.

A given cluster may have several types of firmware including mainboard BIOS, BMC, disk controllers, InfiniBand (`ib`) interfaces, /Ethernet NICs, network switches, and many other types.

The firmware management tools allow the firmware to be stored in a central location (firmware bundle library) to be accessed by command line or graphical tools. The tools allow you to add firmware to the library, remove firmware from the library, install firmware on a given set of nodes, and other related operations.

License Requirement

This framework is licensed. It cannot be used without the appropriate license.

Terminology

This section describes some terminology associated with the firmware management, as follows:

- Raw firmware file

  These are files that you download, likely from SGI, that include the firmware and option tools to flash said firmware. For example, a raw firmware file for an SGI ICE X compute node BIOS update might be downloaded as, `sgi-ice-blade-bios-2009.12.14-1.x86_64.rpm`.

- Firmware bundle

  A firmware bundle is a file that contains the firmware to be flashed in a way that the integrated tools understand. Normally, firmware bundles are stored in the firmware bundle library (see below). However, these bundles can also be checked out of the library and accessed directly in some cases. In most situations, a firmware bundle is a sort of wrapper around the raw firmware file(s) and various attributes and tools. A firmware bundle can contain more than one type of firmware. This is the case when the underlying flash tool supports more than one firmware type. An example of this is the SGI ICE X compute node firmware, that contains several different BIOS files for different mainboards and multiple BMC firmware revisions. Another example might be a raw file that includes both the BIOS and BMC firmware for a given mainboard/server.

- Firmware bundle library

  This is a storage repository for firmware bundles. The management tools allow you to query the library for available bundles and associated attributes.

- Update environment

  Some raw firmware types, like the various SGI ICE X firmware released as RPMs, run "live" on the system admin controller (SAC) to facilitate flashing. The underlying tool may indeed set nodes up to network boot a low level flash tool, but there are many other methods used by the underlying tools. Some firmware types, like BIOS ROMs with associated flash executables, require an update environment to be constructed. One type of update environment is a DOS Update Environment. This update environment may be used, for example, to construct a DOS boot image for the BIOS ROM and associated flash tool. A firmware bundle calls for a specific update environment. In this way, a firmware bundle with an associated update environment form the necessary pieces to facilitate booting of a DOS update environment over the network that flashes the target nodes with the specified BIOS ROM (as an example).

Firmware Update High Level Example

This section describes the steps you need to take to update a set of nodes in your cluster with a new BIOS level, as follows:

- Download the raw firmware file for this system type. You might do this, for example, from SGI Supportfolio web site located at https://support.sgi.com/login.
- Add the raw firmware file to the firmware bundle library using a graphical or command line tool.
- The tool will convert the raw firmware file into a firmware bundle and store it in the firmware bundle library. In some cases, you will be required to provide additional information in order to convert the raw firmware file into a firmware bundle. This could be information necessary to facilitate flashing that the framework can not derive from the file on its own.
- Once the firmware bundle is available in the firmware library, you can use the graphical or command line tool to select a firmware bundle and a list of target nodes to which to push the firmware update.
- The underlying tool then creates the appropriate update environment (if required) and facilitates flashing of the nodes.

Firmware Manager Command Line Interface (`fwmgr`)

The `fwmgr` command is the command line interface (CLI) to the firmware update infrastructure.

For a usage statement, enter `fwmgr --help`. The `fwmgr` command has several sub-commands, each of which can be called with the `--help` option for usage information.

You can use the `fwmgr` command to perform the following:

- List the available firmware bundles
- Add raw firmware files or firmware bundle files to the firmware bundle library. If it is a raw firmware type, it will be converted to a firmware bundle and placed in the library.
- Remove firmware bundles from the firmware bundle library
- Rename an existing firmware bundle in the firmware bundle library

- Install a given firmware bundle on to a list of nodes
- Checkout a firmware bundle which allows you to store the firmware bundle itself

Note: It is currently not necessary to run the `fwmgrd` command (firmware manager daemon) to use the CLI.

Firmware Manager Daemon (`fwmgrd`)

This `fwmgrd` daemon is installed and enabled by default in SGI MC 1.3 on SGI ICE X systems, only. This daemon provides the services needed for the SGI Management Center graphical user interface to communicate with the firmware management infrastructure. This daemon needs to be running in order to access firmware management from the graphical user interface.

Even if you intend to only use the CLI, it is recommended that the `fwmgrd` daemon be left running and available.

By default, the `fwmgrd` log file is located at:

`/var/log/fwmgrd.log`

View this log for important messages during flashing operations from the SGI Management Center graphical interface.

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 compute nodes within a rack and between 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 technologies:

- 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 service 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 *Message Passing Toolkit (MPT) 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 104). The InfiniBand fabric connects the service nodes, rack leader controllers (RLCs), and the compute nodes. It does not connect to the system admin controller (SAC) or the chassis management control (CMC) blades. SGI ICE X systems usually have two separate InfiniBand fabrics, which are generally referred to as `ib0` and `ib1` within this manual.

On SGI ICE X systems, each InfiniBand fabric (also sometimes called an InfiniBand subnet) has its own subnet manager, 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 113.

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 system admin controller (SAC), enter the following command:

```
admin:~ # tempo-configure-fabric
```

The **InfiniBand Management Tool** GUI appears, as shown in Figure 2-1 on page 102.

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**.

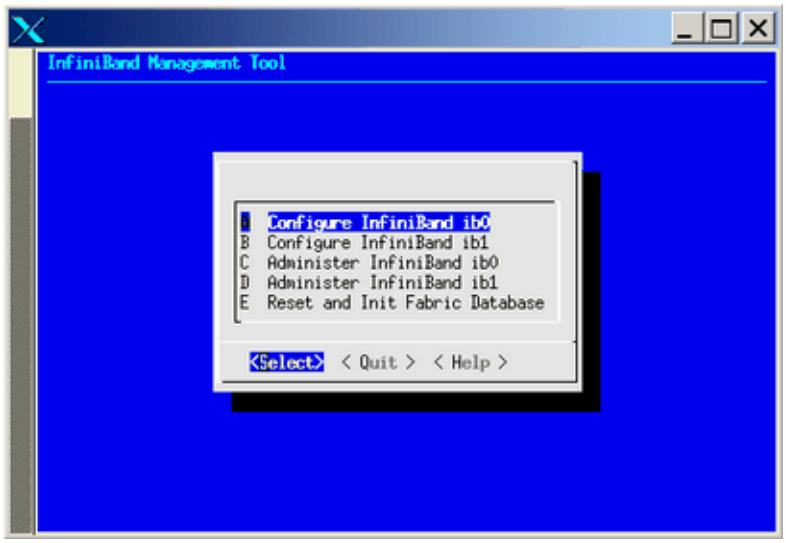

Figure 2-1 InfiniBand Management Tool Screen

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 104) to debug the problem. Alternatively, you can use the **Reset and Init Fabric Database** option from the **InfiniBand Management Tool** main menu (see Figure 2-1 on page 102) 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 2-2 on page 103. For more information, see "Network Topology" on page 109.

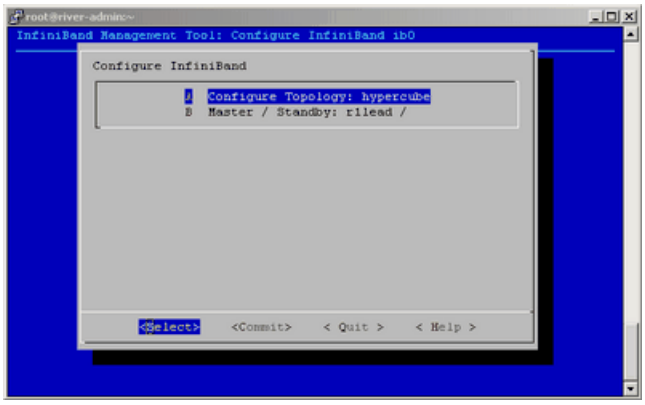

Figure 2-2 Configure Topology Screen

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 2-3 on page 103.

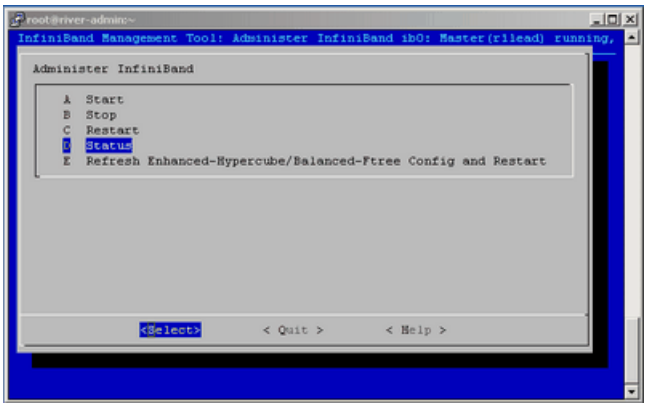

Figure 2-3 Administer InfiniBand Status Option

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 service nodes. The refresh action updates the `guid` routing order file which is used to balance InfiniBand traffic for the Enhanced Hypercube topology. In addition, this action also automatically restarts the master subnet manager and the optional standby subnet manager for the specified fabric (see "InfiniBand Fabric Failover Mechanism" on page 113).

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 101) is entirely sufficient, and recommended. The `sgifmcli`(8) command can be used for more advanced fabric management tasks.

The most common operations that `sgifmcli` would be used for are, as follows:

- Initializing and configuring external InfiniBand switches
- Verifying the integrity of the InfiniBand fabric(s)

For more information, see the `sgifmcli`(8) man page.

Currently, the following switches are supported:

Switch Type	Description
voltaire-isr-9024	Voltaire ISR 9024
voltaire-isr-2004	Voltaire ISR 2004
voltaire-isr-2012	Voltaire ISR 2012
voltaire-isr-9096	Voltaire ISR 9096
voltaire-isr-9288	Voltaire ISR 9288
voltaire4036	Voltaire Grid Director 4036
mellanox5030	Mellanox IS5030

To configure an external InfiniBand switch, cluster-wide InfiniBand connectivity is not required. The only necessity is that the supplied switch host name is resolvable and a working networking connection to the external InfiniBand switch exists. See the sgifmcli(8) man page for more information about adding external InfiniBand switches to your cluster's fabric.

Verify the integrity of an InfiniBand fabric requires that the InfiniBand network is first configured properly. This is most easily done using tempo-configure-fabric (see "InfiniBand Management Tool Graphical User Interface" on page 101). 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 system admin controller (SAC):

```
admin:~ # man sgifmcli
```

The sgifmcli(8) fabric administration utilities man page appears.

sgifmdb Fabric Management Database Command

The fabric component maintains a database (DB) of the objects it manages (managed objects). The database version is automatically set during cluster install. You do not need to set it. Most likely, this database will change over time. To manage multiple database versions and also to aid in field support, SGI has added another command line tool that currently reports the managed objects database version.

