SGI® Foundation Software (SFS) User Guide

007–6410–001

New Features

This new manual supports the SGI Foundation Software 2.12 release.

Record of Revision

Version	Description
001	May 2015 Original publication.

Contents

About This Guide

This publication provides information about the SGI Foundation Software (SFS) tools, commands, and utilities. The SFS software enables you, the system administrator, to tune and monitor your SGI computer system. In addition, these tools facilitate efficient communication with SGI technical support staff members.

Related Publications

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

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

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

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

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

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

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

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

http://docs.sgi.com

The following software documentation might be useful to you:

- *SGI Management Center (SMC) Administration Guide for Clusters*, publication 007–6358–*xxx*
- *SGI Management Center (SMC) Installation and Configuration Guide for Clusters*, publication 007–6359–*xxx*
- *SGI UV System Software Installation and Configuration Guide*, publication 007–5948–*xxx*

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

Procedure 0-1 To retrieve hardware documentation

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

   `docs.sgi.com`

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

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

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

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

   - Check **Search only titles**.
   - Check **Show only 1 hit/book**.
   - Click **search**.

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at the following website:

  http://docs.sgi.com

Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.

- You can view man pages by typing `man` *title* on a command line.

Conventions

The following conventions are used throughout this document:

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

Reader Comments

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

You can contact SGI in either of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

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

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

SGI values your comments and will respond to them promptly.

Overview

This chapter contains the following topics:

About SFS

SGI Foundation Software (SFS) includes performance utilities, optimization tools, and technical support tools. Designed for high-performance computing, these tools help to maximize SGI system performance and availability. Many SFS tools, including the preventive and predictive analysis tools, operate automatically, without human intervention. For example, SFS logging and fault analysis tools interpret a system's historical data and make changes to improve system boot performance automatically. Other tools move pages from a failed memory unit to a working memory unit seamlessly, without downtime or human interaction.

While many SFS utilities and tools work in the background to optimize program performance, other tools require configuration information from you, the system administrator. The SFS components that this guide addresses are as follows:

- Components that you can tune to enhance reliability, serviceability, and serviceability. These includes the `memlog`(8) utility, the CPU frequency configuration commands, the hardware event tracker (HET), and Remote Services.
- Other SFS components that require less user tuning. These components include `sgi_irqbalance`(8) and the `sgtools` package.

Installing SFS

SGI requires SFS to be installed on all computing systems, and SGI installs SFS on all computing systems before the hardware leaves the factory.

If you need to reinstall software on your SGI hardware, use the installation instructions that pertain to your computing environment, as follows:

- On SGI UV systems, first install the operating system on the server, and then install SFS. The following manual includes instructions that explain how to install the operating system and SFS on a standalone SGI UV server:

  SGI UV System Software Installation and Configuration Guide

- On cluster systems, such as SGI Rackable clusters and SGI ICE clusters, the installer automatically includes the SFS software in the software image for each node type. For example, if your cluster includes SGI ICE X hardware, the cluster installer includes SFS in the rack leader controller images and the compute node images.

  The following manual includes information about how to install software on SGI cluster systems:

  SGI Management Center (SMC) Installation and Configuration Guide for Clusters

Configuring Hardware Event Tracker (HET) Notifications

The following topics contain information about HET:

About HET

All of your SGI system's baseboard management controllers (BMCs) send SNMP traps to the management node. The HET tools in the SGI Foundation Software (SFS) process these system alerts and send an email notification after critical hardware events occur. Depending on your equipment, the management node is one of the following:

- The admin node of a cluster system. Cluster systems include SGI ICE clusters, SGI Rackable clusters, and clusters that includes a mix of these systems.
- The system management node (SMN) of an SGI UV 2000, SGI UV 1000, or SGI UV 100 server.

The HET tools are configured by default. You do not need to perform any additional system configuration, but SGI recommends that you customize the email address to which the HET tools send critical event notifications. The `het`(8) man page contains information about HET defaults and internal processes.

Note: The default system configuration process includes the HET configuration, but there can be situations in which manual intervention is required. For information about manual configuration, see the `het`(8) man page.

HET accumulates information about system events in the following log file:

`/var/log/het/het_trap_processor.log`

As an event-driven system monitoring tool, HET listens for system events. When HET receives information about an event, it converts the message from coded numbers into a readable form, as follows:

- When a noncritical event occurs, HET simply logs the event. As an option, you can configure an email address to receive noncritical event notifications.
- When a critical event occurs, HET logs the event and sends an email message. SGI recommends that you edit file `/etc/sysconfig/het` and specify an email address specific to your site. By default, HET sends event information to `root@localhost`. For more information about how to customize HET notifications, see one of the following:
  - "Customizing the General HET Notification Script" on page 5
  - "Using Environment Variables to Create a Site-specific HET Notification" on page 6

  The firmware for each baseboard management controller (BMC) and the firmware for each cooling node on an SGI ICE system includes threshold values for each component. If a system condition becomes too low or to high for its threshold, the BMC sends a critical event alert. The following are examples of critical system events that cause an alert:
  - Ambient air temperature outside of recommended range
  - Voltage sensor unable to attain a critical low voltage
  - Power supply failure
  - Loss of redundant power supply
  - Fan speed unable to attain a critical threshold or a loss of fan redundancy
  - Board processor modules that exceed a critical temperature threshold
  - Memory uncorrectable errors

Note: SGI UV 300 systems include firmware-embedded SNMP traps that you can use to send alerts to a system management package, such as HP's OpenView or IBM's Tivoli. Because the required functionality is embedded in the SGI UV 300 system, HET is not applicable. SGI Management Center (SMC) support for SGI UV 300 systems is deferred.

Customizing the General HET Notification Script

You can customize the email addresses to which event information about NON-RECOVERABLE events is sent. Optionally, you can specify a site-specific email address for less-severe events or for all HET events.

The HET log file, /var/log/het/het_trap_processor.log, contains information about all HET events. You can consult this file periodically to monitor noncritical events.

The following procedure explains how to configure an email address or email alias to receive HET notifications.

Procedure 2-1 To customize HET notifications

1. Log in as root and open the following file with a text editor:

   /etc/sysconfig/het

   On an SGI cluster, log into the admin node.

   On an SGI UV system, log into the SMN. If your SGI UV system does not include an SMN, you cannot enable HET.

2. Search the file for the following string:

   HET_MAIL_NON_RECOVERABLE_TO

3. Change the default recipient, root, to be the email address of a person or the email alias of a group who can attend to the system when NON-RECOVERABLE events occur.

4. (Optional) Configure an email recipient for notifications about CRITICAL events.

   Search the file for the following string:

   HET_MAIL_CRITICAL_TO

   Specify an email address or alias to receive CRITICAL event notifications.

5. Save and close file /etc/sysconfig/het.

6. (Optional) Configure an email recipient for all HET events.

   Complete the following steps:

   - Open file /etc/het.action.d/het_mail with a text editor.

- Search for the following lines in `/etc/het.action.d/het_mail`:

```
# NOTE: Adjust if needed
# Default is an empty mailing list audience for
# non (NON-RECOVERABLE or CRITICAL) events.
 to=""
```

- Edit the `to=""` line to specify an email address or an email alias between the quotation marks.
- Save and close the file.

Caution: If you configure an email recipient for all HET events, be aware that the quantity of email could cause excessive network traffic.

Using Environment Variables to Create a Site-specific HET Notification

HET logs all events to the following file:

`/var/log/het/het_trap_processor.log`

The following topics explain how to use the HET environment variables to send the information that resides in the HET log file to an administrator:

- "Creating a Site-specific HET Notification" on page 6
- "HET Environment Variables" on page 7

Creating a Site-specific HET Notification

SGI includes a sample script that that you can edit to send specific notifications to one or more administrators. Be aware of the following information when you customize this script:

- You can edit the sample file to include the environment variables that you need. The sample script is in the following location:

  `/etc/het.action.d/het_user_action.example`

  For information about the environment variables that are available, see the following:

"HET Environment Variables" on page 7

- Make sure to rename the script to a file name that does not include a period (.) character.

  The example script name includes a period character to ensure that the sample file itself does not run. You need to edit and then rename the script to create a functioning notification. For example, you could rename the script to `het_user_action`.