The sgifmdb command is, as follows:

```
sgifmdb [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]
```

It accepts the following general options:

General Option	Description
-g, --get	Reads the database version object from the database
-d, --dump	Dumps the database. This option allows the you to see what fabric objects are currently stored in the fabric database.
-v, --version	Prints version
-r, --reset	Resets the database and starts clean
-h, --help	-h, –help

Example 2-1 Getting sgifmdb(8) Command Help

For a sgifmdb command usage statement, perform the following from the system admin controller (SAC):

```
admin:~ # sgifmdb -h
SGI Fabric Component DB tool
Usage: db_version [--get|-g] [--dump|-d] [-v|--version] [-r|--reset] [--help|-h]
```

```
-g, --get       Read DB version object from DB
-d, --dump      Dump the DB
-v, --version   Print version
-r, --reset     Reset the database and start clean
-h, --help      Show this text
```

InfiniBand Fabric Management Configuration and Operation Overview

Each subnet manager 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 system admin controller (SAC).

Each `opensm` instance (one for each fabric) associates itself with a particular globally unique identifier (GUID) for a port on the node where `opensm` runs (see). This association is configured with the `guid` entry in the corresponding `opensm-ib[01].conf` file.

Network Topology

For SGI ICE X systems with a hypercube topology, SGI uses the dimension order routing (DOR) algorithm.

The dimension order routing algorithm is based on the min hop algorithm and so uses shortest paths. Instead of spreading traffic out across different paths with the

same shortest distance, it chooses among the available shortest paths based on an ordering of dimensions.

For SGI ICE X systems with a fat-tree topology, SGI uses `updn` as the default routing algorithm. Unicast routing algorithm (UPDN) is also based on the minimum hops to each node, but it is constrained to ranking rules.

For more information on routing variables, see the `opensm`(8) man page.

Hypercube network topology is well suited for smaller node count MPI jobs or jobs that have communication patterns that are not sensitive to bisection bandwidth. Fat-tree network topology is well suited for large node count MPI jobs that are sensitive to bi-section bandwidth.

As stated above, there are two `opensm` daemons, one for each fabric, `opensmd-ib0` and `opensmd-ib1`, respectively. They are controlled by the `init.d` scripts. Each `init.d` script has a separate configuration file for each fabric, `opensm-ib0` and `opensm-ib1`, respectively.

You can use the `sminfo` command to show the GUID of the 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 101).

Procedure 2-1 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 system admin controller (SAC) 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 2-2 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 104.

InfiniBand switches are generally classified as being of two types: edge switches and core or spine switches. Edge switches are used to connect to compute nodes. Core or spine switches are used to connect edge switches together. The integrated InfiniBand switches in SGI ICE X systems are considered to be edge switches and external InfiniBand switches used to connect these edge switches together in a fat-tree topology are considered to be spine switches.

The `sgifmcli` command allows two types of fat-tree topologies to be configured: FTREE and BFTREE. BFTREE is a balanced fat-tree. If the fat-tree topology is not balanced, choose FTREE; otherwise, choose BFTREE for a balanced fat-tree.

SGI recommends that you use the SMC for ICE X `discover` command (see "`discover` Command" on page 10) to discover external IB switches. After discovery is completed, an external switch can also be initialized and added to the InfiniBand system using the `sgifmcli` command.

The `--init` and `--add` options below are completed by the SMC for ICE X`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 2-3 Configuring InfiniBand Fat-tree Network Topology

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

1. Make sure that your switch is properly connected to the InfiniBand network. Also, make sure that the admin port of the switch is properly connected to the Ethernet network.
2. Power on the switch. See the switch manual for operation information.
3. From the system admin controller (SAC), initialize the switch. The syntax to initialize the switch is, as follows:

```
sgifmcli --init --ibswitch --model   --id  --switchtype [leaf | spine]
```

An example command is, as follows:

```
# sgifmcli --init --ibswitch --model voltaire-isr-2004  --id isr2004 --switchtype spine
```

This configures a Voltaire switch ISR2004 with hostname `isr2004` as a spine switch. `isr2004` refers to the admin port of the switch and needs to be configured previously to allow for switch access. The switch is now initialized and the root GUID from the spine switches have been downloaded.

4. From the SAC, add the switch to the fabric. The syntax to add the switch is, as follows:

   ```
   sgifmcli --add --id <fabric> --switch <hostname>
   ```

   An example command is, as follows:

   ```
   # sgifmcli --add --id ib0 --switch isr2004
   ```

   In this example, ISR2004 is connected to the `ib0` fabric.

5. For the new switch to be activated, the 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 2-4 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 10. The switch model to be used is "mellanox5030". The `discover` command supports external switches in a manner similar to racks and service nodes, except that switches do not have BMCs and there is no software to install.
3. Discover all external switches.
4. Use `tempo-configure-fabric` to configure the fabric, as described in "InfiniBand Management Tool Graphical User Interface" on page 101.

   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 101, 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 2-5 Verifying the InfiniBand Network

The fabric can be either `ib0` or `ib1`. This version of the InfiniBand verifier runs the recommended OFED test suite. In addition, the SMC for ICE X cluster view is compared with the InfiniBand cluster view and potential differences are reported.

To verify the `ibo` fabric, perform the following command:

```
# sgifmcli --verify --id <fabric>
```

For more information, see the `sgifmcli`(8) man page.

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 2-6 To retrieve information about OFED tools and diagnostics

1. Log into the system admin controller (SAC) 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:

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 3-1 Temporarily Take a Node Offline for Maintenance

To temporarily Take a node offline for maintenance, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

   # **cpower --down r1i0n0**

3. Mark the node offline, as follows:

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

4. Perform any maintenance to the blade that needs to be done.
5. Mark the node online, as follows:

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

6. Power up the node, as follows:

   # **cpower --boot r1i0n0**

7. Enable the node in the batch scheduler (depends on your batch scheduler).

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 3-2 Permanently Replace a Failed Blade

To permanently replace a failed blade (compute node), perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).
2. Power off the node, as follows:

   # **cpower --down r1i0n0**

3. Mark the node offline, as follows:

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

4. Physically remove and replace the failed blade.

5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon.

6. Set the node to boot your desired compute image (see `cimage --list-images` and "`cimage` Command" on page 40 for your options), as follows:

   # **cimage --set mycomputeimage mykernel r1i0n0**

7. Power up the node, as follows:

   # **cpower --boot r1i0n0**

8. Enable the node in the batch scheduler (depends on your batch scheduler).

Removing a Blade Permanently

This section describes how to permanently remove a blade from your SGI ICE X system.

Procedure 3-3 Permanently Remove a Blade

To permanently remove a blade from your system, perform the following steps:

1. Disable the node in the batch scheduler (depends on your batch scheduler).

2. Power off the node, as follows:

   # **cpower --down r1i0n0**

3. Mark the node offline, as follows:

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

4. Physically remove the failed blade.

5. It is not necessary to run `discover-rack` when a blade is replaced. This is handled by `blademond` daemon.

Adding a New Blade

This section describes how to add a new blade to an SGI ICE X system.

Procedure 3-4 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 40 for your options), as follows:

   # **`cimage --set mycomputeimage mykernel r1i0n0`**

4. Power up the node, as follows:

   # **`cpower --boot r1i0n0`**

5. Enable the node in the batch scheduler (depends on your batch scheduler).

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 system admin controller (SAC). When you replace a switch, you can push the saved configuration file from the SAC 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 3-5 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 SAC 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 SAC. 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 System Admin Controller (SAC), Rack Leader Controller (RLC), or Service Nodes

This section describe how to install and configure a spare SAC, RLC, or managed service node. The cold spare can be a shelf spare or a factory-installed cold spare that ships with your system. For more information on cold spare requirements and tools needed to do this procedure, see "Cold Spare System Admin Controller (SAC) or Rack Leader Controler (RLC) Availability" on page 136.

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 SGI ICE serial number is programmed into the SAC itself. If you try to migrate the SAC to a shelf spare system that does not have the correct SGI ICE system serial number programmed into it, parts of Tempo software may not work correctly.

Depending on the system ordered, your SGI ICE X system should be mounted in an SGI rack or racks. The SAC and RLC are generally installed within (or in some cases on top of) the system rack. The replacement of a failed SAC or RLC is accomplished in four basic steps:

- Identify the failed unit and disconnect system and power cables.
- Transfer the disk drives from the failed server into the cold spare unit.
- Connect the applicable cables to the cold spare server.
- Power-up the new server and restart the ICE system.

For detailed procedures on installing a cold spare, see sections"Identify the Failed Unit and Unplug all Cables" on page 137, "Transfer Disks from Existing Server to the Cold Spare" on page 140, "Migrating to a Cold Spare: Importing the Disk Volumes" on page 141 and "Migrating to a Cold Spare: Booting for the First Time on the Migrated Node" on page 143.

Note: If you are using multiple root slots repeat the procedures described in this section for each slot.

Cold Spare System Admin Controller (SAC) or Rack Leader Controler (RLC) Availability

A cold spare node is like an existing SAC or RLC, but it sits on a shelf or is a factory preinstalled node to be used in an emergency.

If the SAC or RLC node should fail, the cold spare can be swapped in to position to take over the duties of the failed node.

If you wish to make use of cold spare nodes, SGI suggests that you have both a SAC and an RLC on the shelf as available spares. Some of the reasons to have two separate nodes instead of one are (not an exhaustive list), as follows:

- The BIOS settings of a SAC and an RLC are different. For example, a SAC does not PXE boot by default. However, an RLC must PXE boot each boot. This means that the boot order is different for each type.
- The BMC of an RLC is set up to use DHCP by default. A SAC may not be set up this way.
- Given the first two items in this list, if you try to use a shelf-spare SAC as an RLC, the RLC is not discovered properly.

Shelf Spare Hardware Limitations

Currently, the hardware replacement procedure described in this section only supports SGI ICE X `ice-csn` nodes, that is, system admin controller (SAC), rack leader controller (RLC), and managed service nodes.

Tools Required

You will need a Video Graphics Array (VGA) screen and a keyboard to perform this procedure. This is because you need to interact with the LSI BIOS tool to import the root volumes. You cannot do this from an Intelligent Platform Management Interface (IPMI) serial console session because of the following:

- For rack leader controllers (RLCs), the cluster does not know the MAC addresses of the replacement BMC so there is no way for the cluster to connect to it until the migration script is run.
- The LSI BIOS tool requires the use of `Alt` characters which often do not transfer through the serial console properly.

Identify the Failed Unit and Unplug all Cables

If you identified the failed system admin controller (SAC) or rack leader controller (RLC), disconnect the cables from the failed unit. The front panel lights on the server can indicate if the unit has failed and give you information on why, see Figure 3-1 on page 138.

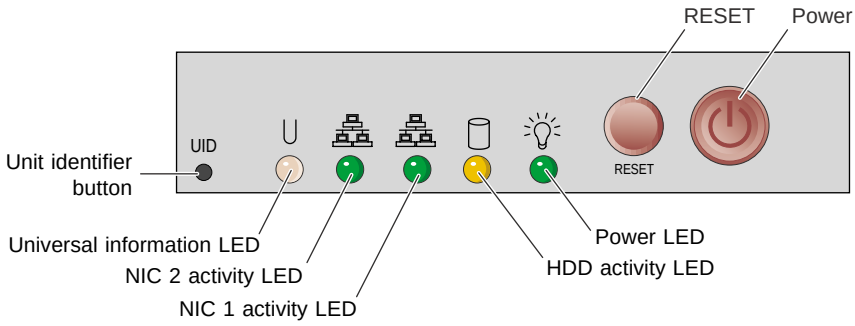

Figure 3-1 Admin/RLC Server Front Panel Controls and Indicator LEDs

The universal information LED (left side of the panel) shows two types of failure that can bring the server down. This multi-color LED blinks red quickly to indicate a fan failure and blinks red slowly for a power failure. A continuous solid red LED indicates a CPU is overheating.

If the unit's power supply has failed or been disconnected, the power LED (far right) will be dark. Check both ends of the power cable for a firm connection prior to switching over to the cold spare.

If you find that a SAC or RLC has failed and you need to replace it with a cold spare system, this section describes what to do in terms of the physical hardware.

The SAC stores the system-wide serial number. The SAC shelf spares must be ordered from the factory as SAC shelf spares so that the proper serial number can be stored within.

Procedure 3-6 Replacing a Node with a Cold Spare: Installing the Hardware

To replace a SAC or RLC that has failed, perform the following steps:

1. Power down the failed node (if possible).
2. Disconnect both power cables, see Figure 3-2 on page 140 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 141.

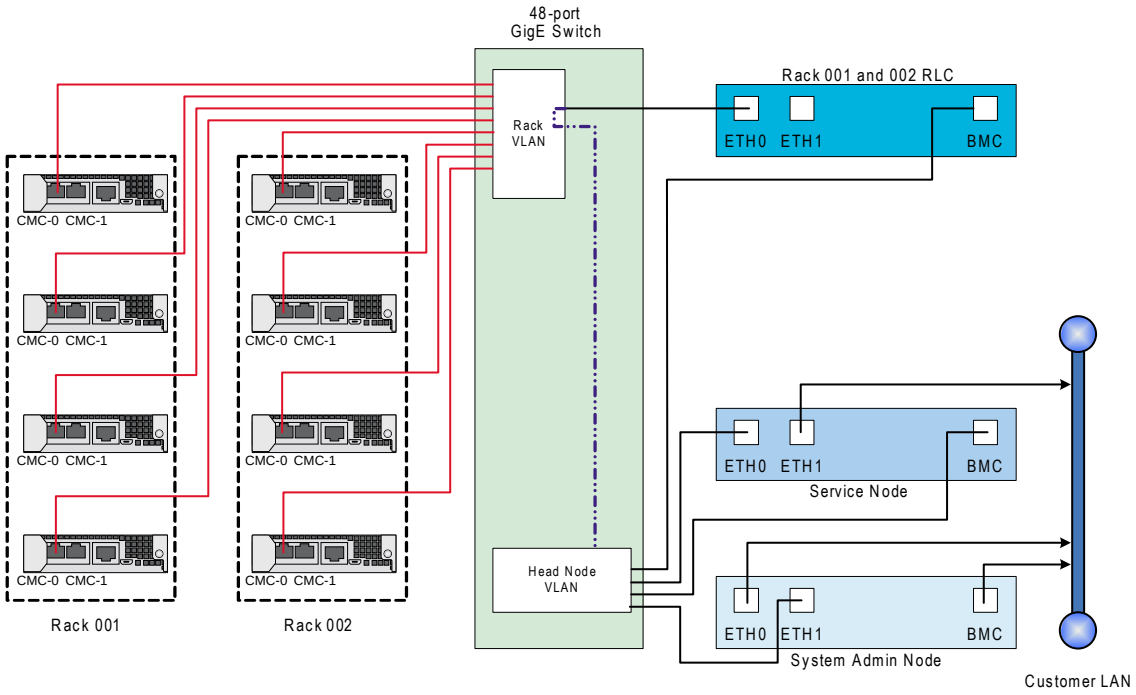

Figure 3-2 Simple CMC LAN (VLAN) Cable Examples

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 3-3 on page 141.

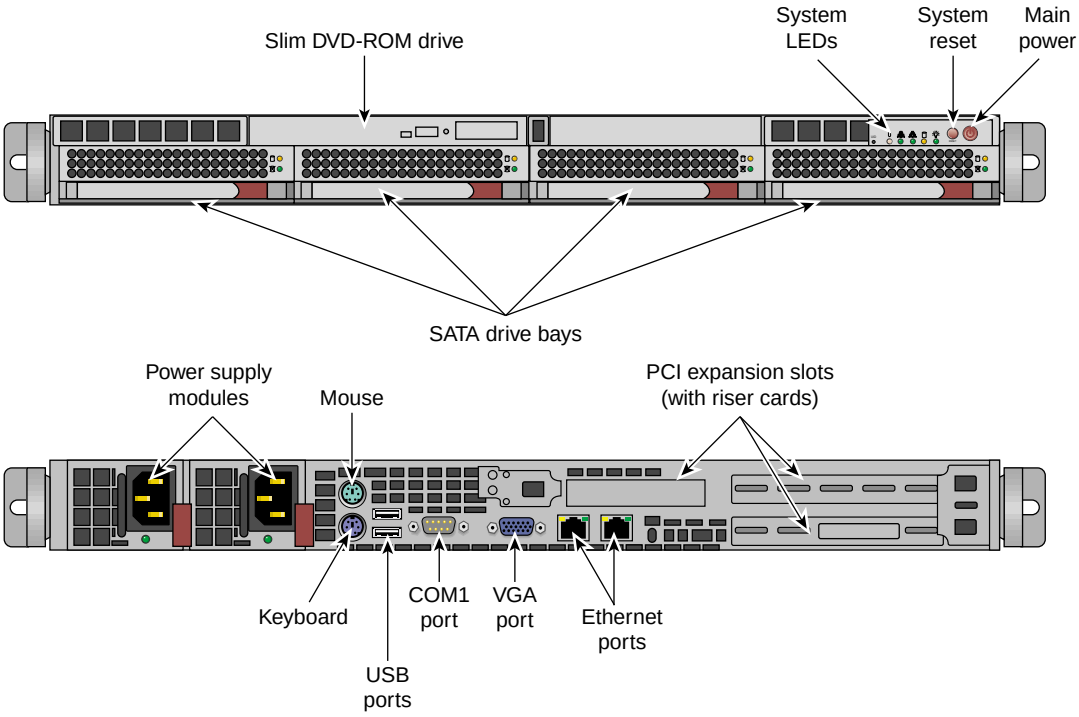

Figure 3-3 SAC and RLC Server Front Features and Rear Connector Locations

Migrating to a Cold Spare: Importing the Disk Volumes

This section describes how to import the disk volumes into the new node installed in "Identify the Failed Unit and Unplug all Cables" on page 137. 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 3-7 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 143.

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

This section provides information on booting the system for the first time on a replacement node.

Note: Important: If your site is using cascading dual boot, only the currently used slot will be updated or repaired. Therefore, if the system admin controller (SAC) is booted to slot 2, the fix up operations documented in these sections only apply to slot 2. The instructions need to be done for each slot you wish to fix up.

In a prior release, automatic recovery was implemented for cascading dual boot clusters. This means, if cascading dual boot is in use, when a managed service node or rack leader controller (RLC) boots after having procedure 5-6 performed, it will go in to an automatic recovery boot, perform some fix up, then reboot again in to its normal operating mode. For the case of the SAC, a script is run by hand to integrate the repaired SAC with the cluster.

Note: Automatic Recovery is disabled by default because it can make certain `discovery` operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]#  cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, when doing `discovery` operations, you may find it convenient to disable it.

For the case of the SAC, you will need to ensure your console output goes to the VGA screen and not serial-over-lan (SOL). For managed service nodes and RLCs in cascading dual boot clusters, the default output location during the auto recovery boot is VGA. It is best to leave it VGA since part of the repair procedure will affect the network configuration for the BMC.

How do I know which procedure to follow?

- SACs, all cases: Procedure 3-8, page 144.
- Managed Service nodes and RLCs in a non-cascading dual boot cluster: Procedure 3-8, page 144.
- Managed service, RLCs in a cascading dual boot cluster: Procedure 3-9, page 146.

Procedure 3-8 Migrating to a Cold Spare in a Non-cascading Dual Boot Cluster Node

This section describes how to boot the SAC or RLC or service node in non-cascading dual boot clusters.

Note: This section applies to SACs and sites that are **not** making use of cascading dual boot. Cascading dual boot is set up by default on newer SMC for SGI ICE X software releases. If you are using cascading dual boot, follow these instructions **only** for the SAC.

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

1. Ensure that the VGA console is powered on.
2. At this moment, the node is in the process of resetting because you exited the LSI BIOS tool at the end of the procedure, above (see "Migrating to a Cold Spare: Importing the Disk Volumes" on page 141).

   Note: After rebooting, drive 1 will resync with drive 0, automatically. Drive 1 will have the RED LED on during this time. This process takes from eight to 48 hours depending on the drive size. During that period, the RAID redundancy is not available but the system will function normally.

   When you see the GRUB boot menu come up, the first boot option will be highlighted by default. This should NOT be the choice starting with Failsafe. As an example, in SMC for SGI ICE 1.5 the highlighted choice should be : **SUSE**

Linux Enterprise Server 11 SP1. Enter **e** to edit the boot parameters for this boot only.

3. Enter **e** to edit the kernel parameters.
4. Arrow down once so that the line starting **kernel** is highlighted.
5. Look at the settings. If no serial console is defined, you do not need to change anything. If a serial console is defined, append `console=tty0` to the end of the parameter list. This will ensure that console output goes to the VGA screen for this boot.

   Note: By default, the SAC goes to the VGA screen. Therefore, this adjustment does not need to be made. RLCs and service nodes have serial consoles by default.

6. Press the `Enter` key.
7. Enter **b** to boot the system.

   The system will now boot with console output going to the VGA screen.

   Networking will fail to start and some error messages will appear.

   It is normal to see that the Ethernet devices were renumbered. This will be fixed below.

   Eventually the login prompt will appear.
8. Log in as root.
9. The following script fixes the network settings and update the SMC for ICE X database for the new network interfaces, as follows:

   # **migrate-to-shelf-spare-node**

   Note: If you have additional Ethernet cards installed, you may need to check the settings of interfaces not controlled or managed by SMC for X ICE software.

10. Reboot the node and let it boot normally.

Procedure 3-9 Migrating to a Cold Spare: Service Node or RLC Using Cascading Dual Boot

This section describes what to do for managed service nodes and RLCs in a cluster making use of cascading dual boot. It does **not** apply to SACs. For SACs, see Procedure 3-8, page 144.

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 141).

   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 141). It is true that the node network boots, but the network boot does a chainload to the first disk and it is still impacted by the BIOS and LSI firmware settings.

Migrating to a Cold Spare: Advanced Details on the Auto Recovery Mode

This section gives some advanced details on the Auto Recovery feature including how it is set up and how to control the feature.

Overview

The auto recovery feature allows managed service nodes and rack leader controllers (RLCs) to automatically make the necessary adjustments for both the node setup itself and the SMC for SGI ICE X cluster database. This feature is mainly useful for clusters

making use of cascading dual boot. The automated recovery mode applies to managed service nodes and RLCs in cascading dual boot clusters. The goal is to provide an easy way for these nodes to perform any fix ups to themselves and the SMC for SGI ICE X cluster at large when faulty systems are replaced.

Enable or Disable Auto Recovery Mode

Note: Automatic Recovery is disabled by default because it can make certain `discovery` operations harder to manage.

When you perform a field replacement operation, you can enable automatic recovery, as follows:

```
[sys-admin ~]#  cadmin --enable-auto-recovery
```

It is safe to leave automatic recovery enabled. However, when performing `discovery` operations, you may find it convenient to disable it.