- The content of the sample script is as follows:

```
#! /bin/sh

#   Copyright (c) 2013 Silicon Graphics, Inc.
#   This work is held in copyright as an unpublished work by
#   Silicon Graphics, Inc.  All rights reserved.

#==============================================================================
#HET - Example Action Program
#==============================================================================
# See README in this directory.
# For this script to run on every alert, it needs to be renamed to remove '.':
#   $mv het_user_action.example het_user_action

OUTPUTDIR=/tmp
HET_OUTFILE=het_user_action.out

echo "The following HET(Hardware Environment Tracking) event has been recorded:" >>$OUTPUTDIR/$HET_OUTFI
# Without some kind of filter, this will run on every alert.
echo "Severity: $HET_ALERTSEVERITY">>$OUTPUTDIR/$HET_OUTFILE
echo "Event Details:">>$OUTPUTDIR/$HET_OUTFILE
printenv | grep HET | sort -t= -k 1 | awk -F= '{printf "\t%-30s %s\n",$1,$2;}'>>$OUTPUTDIR/$HET_OUTFILE
```

HET Environment Variables

The HET README file lists all the HET environment variables that you can include in a notification script. This file resides in `/etc/het.action.d/README` and is as follows:

```
==============================================================================
HET - Action programs ( scripts or binaries )
==============================================================================
```

```
Every file present in this directory will get executed with the
exception of:
- README*
- *template
- directories
- permissions not set to 500 or 700
- filename contains a '.'. Example: het_mail.rpmnew will not get executed.
================================================================================
Every program is called without any command-line argument. However, the
environment is filled using HET_xxx=value for each piece of info avalable
for the TRAP.
================================================================================
List of HET_xxx variables
================================================================================

         Variable                       Example Value
--------------------------------------------------------------------------
         HET_AGENTADDR                  172.19.1.1
         HET_ALERTSEVERITY              CRITICAL
         HET_COMMUNITY                  sgi
         HET_ESP_SEPARATOR              800009
         HET_EVENT1                     0x5b
         HET_EVENT2                     0x37
         HET_EVENT3                     0x36
         HET_EVENT                      unrGoingHigh
         HET_EVENTCLASSNAME             threshold
         HET_EVENTOFFSET                11
         HET_EVENTSOURCETYPE            0x20
         HET_EVENTTYPE                  1
         HET_GUID                       d00101-10050
         HET_HOST                       r1i1c
         HET_IDENT                      Harp 50 4 rack UV2 SSI
         HET_IFACE
         HET_INTERVAL                   10
         HET_OID                        .1.3.6.1.4.1.3183.1.1
         HET_PASSWORD                   ADMIN
         HET_PORT                       22162
         HET_SENSORNAME                 Volt0
         HET_SENSORNUMBER               0x00
         HET_SENSORTYPE                 2
         HET_SENSORTYPENAME             voltage
```

```
HET_SENSORVALUE              3.33 Volts
HET_SN                       UV2-00000050
HET_SPECIFICTRAP             0x0002010b
HET_SRCADDR                  127.0.0.1
HET_TIMESTAMP                680380536
HET_TRAPSOURCETYPE           0x20
HET_USER                     ADMIN
HET_WHEN                     2012-05-28.17.21.07 CDT  FIN
```

HET Examples

Example 1. The following is an example of a HET log file that contains critical information:

```
dump     2013-10-23.07.13.21 [het_process_thread:2] # begin ---------------------------------
dump     2013-10-23.07.13.21 [het_process_thread:2] agentAddr        172.24.0.2
dump     2013-10-23.07.13.21 [het_process_thread:2] het_type         ipmi
dump     2013-10-23.07.13.21 [het_process_thread:2] guid             r1lead
dump     2013-10-23.07.13.21 [het_process_thread:2] sn               X1------
dump     2013-10-23.07.13.21 [het_process_thread:2] alertSeverity    NONE
dump     2013-10-23.07.13.21 [het_process_thread:2] event            uncorrectableECC
dump     2013-10-23.07.13.21 [het_process_thread:2] sensorName       None-memory
dump     2013-10-23.07.13.21 [het_process_thread:2] sensorNumber     0x00
dump     2013-10-23.07.13.21 [het_process_thread:2] sensorTypeName   memory
dump     2013-10-23.07.13.21 [het_process_thread:2] eventClassName   discrete
dump     2013-10-23.07.13.21 [het_process_thread:2] event1           0x51
dump     2013-10-23.07.13.21 [het_process_thread:2] event2           0xff
dump     2013-10-23.07.13.21 [het_process_thread:2] event3           0x51
dump     2013-10-23.07.13.21 [het_process_thread:2] flap_count       1
dump     2013-10-23.07.13.21 [het_process_thread:2] # end -----------------------------------
```