Use the `cadmin --show-auto-recovery` command to show the current state. Use the `cadmin --disable-auto-recovery` command to disable it.

IP Addresses Reserved for Auto Recovery Mode

Four IP addresses are reserved on the head network for auto recovery operations. For clusters being installed for the first time, these tend to be low numbers as they are reserved before any service nodes or rack leader controllers (RLCs) are discovered. For systems being upgraded from previous SMC for SGI ICE X releases, the allocated IP addresses are allocated the first boot after the upgrade and would tend to have higher numbers.

DHCP Set Up for Auto Recovery Mode

When the auto recovery feature is enabled, the `dhcpd.conf` file is configured with DHCP addresses available to unknown systems. That is, when this mode is enabled, any system attached to the head network that is performing DHCP requests will get a generic pool address and then boot in to the auto recovery mode. When the auto recovery mode is disabled, DHCP is configured to not offer these special IP addresses.

Auto Recovery and the `discover` Command

The auto recovery mode conflicts with the way that the `discover` command operates by default. Therefore, the `discover` command automatically and temporarily disables auto recovery (if it was enabled) for the duration of the run of the `discover` command. For more information on the `discover` command, see "`discover` Command" on page 10.

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 ICE X Installation and Configuration Guide*.

In addition, you need to add or remove the RLC from the `/var/yp/ypservers` file on NIS Master service node. Remember to use the `-ib1` name for the RLC, as service nodes cannot resolve `r2lead` style names. For example, use `r2lead-ib1`.

```
# cd /var/yp && make
```

How To Avoid Out of Memory Occurrences on SLES11 When Using the PBS Professional Batch Scheduler

SGI ICE X is a diskless blade server typically configured with `nfs` root and a small (50 MB) swap space that is served via `iscsi`. A maximum of 64 blades boot from a rack leader controller (RLC). The RLC typically has SATA disks in a mirrored pair for blade filesystems and blade swap space. Some users turn off swap entirely because a full rack of blades swapping has proven to be stressful to the RLCs. When a Linux system has more memory requests than it can provide the kernel takes steps to defend the system using the out of memory (OOM) killer. The following section describes strategies for avoiding the loss of ICE blades due to OOM occurrences when the operating system is SLES11 and the batch scheduler is PBS Professional.

Some general guidelines are, as follows:

- Make sure that your application requests the proper amount of memory.
- After you ensure that your application asks for memory correctly, configure the `pbs_mom` process in PBS Professional to enforce memory limits. See your PBS Professional documentation for a complete description of the `pbs_mom` process.

  This only works well when the SGI `memacct` function is installed to properly compute the amount of memory used. This requires that Linux kernel jobs and Comprehensive System Accounting (CSA) are installed. For more information, see the *Linux Resource Administration Guide*. CSA does not have to be configured to log. Modify `/var/spool/PBS/mom_priv/config` file by adding `$enforce mem` to the file. As an example, an application that just allocates memory one megabyte at a time will be killed once it goes over the limit. Applications that allocate in bigger chunks can still get above the limit before PBS can kill the job.
- The PBS Pro enforce `mem` variable has no configuration options. To avoid OOM occurrences you need your own daemon, such as the `policykill` daemon.

  The `policykill` daemon looks for swapping in cpusets and works well in both large single-system image (SSI) with multiple cpusets and cluster (single cpuset). On large SSI, use of PBSPro's cpuset mom is required. On SGI ICE X systems use of SGI Altix bundle (example `PBSPro_10.1.0-SGIAltix_pp6_x86_64.tar.gz`) from Altair Engineering, Inc. is suggested. `policykill` has an `init` script, configuration file and daemon process itself. It requires customization for limits and notification methods.
- The Linux kernel Out Of Memory killer (`mm/oom_kill.c`) is responsible for keeping the system alive when memory has been exhausted. A snippet from the code is, as follows:

```
* The formula used is relatively simple and documented inline in the
* function. The main rationale is that we want to select a good task
* to kill when we run out of memory.
*
* Good in this context means that:
* 1) we lose the minimum amount of work done
* 2) we recover a large amount of memory
* 3) we don't kill anything innocent of eating tons of memory
* 4) we want to kill the minimum amount of processes (one)
* 5) we try to kill the process the user expects us to kill, this
*    algorithm has been meticulously tuned to meet the principle
*    of least surprise ... (be careful when you change it)
```

You can use `arrayd` to manage what processes gets killed. For more information on `arrayd`, see the `arrayd`(8) man page and the *Linux Resource Administration Guide*. `arrayd` has a configuration option to protect the daemon:

```
-oom oom_daemon,oom_child
Specify oom_adj ( OutOfMemory Adjustments ) respectively for the main
arrayd 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. On SMC for SGI ICE X systems, `cron` is configured off in `80-compute-distro-services` which is in

```
/var/lib/systemimager/images/<your compute image>/etc/opt/sgi/conf.d/80-compute-distro-services
```

by commenting out the following line:

```
initDisableServiceIfExists cron
```

To just enable `cron` on a blade is **not** a good practice. Files in

```
/var/lib/systemimager/images/<your compute image>/etc/cron*
```

must be reviewed for correctness in mixed writeable and read-only environment. For example, `sysstat`, `logrotate`, `suse.de-cron-local`, are the only services available in `/etc/cron*` directories. For a list of sample scripts, see Appendix A, "Out of Memory Adjustment" on page 171.

- Virtual memory `sysctl` tuning tries to balance use of system resources for user jobs and for system threads. The default setup is skewed towards user jobs but in the face of OOM system threads need more resources. For more information on `sysctl`, see the `sysctl`(8) man page. For an SMC for SGI ICE X system, the `sysctl` parameters might be predefined similar to the following:

```
# Give the kernel a bit more breathing room by requiring more free space
vm.min_free_kbytes = 131072
# Push dirty pages out faster
vm.dirty_expire_centisecs = 1000          # Default is 3000
vm.dirty_writeback_centisecs = 500        # Default (unchanged)
```

```
vm.dirty_ratio = 20                     # Default is 40
vm.dirty_background_ratio = 5           # Default is 10
```

If blades are run without swap, set the following variable:

```
vm.swappiness = 0
```

System Monitoring

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

Overview

SMC for SGI ICE X uses a Ganglia model for SGI ICE X system monitoring. Ganglia is a scalable, distributed monitoring system for high-performance computing systems, such as the SGI ICE X system. It displays web browser-based, real-time (on demand) histograms of system metrics. Figure 3-4 on page 152 shows an example display.

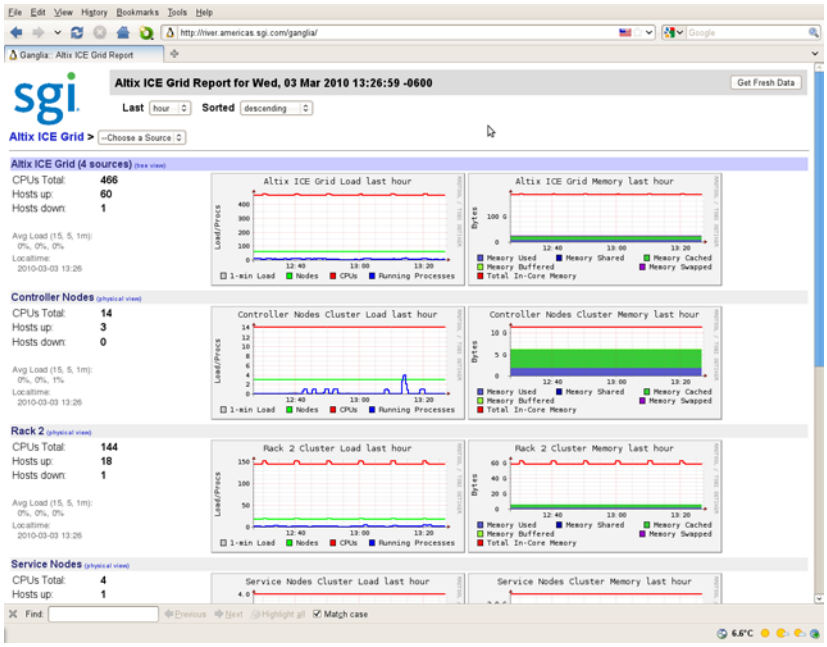

Figure 3-4 Ganglia System Monitor

Each 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 system admin controller (SAC). The RLC also sends its own statistics to the SAC. The SAC 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 3-5 on page 153 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.

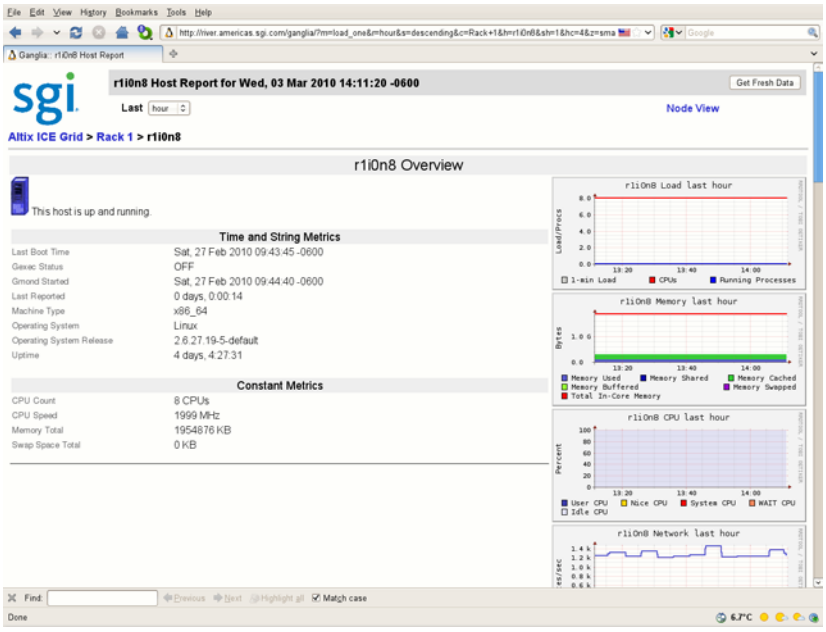

Figure 3-5 Ganglia System Monitoring Node View

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_pub_name*/ganglia

Monitoring System Metrics

By default, Ganglia monitors standard operating system metrics like CPU load, memory usage. The **Grid Report** view shows an overview of your system, such as the number of CPUs, the number of hosts (compute nodes) that are up or down, service node information, memory usage information, and so on.

The **Last** pull down menu allows you to view performance data on an hourly, daily, weekly, or yearly basis. The **Sorted** pull down menu allows provides an ascending, descending, or by host view of performance data. The **Grid** pull-down menu allows you to see performance data for a particular rack or service node. The **Get Fresh Data** button allows you to see current data performance.

SEL/Hardware Event Monitoring

The system admin controller (SAC), rack leader controllers (RLCs), the service nodes, the chassis management controllers (CMCs), and all the compute nodes (blades) are equipped with a specialized controller, called the Board Management Controller (BMC). This unit provides a broad set of functions as described in the IPMI 2.0 standard. SMC for SGI ICE X software uses the BMCs predominantly for remote power management, remote system configuration, and for gathering critical hardware events.

Currently, critical hardware events are gathered for the following nodes: RLCs, CMCs, and compute nodes (blades). These events are logged in the following locations:

- `/var/log/messages` via `syslog`
- `var/log/sel/sel.log`

All critical hardware events are summarized under the `BMC_CMC` event type. One particular event holds the following useful information:

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

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*event*>	`BMC_CMC`
<*app*>	SEL-LOGGER
<*date*>	date / time of the event
<*version*>	1.0
<*text*>	Exact copy of the hardware event description from the BMC

After reading the events from the BMCs, the BMC event logs are cleared on the controller to avoid duplicate events.

Node Availability Monitoring

The availability of each node in an SGI ICE X system is monitored by a lightweight daemon called `smchbc`. Each managed service node, rack leader controller (RLC), and compute node runs this daemon and reports its status to the server which monitors it. The server daemon, which runs on the system admin controller (SAC) and RLC, reports if the client is down after approximately 120 seconds. In this event, administrator-derived actions can be triggered, for instance sending an e-mail notification to the system administrator.

The HEARTBEAT event contains the following useful information:

`MSG ::=` <*syslog-prefix*> `SMC:`<*node*> `EVENT:HEARTBEAT APP:SMCHBD Date:`<*date*> `VERSION:1.0 TEXT` <*text*>

The HEARTBEAT event is created when nodes fail or recover, described by the `TEXT` field.

The following fields are all of the type string:

<*node*>	node name, for example, `r1i0n5`
<*date*>	date / time of the event
<*text*>	Description of event:
	`'Heartbeat not detected'`
	`'Heartbeat lost'`

Monitoring System Metrics with Performance Co-Pilot

A wealth of system metrics are also available through the Performance Co-Pilot (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The Performance Co-Pilot collection daemon (PMCD) runs on the system admin controller (SAC), rack leader controllers (RLCs), and managed service nodes. A performance metrics domain agent (PMDA) is running on the RLCs, which collects metrics from the compute nodes.

The new cluster metrics domain contains metrics that were previously available in other PMDAs. The method in which they are collected is different in a SMC for ICE X system, in order to minimize load on the compute nodes. The following metrics are available for each compute node in a system by querying the PMCD on their RLC:

```
admin:~ # pminfo -h r1lead cluster
cluster.control.suspend_monitoring
cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
```

```
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
cluster.network.ib.in.bytes
cluster.network.ib.in.errors.drop
cluster.network.ib.in.errors.filter
cluster.network.ib.in.errors.local
cluster.network.ib.in.errors.remote
cluster.network.ib.out.bytes
cluster.network.ib.out.errors.drop
cluster.network.ib.out.errors.filter
cluster.network.ib.total.errors.link
cluster.network.ib.total.errors.recover
cluster.network.ib.total.errors.integrity
cluster.network.ib.total.errors.vl15
cluster.network.ib.total.errors.overrun
cluster.network.ib.total.errors.symbol
```

Configuring Compute Blade Metrics

The list of metrics that are monitored by the compute node and are pushed to the PMCD on the rack leader controller (RLC) is configurable. In some cases , it may be even be desirable to disable metric collection entirely, as follows:

```
# cexec --head --all pmstore cluster.control.suspend_monitoring 1 pmstore \
-h r1lead cluster.control.suspend_monitoring 1
```

The default list of metrics that are collected by each compute node contains 41 metrics. There are dozens more available in the `cluster.* namespace`. The default list is stored on each RLC in the `/var/lib/pcp/pmdas/cluster/config` file. Changing this file will allow you to modify the default metric list with rack granularity. To change the list on a single node store a newline-delimited list of metrics to the node's instance of the `cluster.control.metrics` metric.

To see the current metric list for a compute node, perform the following:

```
# pmval -h r1lead -s 1 -i 'r1i1n0' cluster.control.metrics

metric:    cluster.control.metrics
host:      r1lead
semantics: discrete instantaneous value
units:     none
samples:   1

              r1i1n0
"cluster.kernel.percpu.cpu.user
cluster.kernel.percpu.cpu.nice
cluster.kernel.percpu.cpu.sys
cluster.kernel.percpu.cpu.idle
cluster.kernel.percpu.cpu.intr
cluster.kernel.percpu.cpu.wait.total
cluster.mem.util.free
cluster.mem.util.bufmem
cluster.mem.util.dirty
cluster.mem.util.writeback
cluster.mem.util.mapped
cluster.mem.util.slab
cluster.mem.util.cache_clean
cluster.mem.util.anonpages
cluster.infiniband.port.rate
cluster.infiniband.port.in.bytes
cluster.infiniband.port.in.packets
cluster.infiniband.port.in.errors.drop
cluster.infiniband.port.in.errors.filter
cluster.infiniband.port.in.errors.local
cluster.infiniband.port.in.errors.remote
cluster.infiniband.port.out.bytes
cluster.infiniband.port.out.packets
cluster.infiniband.port.out.errors.drop
```

```
cluster.infiniband.port.out.errors.filter
cluster.infiniband.port.total.bytes
cluster.infiniband.port.total.packets
cluster.infiniband.port.total.errors.drop
cluster.infiniband.port.total.errors.filter
cluster.infiniband.port.total.errors.link
cluster.infiniband.port.total.errors.recover
cluster.infiniband.port.total.errors.integrity
cluster.infiniband.port.total.errors.vl15
cluster.infiniband.port.total.errors.overrun
cluster.infiniband.port.total.errors.symbol
cluster.network.interface.in.bytes
cluster.network.interface.in.errors
cluster.network.interface.in.drops
cluster.network.interface.out.bytes
cluster.network.interface.out.errors
cluster.network.interface.out.drops
"
```

An example that changes the metric list to only include the CPU metrics for `r1i1n0` is, as follows:

```
# pmstore -h r1lead -i 'r1i1n0' cluster.control.metrics \
'cluster.kernel.percpu.cpu.user cluster.kernel.percpu.cpu.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 (see *Performance Co-Pilot Linux User's and Administrator's Guide*). The SDR provides temperature, voltage, and fan speed information for all service nodes, rack leader controllers (RLCs), compute nodes, and CMCs. This information is collected from service and compute nodes through their BMC interface, so it is out-of-band and does not impact the performance of the node.

The following metrics are available through the PMCD:

```
admin:~ # pminfo -h r1lead sensor
sensor.value.fan
sensor.value.voltage
sensor.value.temperature
```

Each sensor will have a separate instance within the domain, with the instance of the form:

```
<nodeName>:<nodeType>:<metricName>

nodeName ::= SMC for SGI ICE X node names (rXlead, rXiYc, rXiYnZ)
nodeType ::= "service", "cmc", "blade", "leader"
```

For example, to view voltages for the RLC, perform the following

```
admin:~ # pminfo -h r1lead -f sensor.value.voltage | grep -E '(^$|^sensor|r1lead)'

sensor.value.voltage
    inst [0 or "r1lead:leader:CPU1_Vcore"] value 1.3
    inst [1 or "r1lead:leader:CPU2_Vcore"] value 1.3
    inst [2 or "r1lead:leader:3.3V"] value 3.26
    inst [3 or "r1lead:leader:5V"] value 4.9
    inst [4 or "r1lead:leader:12V"] value 11.71
    inst [5 or "r1lead:leader:-12V"] value -12.3
    inst [6 or "r1lead:leader:1.5V"] value 1.47
    inst [7 or "r1lead:leader:5VSB"] value 4.9
    inst [8 or "r1lead:leader:VBAT"] value 3.31
```

For additional examples on how to retrieve values using `pmval`(1) and for using this data in trend analysis using `pmie`(1), see the appropriate man page and the *Performance Co-Pilot Linux User's and Administrator's Guide.*

Turning Off the `temperature.pmie` Feature

Currently, in `temperature.pmie` there are values that will `"Monitor: shut down components if temp too high"` . This feature is enabled by default as a safety mechanism. The procedure below describes how to turn it off.

Procedure 3-10 Turning Off the `temperature.pmie` Feature

To turn off the `temperature.pmie` feature, perform the following steps:

1. Edit the `/var/lib/pcp/config/pmie/control` file to comment out or remove the line that calls `/opt/sgi/lib/temperature.pmie`. For example,

```
#LOCALHOSTNAME n PCP_LOG_DIR/pmie/LOCALHOSTNAME/temperaturepmie.log -c /opt/sgi/lib/temperature.pmie
```

2. Run the **/etc/init.d/pmie restart** command. If you just want to adjust `temperature.pmie` values, see "Adjusting `temperature.pmie` Values" on page 160.

This has to be done on the system admin controller (SAC) and rack leader controller (RLC). In that case, it is recommended that you turn it off on the RLC images too.

Adjusting `temperature.pmie` Values

This section describes how to adjust `temperature.pmie` values.

Procedure 3-11 Adjusting `temperature.pmie` Values

You can adjust the warning or shutdown temperature values manually on the system admin controller (SAC) and on each one of the rack leader controllers (RLCs). If you adjust the values on the RLC, adjust the values on the RLC images, too. The settings will be preserved between reboots. To change the values, perform the following steps:

1. Edit the `/opt/sgi/lib/temperature.pmie` file:

```
admin_warning_temperature = 68; // degree Celsius
admin_shutdown_temperature = 73; // degree Celsius
leader_warning_temperature = 68; // degree Celsius
leader_shutdown_temperature = 73; // degree Celsius
service_warning_temperature = 68; // degree Celsius
service_shutdown_temperature = 73; // degree Celsius
cmc_warning_temperature = 48; // degree Celsius
cmc_shutdown_temperature = 53; // degree Celsius
cn_warning_temperature = 68; // degree Celsius
cn_shutdown_temperature = 73; // degree Celsius
sensor_temperature = "sensor.value.temperature"; // degree Celsius
```

2. Perform the following command to verify that you updated the script correctly, as follows:

```
# pmie -C /opt/sgi/lib/temperature.pmie
```

   If there are no errors, the `pmie -C` command returns with no message.

3. Run the **/etc/init.d/pmie restart** command or the **service pmie restart** command to restart the `pmie` service.

To turn off the `temperature.pmie` value, see "Turning Off the `temperature.pmie` Feature" on page 159.

Cluster Performance Monitor

You can use the Cluster Performance Monitor to monitor your SGI ICE X system. Log into the system admin controller (SAC) using the `ssh -X` command. Execute the pmice command and the **pmice - Cluster Performance Monitor** appears, as follows:

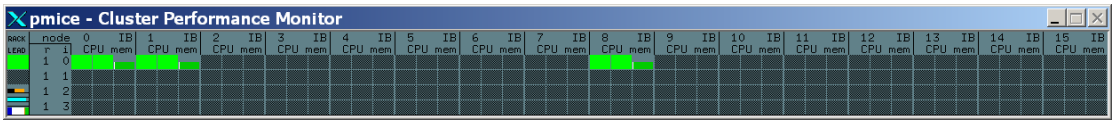

Figure 3-6 pmice- Cluster Performance Monitor

For a usage statement, use the `pmice --h` command, as follows:

```
admin:~ # pmice --h
/usr/bin/pmice: illegal option -- -
Info:
Usage: pmice [options] [pmgadgets options]

options:
  -K list  Show these CPUs. Comma-separated list
  -N list  Show these nodes. Comma-separated list
  -R list  Show these racks. Comma-separated list
  -V       Verbose/diagnostic output

pmgadgets(1) options:

  -C                    check configuration file and exit
  -h host               metrics source is PMCD on host
  -n pmnsfile           use an alternative PMNS
  -t interval           sample interval [default 2.0 seconds]
  -z                    set reporting timezone to local time of metrics source
  -Z timezone           set reporting timezone

  -zoom factor          make the gadgets bigger by a factor of 1, 2, 3 or 4
  -infofont fontname    use fontname for text in info dialogs
  -defaultfont fontname  use fontname for label gadgets

  -display display-string
```

```
-geometry geometry-string
-name name-string
-title title-string
-xrm resource
```

Troubleshooting

This section describes some troubleshooting tools and covers these topics:

dbdump Command

You can run the dbdump script to see an inventory of the SGI ICE X database.