The corresponding email message that HET sends is as follows:

```
X-Original-To: root
Delivered-To: root@saturn9-1.americas.sgi.com
Date: Wed, 18 Dec 2014 14:36:52 -0600
From: HET.ALERT.donotreply@saturn9-1.americas.sgi.com
To: root@saturn9-1.americas.sgi.com
Subject: HET ALERT from cb9 - NON-RECOVERABLE
User-Agent: Heirloom mailx 12.2 01/07/07
```

```
The following HET(Hardware Environment Tracking) event has been recorded:
HET ALERT from cb9 - NON-RECOVERABLE

Event Details:

        EVENT                           uncorrectableECC
        HET                             r1i0n4
        LOCATION                        r1i0n4
        SENSOR                          None-memory
        SENSORNUMBER                    0x00
        SENSORTHRESHOLD                 81
        SENSORTYPE                      memory
        SENSORVALUE                     255
        SEVERITY                        NON-RECOVERABLE
        SN                              X1------
        TYPE                            ipmi
```

Enabling CPU Frequency Scaling

This chapter includes the following topics:

About CPU Frequency Scaling

CPU frequency scaling allows the operating system to scale the processor frequency automatically and dynamically. CPU frequency scaling needs to be enabled in a compute node image if you want your SGI system to take advantage of the Intel Turbo Boost technology that is built into each processor.

The Intel Turbo Boost Technology allows processor cores to run faster than the base operating frequency as long as they are operating below the limits set for power, current, and temperature. The CPU frequency scaling setting also affects power consumption and enables you to manage power consumption. For example, theoretically, you can cut power consumption if you clock the processors from 2 GHz down to 1 GHz.

Determining Your System's Possible CPU Frequency Settings

The CPU frequency settings that are available on your SGI system depend on the presence of the following directory:

`/sys/devices/system/cpu/intel_pstate`

After you check your SGI system for the presence of the `intel_pstate` directory, you can select from the following CPU frequency governor settings:

- Governor settings for SGI systems that include the `intel_pstate` directory:
  - `performance` (default)

  - `powersave`

- Governor settings for SGI systems that do not include the `intel_pstate` directory:
  - `conservative`
  - `ondemand` (default)
  - `performance`
  - `powersave`
  - `userspace`

The `performance` setting directs the processors to run at or near their maximum speeds. The `powersave` setting slows down the CPUs and might be suitable for your site during periods of low use. If you want to set a nondefault option, use the documentation in this chapter to choose the option and update your configuration.

CPU Frequency Scaling for SGI ICE Clusters and SGI Rackable Clusters

The following topics explain how to configure CPU frequency scaling on SGI clusters:

- "Enabling or Disabling CPU Frequency Scaling (SGI ICE Clusters and SGI Rackable Clusters)" on page 12
- "(Optional) Changing the Governor Setting and Configuring Turbo Mode (SGI ICE Clusters and SGI Rackable Clusters)" on page 14

Enabling or Disabling CPU Frequency Scaling (SGI ICE Clusters and SGI Rackable Clusters)

The procedure in this topic explains how to enable or disable CPU frequency scaling. CPU frequency scaling is disabled by default on SGI clusters.

The following procedure explains how to change your CPU frequency scaling setting.

Procedure 3-1 To control CPU frequency scaling

1. Log into the admin node as `root`.
2. Use the `cimage --list-images` command to retrieve a list of the compute node images you can edit:

For example:

```
# cimage --list-images
image: ice-compute-sles11sp3.mpt
        kernel: 3.0.76-0.11-default
        kernel: 3.0.76-0.11-trace
image: ice-compute-sles11sp3
        kernel: 3.0.76-0.11-default
```

The previous example shows the names of two images: `ice-compute-sles11sp3.mpt` and `ice-compute-sles11sp3`.

3. Type the following command to install the system images that support CPU frequency scaling:

   # **cinstallman --yum-image --image *image_name* install sgi-base-configuration**

   For *image_name*, specify one of the compute images. For example, `ice-compute-sles11sp3.mpt` and `ice-compute-sles11sp3`.

4. Type the following command to change to the directory that contains the image you want to edit:

   # **chroot /var/lib/systemimager/images/*image_name***

   For *image_name*, specify one of the compute node image names that the `cimage` command returned. For example, using the output from the preceding step, specify either `ice-compute-sles11sp3.mpt` or `ice-compute-sles11sp3`.