The dbdump command is, as follows:

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

- Use the --admin argument to dump the system admin controller (SAC).
- Use the --leader argument to dump all rack leader controllers (RLCs).
- Use the --rack argument to dump a specific rack.
- Use the dbdump command without any argument to dump the entire SGI ICE X system.

EXAMPLES

Example 3-1 dbdump Command Examples

To dump the entire database, perform the following:

```
admin:~ # dbdump
0 is { cluster=oscar ifname=service0-bmc dev=bmc0 ip=172.24.0.3 net=head-bmc node=service0
```

```
  nodetype=oscar_service mac=00:30:48:8e:
1 is { cluster=oscar ifname=service0 dev=eth0 ip=172.23.0.3 net=head node=service0
  nodetype=oscar_service mac=00:30:48:33:53:2e }
2 is { cluster=oscar ifname=service0-ib0 dev=ib0 ip=10.148.0.2 net=ib-0 node=service0
  nodetype=oscar_service }
3 is { cluster=oscar ifname=service0-ib1 dev=ib1 ip=10.149.0.2 net=ib-1 node=service0
  nodetype=oscar_service }
4 is { cluster=oscar dev=eth0 ip=128.162.244.86 net=public node=oscar_server
  nodetype=oscar_server mac=00:30:48:34:2B:E0 }
...
```

Note: Some of the sample output in this section has been modified to fit the format of this manual.

To dump just the RLC, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --leader
0 is { cluster=rack1 ifname=r1lead-bmc dev=bmc0 ip=172.24.0.2 net=head-bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:8a:a4:c2 }
1 is { cluster=rack1 ifname=lead-bmc dev=eth0 ip=192.168.160.1 net=bmc node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
2 is { cluster=rack1 ifname=lead-eth dev=eth0 ip=192.168.159.1 net=gbe node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
3 is { cluster=rack1 ifname=r1lead dev=eth0 ip=172.23.0.2 net=head node=r1lead
  nodetype=oscar_leader mac=00:30:48:33:54:9e }
4 is { cluster=rack1 ifname=r1lead-ib0 dev=ib0 ip=10.148.0.1 net=ib-0 node=r1lead
  nodetype=oscar_leader }
5 is { cluster=rack1 ifname=r1lead-ib1 dev=ib1 ip=10.149.0.1 net=ib-1 node=r1lead
  nodetype=oscar_leader }
```

To dump just one rack, perform the following:

```
admin:~ # /opt/sgi/sbin/dbdump --rack 1
0 is { cluster=rack1 ifname=i0n0-bmc dev=bmc0 ip=192.168.160.10 net=bmc node=r1i0n0
  nodetype=oscar_clients mac=00:30:48:7a:a7:96 }
1 is { cluster=rack1 ifname=i0n0-eth dev=eth0 ip=192.168.159.10 net=gbe node=r1i0n0
  nodetype=oscar_clients mac=00:30:48:7a:a7:94 }
2 is { cluster=rack1 ifname=r1i0n0-ib0 dev=ib0 ip=10.148.0.3 net=ib-0 node=r1i0n0
  nodetype=oscar_clients }
3 is { cluster=rack1 ifname=r1i0n0-ib1 dev=ib1 ip=10.149.0.3 net=ib-1 node=r1i0n0
  nodetype=oscar_clients }
4 is { cluster=rack1 ifname=i0n1-bmc dev=bmc0 ip=192.168.160.11 net=bmc node=r1i0n1
```

```
  nodetype=oscar_clients mac=00:30:48:7a:a7:86 slot=1 }
5 is { cluster=rack1 ifname=i0n1-eth dev=eth0 ip=192.168.159.11 net=gbe node=r1i0n1
  nodetype=oscar_clients mac=00:30:48:7a:a7:84 slot=1 }
6 is { cluster=rack1 ifname=r1i0n1-ib0 dev=ib0 ip=10.148.0.4 net=ib-0 node=r1i0n1
  nodetype=oscar_clients slot=1 }
7 is { cluster=rack1 ifname=r1i0n1-ib1 dev=ib1 ip=10.149.0.4 net=ib-1 node=r1i0n1
  nodetype=oscar_clients slot=1 }
8 is { cluster=rack1 ifname=i0n10-bmc dev=bmc0 ip=192.168.160.20 net=bmc node=r1i0n10
  nodetype=oscar_clients slot=10 }
9 is { cluster=rack1 ifname=i0n10-eth dev=eth0 ip=192.168.159.20 net=gbe node=r1i0n10
  nodetype=oscar_clients slot=10 }
10 is { cluster=rack1 ifname=r1i0n10-ib0 dev=ib0 ip=10.148.0.13 net=ib-0 node=r1i0n10
  nodetype=oscar_clients slot=10 }
...
```

`smc-info-gather` Command

The `smc-info-gather` command enables to collect vital system data especially when troubleshooting problems. The `smc-info-gather` command collects the information about the following:

- Digital media `dminfo` files, system logs, Dynamic Host Configuration Protocol (DHCP), network file system (NFS)
- MySQL cluster database dump
- Network service configuration files, for example, C3, Ganglia, DHCP, domain name service (DNS) configuration files
- A list of installed system images
- Log files in `/var/log/messages`
- Chassis management control (CMC) slot table for each rack
- basic input-output system (BIOS), Baseboard Management Controller (BMC), CMC and InfiniBand fabric software versions from all SGI ICE X nodes

To see a usage statement for the `smc-info-gather` command, perform the following:

```
admin:/opt/sgi/sbin # smc-info-gather  -h
 usage: smc-info-gather [-h] [-P path] [-o file]
```

```
        smc-info-gather -h            # Print this usage page
        smc-info-gather -o file       # Tar and gzip the directories
into file (imply -n)
        smc-info-gather -p path       # Directory to write the data
(default /var/tmp/smc)
```

`cminfo` Command

The `cminfo` command is used internally by many of the SMC for SGI ICE X scripts that are used to discover, configure, and manage an SGI ICE X system.

In a troubleshooting situation, you can use it to gather information about your system. To see a usage statement from a rack leader controller (RLC), perform the following:

```
r1lead:~ # cminfo --help
Usage: cminfo [--bmc_base_ip|--bmc_ifname|--bmc_iftype|--bmc_ip|--bmc_mac|--bmc_netmask|--bmc_nic|
--dns_domain|--gbe_base_i
p|--gbe_ifname|--gbe_iftype|--gbe_ip|--gbe_mac|--gbe_netmask|--gbe_nic|--head_base_ip|
--head_bmc_base_ip|--head_bmc_ifname|
--head_bmc_iftype|--head_bmc_ip|--head_bmc_mac|--head_bmc_netmask|--head_bmc_nic|--head_ifname|
--head_iftype|--head_ip|--he
ad_mac|--head_netmask|--head_nic|--ib_0_base_ip|--ib_0_ifname|--ib_0_iftype|--ib_0_ip|--ib_0_mac|
--ib_0_netmask|--ib_0_nic|
--ib_1_base_ip|--ib_1_ifname|--ib_1_iftype|--ib_1_ip|--ib_1_mac|--ib_1_netmask|
--ib_1_nic|--name|--rack]
r1lead:~ # cminfo --bmc_base_ip
```

EXAMPLES

Example 3-2 `cminfo` Command Examples

To see the RLC's BMC IP address, perform the following:

```
r1lead:~ # cminfo --bmc_base_ip
192.168.160.0
```

To see the RLC's DNS domain, perform the following:

```
r1lead:~ # cminfo --dns_domain
ice.domain_name.mycompany.com
```

To see the BMC NIC, perform the following:

```
r1lead:~ #  cminfo --bmc_nic
eth0
```

To see the IP address of the ib1 InfiniBand fabric, perform the following:

```
r1lead:~ # cminfo --ib_1_base_ip
10.149.0.0
```

kdump Utility

The `kdump` utility is a `kexec`-based crash dumping mechanism for the Linux operating system. You can downlonad `debuginfo` kernel RPMs for use with crash and any kernel dumps at the following location:http://support.novell.com/linux/psdb/byproduct.html.

To get a traceback or system dump, perform the following from the system console:

```
console r1i0n0
^e c l 1 8
^e c l 1 t       #traceback
^e c l 1 c       #dump
```

Note: This example shows the letter "`c`", a lowercase L "`l`", and the number one "`1`" in all three lines.

On the system admin controller (SAC), go to `/net/r1lead/var/log/consoles` for the traceback and `/net/r1lead/var/log/dumps/r1i0n0` for the system dump.

You can dump a compute node, the rack leader controller (RLC) (`r1lead`), or a service node, such as, `service0`.

System Firmware

Note: Your SGI ICE X system comes preinstalled with the appropriate firmware. See your SGI field support person for any BMC, BIOS, and CMC firmware updates.

The SGI ICE X system firmware software consists of the following components:

`sgi-ice-blade-bmc-1.43.5-1.x86_64.rpm`

Blade BMC firmware and update tool

`sgi-ice-blade-bios-2007.08.10-1.x86_64.rpm`

Blade BIOS image and update tool

`sgi-ice-cmc-0.0.11-2.x86_64.rpm`

CMC firmware and update tool

BIOS Version Interrogation

To identify the BIOS you need both the version and the release date. You can get these using the `dmidecode` command. Log onto the node on which you want to interrogate BIOS level and perform the following:

```
# dmidecode -s bios-version; dmidecode -s bios-release-date
```

BMC Revision Interrogation

The BMC firmware revision can be retrieved using the `ipmiwrapper`. For example, from the system admin controller (SAC), the following command gets the BMC firmware revision for `r1i0n0`:

```
# ipmiwrapper r1i0n0 bmc info | grep 'Firmware Revision'
```

CMC Version Interrogation

The CMC firmware version can be retrieved using the `version` command to the CMC. For example, if you are logged onto the `r1lead` rack leader controller (RLC), the following command gets the CMC firmware version:

```
# ssh root@r1i0-cmc version
```

InfiniBand Version Interrogation

The `ibstat` command retrieves information for the InfiniBand links including the firmware version. The following command gets the InfiniBand firmware version:

```
# ibstat | grep Firmware
```

Getting Firmware Information for All System Nodes

The `firmware_revs` script on the system admin controller (SAC) collects the firmware information for all nodes in the SGI ICE X system, as follows:

```
admin:~ # firmware_revs
BIOS versions:
--------------
admin: 6.00
r1lead: 6.00
service0: 6.00
r1i0n0: 6.00
r1i0n1: 6.00
r1i0n8: 6.00
r1i1n0: 6.00
r1i1n1: 6.00
r1i1n8: 6.00


BIOS release dates:
-------------------
admin: 05/10/2007
r1lead: 05/10/2007
service0: 05/10/2007
r1i0n0: 05/29/2007
r1i0n1: 05/29/2007
r1i0n8: 05/29/2007
r1i1n0: 05/29/2007
r1i1n1: 05/29/2007
r1i1n8: 05/29/2007


BMC versions:
-------------
```

```
admin: 1.31
r1lead: 1.31
service0: 1.31
r1i0n0: 1.29
r1i0n1: 1.29
r1i0n8: 1.29
r1i1n0: 1.29
r1i1n1: 1.29
r1i1n8: 1.29


CMC versions:
-------------
r1i0c: 0.0.9pre10
r1i1c: 0.0.9pre10


Infiniband versions:
--------------------
r1lead: 4.7.600
service0: 4.7.600
r1i0n0: 1.2.0
r1i0n0: 1.2.0
r1i0n1: 1.2.0
r1i0n1: 1.2.0
r1i0n8: 1.2.0
r1i0n8: 1.2.0
r1i1n0: 1.2.0
r1i1n0: 1.2.0
r1i1n1: 1.2.0
r1i1n1: 1.2.0
r1i1n8: 1.2.0
r1i1n8: 1.2.0
```

Out of Memory Adjustment

This section describes sample set of out of memory OOM adjust scripts for `cron` and PBS `prologue` and `epilogue`.

Example A-1 `oom_adj.user.pl.txt`: OOM Adjustment Script

```
#!/usr/bin/perl
use strict;
use Sys::Hostname;
my $host = hostname();
my $DEBUG=0; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];

sub ResetOomAdj {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $CMD_LINE;
my $RETURN;
$AVOID_UIDS="root|100|nobody|ntp|USER|daemon|postfix|vtunesag";
   open (PS_CMD, "-|") || exec 'ps -e -o user,pid';
   while (<PS_CMD>) {
     chomp;
     ($_userid, $tpid) = split (/\s+/, $_);

       if ( $_userid !~ m/^${AVOID_UIDS}/ && $tpid =~ /^[0-9]/ && -e
"/proc/$tpid/oom_adj" ) {
          print "$CALL_SCPT $host: Found processes to set to zero
oom_adj...\n" if $DEBUG;
          $CMD_LINE="echo 0 > /proc/$tpid/oom_adj";
          $RETURN=`$CMD_LINE`;
       }
       elsif ( $tpid =~ /^[0-9]/ && -e "/proc/$tpid/oom_adj" ) {
          print "$CALL_SCPT $host: Found processes to set to protect
oom_adj...\n" if $DEBUG;
          $CMD_LINE="echo -17 > /proc/$tpid/oom_adj";
          $RETURN=`$CMD_LINE`;
       }
   }
```

```
   close PS_CMD;

}

&ResetOomAdj();
```

Example A-2 `cronentry`: Sample `cron` Entry for `oom_adj` Script

```
-*/2 *   * * *     /root/oom_adj.user.pl
```

Example A-3 `prologue`: Sample `prologue` Script

```
#!/bin/bash
###############################################################################
#
#  Version: 2.3.1 : Updated 8/12/09
#  Date: Oct 16, 2007
#  Author: Scott Shaw, sshaw@sgi.com
#
#  Script Name: PBS Pro Prologue Script
#  The purpose of the Prologue script is to terminate leftover user processes and
#  allocated IPCs resources. The prologue script consists of two scripts, the main
#  prologue script and a chk_node.pl script.  To minimize accessing each node the
#  prologue script executes a parallel ssh shell across a set of nodes based on the
#  PBS_NODEFILE.  For large clusters over 64 nodes serial ssh access is slow so having
#  a flexible parallel ssh to help speed up the clean-up process of each node.  In
#  some cases, a PBS jobs can normally terminate but some MPI implementations do not
#  normally terminate the MPI processes due to crappy error code handling or
#  segmentation faults within the MPI application thus leaving behind user processes
#  still consuming system resources.
#
#  When the prologue script is launched by PBS MOM the ssh session is executed and will
#  execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
#  commands which are executed on each node based on the PBS_NODEFILE.
#
#  Execution of the prologue script is based on the root account.
#
#  This script needs to reside on each execution host/node
#  Location: /var/spool/PBS/mom_priv
```

```
#  File name: prologue
#  Permissions: 755
#  Owner: root
#  Group: root
#
#  ls output: ls -l /var/spool/PBS/mom_priv/prologue
#       -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/prologue
#
#  Modification of the prologalarm maybe necessay if the network access is slow to
#  each node.  30 seconds may not be enough time to check 256 nodes in a cluster.
#  prologalarm #  Defines the maximum number of seconds the prologue
#  and prologue may run before timing out. Default:
#  30. Integer. Example:
#  $prologalarm 30
#
#################################################################################


JOBID=$1
USERNAME=$2
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

spawn ()
{
        if [[ ‘jobs | grep  -v Done | wc -l’ -ge $1 ]]; then
                wait
        fi
        shift
```

```
        $@  &
}

exec_cmd ()
{
   for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
   do
      spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
   done
   wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
   QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
   QSTAT=/opt/pbs/default/bin/qstat

else
  echo "Epilogue Error: The qstat command could not be detected, exiting..."
  exit 1
fi


prefix_flag=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $4}’ | awk -F. ’{print $1}’‘
queue=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $3}’‘

   echo "Start Prologue ${VERSION} ‘date‘ "

   if [ $( /bin/uname -m ) = "x86_64" ]; then
         echo "Prefix passed: ${prefix_flag}"
         echo "destination queue: ${queue}"

         case $prefix_flag in
            TB)
              # Enable turbo and do node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} TB"
              exec_cmd
              ;;
```

```
            BP)
              # Bypass the turbo setting and P/Elog cleanup
              echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
              ;;
            JT)
              # Enable turbo but do not run the node cleanup p/elog scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} JT"
              exec_cmd
              ;;
            NT)
              # bypass turbo settings but run the node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue} NT"
              exec_cmd
              ;;
             *)
              # disable turbo and run the node cleanup scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Plog ${queue}"
              exec_cmd
          esac
   else
      echo "The prologue script is intended to run on x86_64 nodes not ‘uname -m‘."
      echo "End Prologue ${VERSION} ‘date‘ "
      exit -1
   fi
   echo "End Prologue ${VERSION} ‘date‘ "


#Output the cluster details file
   if [ -f /var/spool/PBS/mom_priv/cluster_info.out ]; then
      cat /var/spool/PBS/mom_priv/cluster_info.out
   else
      echo "WARNING: The cluster info file does not exist. Contact hpc_support and report this warning."
   fi
```

Example A-4 epilogue: Sample epilogue Script

```
#!/bin/bash
##############################################################################
#
#  Version: 2.3.1 : Updated 8/12/09
#  Date: Oct 16, 2007
#  Author: Scott Shaw, sshaw@sgi.com
```

```
#
#  Script Name: PBS Pro Epilogue Script
#  The purpose of the epilogue script is to terminate leftover user processes and
#  allocated IPCs resources. The epilogue script consists of two scripts, the main
#  epilogue script and a chk_node.pl script.  To minimize accessing each node the
#  epilogue script executes a parallel ssh shell across a set of nodes based on the
#  PBS_NODEFILE.  For large clusters over 64 nodes serial ssh access is slow so having
#  a flexible parallel ssh to help speed up the clean-up process of each node.  In
#  some cases, a PBS jobs can normally terminate but some MPI implementations do not
#  normally terminate the MPI processes due to crappy error code handling or
#  segmentation faults within the MPI application thus leaving behind user processes
#  still consuming system resources.
#
#  When the epilogue script is launched by PBS MOM the ssh session is executed and will
#  execute the chk_node.pl script. The chk_node.pl script contains a series of clean-up
#  commands which are executed on each node based on the PBS_NODEFILE.
#
#  Execution of the epilouge script is based on the root account.
#
#  This script needs to reside on each execution host/node
#  Location: /var/spool/PBS/mom_priv
#  File name: epilogue
#  Permissions: 755
#  Owner: root
#  Group: root
#
#  ls output: ls -l /var/spool/PBS/mom_priv/epilogue
#       -rwxr-xr-x 1 root root 2054 Sep  6 19:39 /var/spool/PBS/mom_priv/epilogue
#
#  Modification of the prologalarm maybe necessay if the network access is slow to
#  each node.  30 seconds may not be enough time to check 256 nodes in a cluster.
#  prologalarm #  Defines the maximum number of seconds the prologue
#  and epilogue may run before timing out. Default:
#  30. Integer. Example:
#  $prologalarm 30
#
##############################################################################


JOBID=$1
USERNAME=$2
```

```
GROUPNAME=$3
JOBNAME=$4
P_PID=$5
NPCUS=$6
CPU_PERCENT=$7
QUEUE=$8
TTY_TYPE=$9
UNKNOWN_ARG=$10
VERSION="v2.3.1"

SSHOPTS="-o StrictHostKeyChecking=no -o ConnectTimeout=6"

# If the cluster blade layout is not in sequentially than use a flat file.
NODES_FILE="/var/spool/PBS/aux/${JOBID}";

spawn ()
{
        if [[ ‘jobs | grep  -v Done | wc -l‘ -ge $1 ]]; then
                wait
        fi
        shift
        $@  &
}

exec_cmd ()
{
   for HOSTNAME in $( cat ${NODES_FILE} | sort -u )
   do
      spawn 25 ssh ${SSHOPTS} ${HOSTNAME} $CMDLINE
   done
   wait
}

# main()
#Find PBS qstat command
if [ -f /usr/pbs/bin/qstat ]; then
   QSTAT=/usr/pbs/bin/qstat

elif [ -f /opt/pbs/default/bin/qstat ]; then
   QSTAT=/opt/pbs/default/bin/qstat
```

```
else
  echo "Epilogue Error: The qstat command could not be detected, exiting..."
  exit 1
fi



prefix_flag=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $4}’ | awk -F. ’{print $1}’‘
queue=‘${QSTAT} -a ${JOBID} | grep "^[0-9]" |awk ’{print $3}’‘

   echo "Start Epilogue ${VERSION} ‘date‘ "
   if [ $( /bin/uname -m ) = "x86_64" ]; then
         echo "Prefix passed: ${prefix_flag}"
         echo "destination queue: ${queue}"

         case $prefix_flag in
            TB)
              # Enable turbo and do node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
              exec_cmd
              ;;
            BP)
              # Bypass the turbo setting and P/Elog cleanup
              echo "* * * * Bypassing the PBS Prologue and Epilogue scripts * * * *"
              ;;
            JT)
              # Enable turbo but do not run the node cleanup p/elog scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset JT"
              exec_cmd
              ;;
            NT)
              # bypass turbo settings but run the node cleanup
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog noreset NT"
              exec_cmd
              ;;
             *)
              # disable turbo and run the node cleanup scripts
              CMDLINE="/var/spool/PBS/mom_priv/chk_node.pl Elog reset"
              exec_cmd
          esac
```

```
else
   echo "The epilogue script is intended to run on x86_64 nodes not ‘uname -m‘."
   echo "End Epilogue ${VERSION} ‘date‘ "

   exit -1
fi
   echo "End Epilogue ${VERSION} ‘date‘ "
```

Example A-5 chk_node.pl.txt: Script epilogue and prologue Use.

```
#!/usr/bin/perl
#  Version: 2.3.1 : Updated 8/12/09
#  Orig Date: Oct 10, 2007
#  Author: Scott Shaw, sshaw@sgi.com
#
#  This perl script is called by PBS Pro prologue and epilogue scripts when
#  a user submits a job through PBS Pro.  The purpose of this script is to
#  sanitize a range of nodes identified by the $PBS_NODEFILE list by
#  terminating old user processes, old ipc allocations, temp files,
#  and to flush the system buffer cache.
#
#  Changes:
#   2/1/08 sshaw@sgi.com
#          - Added a subroutine to clean-up /tmp directory
#          - changed system() to exec since it was corrupting memory
#          - declared all vars to be local to subroutine, before it was loosely defined
#          - added strict checking of perl script
#   3/24/08 sshaw@sgi.com
#          - fixed debug conditional
#          - cleaned up the CleanUpProcesses procedure and added which processes
#            and user being terminated.
#          - Changed the killall to pkill due to userid > 8 chars
#   11/13/08 sshaw@sgi.com
#          - added a subroutine to clean-up /dev/shm since several users
#            use this location for temporary scratch space.
#   03/31/09 sshaw@sgi.com
#          - added subroutines to enable/disable Turbo mode on Intel series 5500 CPUs
#   04/22/09 sshaw@sgi.com
#          - added subroutines to speed step the core processor frequency to a lower freq
#   08/12/09 sshaw@sgi.com
#          - fixed minor issues with setting the frequency and fixed cpu freq to max speed
```

```
use strict;

use Sys::Hostname;
my $host = hostname();
my $DEBUG=1; # 0=turn off, 1=turn on
my $CALL_SCPT=$ARGV[0];
my $queue_destination=$ARGV[1];
my $prefix_option=$ARGV[2];
my $set_freq=0;

#####
# The following lines are added for Turbo/SMT mode starting with Intel 5500 series CPUs
my $rdmsr = "/var/spool/PBS/mom_priv/rdmsr";
my $wrmsr = "/var/spool/PBS/mom_priv/wrmsr";
my $msr = "0x199";
my $tbit = 1 <<  32;


# Several MPI implementations or MPI applications use IPC shared memory.  When
# a MPI application abnormally terminates it leaves behind allocated resources.
# this subroutine will remove any IPC resources allocated for the user’s job.
sub CleanUpIPC_Table {
my $tkey;
my $tshmid;
my $towner;
my $tperms;
my $tbytes;
my $tnattch;
my $tstatus;
my $CMD_LINE;
my $RETURN;

open(IPC_SHARMEM, "-|") || exec ’ipcs -m’;
   while () {
   chomp;
   ($tkey, $tshmid, $towner, $tperms, $tbytes, $tnattch, $tstatus) = split (/\s+/, $_);
        if ( $tkey =~ /^[0-9]/ ) {
           if ( $towner !~ m/root|^ / ) {
           print "$CALL_SCPT $host: Found IPC_SHR_MEM allocation: $tshmid $towner, terminating...\n" if $DEBUG;
            $CMD_LINE="ipcrm -m $tshmid";
```

```
            $RETURN='$CMD_LINE';
           }
         }
   }
   close IPC_SHARMEM;
}

# This subroutine will parse the process list and terminate any user processes or logins
# into the node(s)
sub CleanUpProcesses {
my $AVOID_UIDS;
my $_userid;
my $tpid;
my $tppid;
my $tcpu;
my $tstime;
my $ptty;
my $ttime;
my $tcmd;
my @TERM_USER;
my @TEMP;
my $USER;
my $CMD_LINE;
my $RETURN;

$AVOID_UIDS="root|100|101|nobody|bin|ntp|UID|daemon|postfix|vtunesag";
   open (PS_CMD, "-|") || exec 'ps -ef';
   while () {
     chomp;
     ($_userid, $tpid, $tppid, $tcpu, $tstime, $ptty, $ttime, $tcmd) = split (/\s+/, $_);

       if ( $_userid !~ m/^${AVOID_UIDS}/ ) {
          if ( $_userid =~ /^[0-9]/ ) {
             $_userid='ypcat passwd | egrep ${_userid} | cut -d \":\" -f 1';
             chomp $_userid;
          }
          print "$CALL_SCPT $host: Found leftover processes $tcmd from $_userid terminating...\n" if $DEBUG;
          $CMD_LINE="pkill -9 -u $_userid";   # Switched to pkill due to length of usernames.
          $RETURN='$CMD_LINE';
       }
   }
```

```
   close PS_CMD;
   system("/root/oom_adj.user.pl");
}
# This subroutine will remove any temporary files created by MPI application under /tmp.
sub CleanUpTmp {
my $_filename;
my @TEMP;
my @TERM_FILE;
my $CMD_LINE;
my $RETURN;
my $_nofiles;
my $FILE;


   open (LS_CMD, "-|") || exec 'ls /tmp';
   while () {
      chomp;
      ($_filename) = split (/\s+/, $_);
      if ( $_filename =~ m/^mpd/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
      }
      elsif ( $_filename =~ m/^ib_pool/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
         }
      elsif ( $_filename =~ m/^ib_shmem/ ) {
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
         }
   }
   close LS_CMD;

   foreach $FILE (@TERM_FILE) {
      $CMD_LINE="rm -f /tmp/${FILE}";
      $RETURN='$CMD_LINE';
   }

   $_nofiles = scalar @TERM_FILE;
      if ($_nofiles ne 0) {
          print "$CALL_SCPT $host: Found $_nofiles MPI temp files under /tmp. Removing...\n" if $DEBUG;
```

```
      }
}

# Flush the Linux IO buffer cache and the slab cache using the  bcfree command.
sub FreeBufferCache {
my $CMD_LINE;
my $RETURN;
my $BCFREE;
my $BCFREE_OPTS;

$BCFREE="/usr/bin/bcfree";
$BCFREE_OPTS="-a -s";

   if (-e "${BCFREE}") {
      $CMD_LINE="${BCFREE} ${BCFREE_OPTS}";
      $RETURN=‘$CMD_LINE‘;
   }
}

# This subroutine will remove any temporary files created by MPI application under /dev/shm.
sub CleanUpshm {
my $_filename;
my @TEMP;
my @TERM_FILE;
my $CMD_LINE;
my $RETURN;
my $_nofiles;
my $FILE;


   open (LS_CMD, "-|") || exec ’ls /dev/shm’;
   while () {
      chomp;
      ($_filename) = split (/\s+/, $_);
         @TEMP=$_filename;
         push @TERM_FILE, $TEMP[0];
   }
   close LS_CMD;

   foreach $FILE (@TERM_FILE) {
      if (${FILE} !~ m/sysconfig/) {
```

```
         $CMD_LINE="rm -rf /dev/shm/${FILE}";
         $RETURN=‘$CMD_LINE‘;
         print "${RETURN}" if $DEBUG;
         print "$CALL_SCPT $host: Found ${FILE} dir/file under /dev/shm. Removing it...\n" if $DEBUG;
      }
   }

}

sub chk_msr_state {
# Hyperthreading Assumption, if the first core has the bit set to enable/disable
# then it is assumed all other cores within the node have the same setting.

my $msr_lsmod=‘lsmod | grep -c msr‘;    # 0=not loaded, 1=msr loaded

   if ( $msr_lsmod == 0 ) {
      print "Loading MSR Kernel Modules...\n";
      ‘modprobe msr‘;  # we need the msr kernel modules loaded to read the msr values
       sleep(1);  # give time for the msr modules to load
   }
}

sub enable_turbo_mode {
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
my $i;
my $val;
my $nval;
        chk_msr_state();
        print "${host}: Enabling turbo mode...\n";
        chomp($val = ‘$rdmsr -p 0  $msr‘);
        $val = hex("100000017");
        $nval = $val ^ $tbit;
        printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
        for ($i = 0; $i < $ncpus; $i++) {
           ‘$wrmsr -p $i $msr $nval‘;
        }
        load_system_services();
}

sub disable_turbo_mode {
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
```

```
my $i;
my $val;
my $nval;
        chk_msr_state();
        print "${host}: Disabling turbo mode...\n";
        chomp($val = `$rdmsr -p 0  $msr`);
        $val = hex(16);
        #$val = hex($val);
        $nval = $val ^ $tbit;
        printf("${host}: Changing msr $msr on all cores from 0x%lx to 0x%lx\n", $val, $nval);
        for ($i = 0; $i < $ncpus; $i++) {
           `$wrmsr -p $i $msr $nval`;
        }
}


sub load_system_services {
my $powersave_loaded=`ps -ef | grep -v grep | grep -c power`;

   if ($powersave_loaded == 0 ) {
      print "${host}: Loading system services...\n";
      system("(/etc/init.d/acpid start;/etc/init.d/powersaved start)&> /dev/null");
      sleep(1);
      system("/usr/bin/powersave -f");
   }
   else {
      print "Powersaved already loaded.\n";
   }
}

sub unload_system_services {
   print "${host}: Unloading system services...\n";
   system("(/etc/init.d/acpid stop;/etc/init.d/powersaved stop)&> /dev/null");
}

sub run_cleanup {
    &CleanUpshm();
    &CleanUpTmp();
    &CleanUpIPC_Table();
    &CleanUpProcesses();
    &CleanUpProcesses();
```

```
}

sub set_processor_speed {
my $freq=shift;
my $ncpus = ‘cat /proc/cpuinfo | grep processor | wc -l‘;
my $i;
my $file;
     load_system_services();
     $freq = $freq * 1000;
     printf("${host}: Setting Proc Core speed to: %.3f GHz\n",($freq/1000000)) ;
     for ($i = 0; $i < $ncpus; $i++) {
       $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_min_freq";
       open FILE1, ">", $file or die $!;
          print FILE1 "$freq\n";
       close FILE1;

       $file = "/sys/devices/system/cpu/cpu" . $i . "/cpufreq/scaling_max_freq";
       open FILE2, ">", $file or die $!;
          print FILE2 "$freq\n";
       close FILE2;
     }
}


#
#print "$prefix_option\n";
#print "$queue_destination\n";
#
#   if ( $queue_destination =~ /^f/ ) {
#      my $b=0;
#      ($a,$set_freq) = split (/f/, $queue_destination);
#      set_processor_speed($set_freq);
#   }


# Don’t run on systems with earlier than Nehalem processors
# Based on the prefix_option set turbo mode accordingly and run node cleanup routines.
   #if( $prefix_option =~ m/TB/ ){
        #enable_turbo_mode();
        #run_cleanup();
   #}
```

```
   #elsif ( $prefix_option =~ m/JT/ ) {
      #print "* * * * ENABLE TURBO and bypass PBS Prologue and Epilogue scripts * * * *\n";
      #enable_turbo_mode();
   #}
   #elsif ( $prefix_option =~ m/NT/ ) {
      #print "* * * * Bypassing the Turbo checks and run just node clean-up * * * *\n";
      #run_cleanup();
   #}
   #elsif ( $queue_destination =~ /^f/ ) {
      #my $b=0;
      #($a,$set_freq) = split (/f/, $queue_destination);
      #set_processor_speed($set_freq);
   #}
   #elsif ( $queue_destination =~ /^reset/ ) {
         #set_processor_speed(2934);
         #disable_turbo_mode();
         #unload_system_services();
         #run_cleanup();
     #}
   #else {
      #disable_turbo_mode();
      #unload_system_services();
      #run_cleanup();
   #}

run_cleanup();
```

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