5. Use a text editor to open file `/etc/modprobe.d/acpi-cpufreq.conf`.

6. Note the following line in this file:

   ```
   install acpi-cpufreq /bin/true
   ```

   To enable CPU frequency scaling, insert a pound (#) character as the first character in this line, which makes the line appear as follows:

   ```
   #install acpi-cpufreq /bin/true
   ```

   To disable CPU frequency scaling, make sure that the `install acpi-cpufreq /bin/true` line does not contain a # character in column 1, which makes the line appear as follows:

   ```
   install acpi-cpufreq /bin/true
   ```

7. Save and close file `/etc/modprobe.d/acpi-cpufreq.conf`.
8. Push the changes out to the compute nodes.

   For information about how to push changes to compute nodes, see *SGI Management Center (SMC) Administration Guide for Clusters.*
9. (Optional) Change the CPU frequency governor setting and configure turbo mode.

   The default governor setting and the default turbo mode setting are appropriate for most SGI ICE systems. If you want to change these settings, proceed to the following:

   "(Optional) Changing the Governor Setting and Configuring Turbo Mode (SGI ICE Clusters and SGI Rackable Clusters)" on page 14

(Optional) Changing the Governor Setting and Configuring Turbo Mode (SGI ICE Clusters and SGI Rackable Clusters)

Use the procedure in this topic to change the governor setting and, optionally, to configure turbo mode. When you enable turbo mode, you enable the CPU frequency to exceed its nominal level for short periods of time, depending on the processor, temperature, current, power, and other factors. For general information about turbo mode, see the following website:

https://www-ssl.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html

The following procedure explains how to set the CPU frequency governor appropriately and how to configure turbo mode.

Procedure 3-2 To change the governor setting and configure turbo mode

1. Make sure that CPU frequency is enabled.

   For information, see "Enabling or Disabling CPU Frequency Scaling (SGI ICE Clusters and SGI Rackable Clusters)" on page 12.
2. Examine the following list and choose a power governor setting:

***governor* Setting**	**Effect**
`ondemand`	Dynamically switches between the available CPUs if at 95% of CPU load. Default.

`performance`	Runs the CPUs at the maximum frequency.
`conservative`	Dynamically switches between the available CPUs if at 75% of CPU load.
`powersave`	Runs the CPUs at the minimum frequency.
`userspace`	Runs the CPUs at user-specified frequencies.

3. Use the `cimage --list-images` command to retrieve a list of the compute node images you can edit:

   For example:

   ```
   # cimage --list-images
   image: ice-compute-sles11sp3.mpt
           kernel: 3.0.76-0.11-default
           kernel: 3.0.76-0.11-trace
   image: ice-compute-sles11sp3
           kernel: 3.0.76-0.11-default
   ```

   The previous example shows the names of two images: `ice-compute-sles11sp3.mpt` and `ice-compute-sles11sp3`.

4. Type the following command to change to the directory that contains the image you want to edit:

   # **chroot /var/lib/systemimager/images/*image_name***

   For *image_name*, specify one of the compute node image names that the `cimage` command returned. For example, using the output from the preceding step, specify either `ice-compute-sles11sp3.mpt` or `ice-compute-sles11sp3`.

5. Use one of the following platform-specific methods to change the setting:

   - On RHEL platforms, complete the following steps:

     1. Open file `/etc/sysconfig/cpuspeed`.
     2. Search for the `GOVERNOR=` string.
     3. Edit the setting, adding the *governor* setting you chose in the previous step.
     4. Save and close the file.

   - On SLES platforms, complete the following steps:

     1. Use a text editor to open file `/etc/init.d/after.local`.

2. Add the following line to ensure that the system sets the *governor* setting you specified after each boot:

   `cpupower freqency-set -g` *governor*

6. Type the following command to exit the chroot environment:

   # **exit**

7. Push the changes out to the compute nodes.

   For information about how to push changes to compute nodes, see *SGI Management Center (SMC) Administration Guide for Clusters*.

8. Use the `ssh`(1) command to log into one of the compute nodes.

   Make sure that the compute node you select is one of the compute nodes upon which you want to configure CPU frequency scaling.

9. Use the `cat`(1) command to retrieve the list of available frequencies. For example:

   # **cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies**
   3301000 3300000 3200000 3100000 3000000 2900000 2800000 2700000 2600000
   2500000 2400000 2300000 2200000 2100000 2000000 1900000 1800000 1700000
   1600000 1500000 1400000 1300000 1200000

   The preceding output shows the available frequencies, listed in order from the highest, 3301000 KHz, to the lowest, 1200000 KHz.

   On SGI systems, the second frequency listed is always the processor's nominal frequency. This is a 3.3 GHz processor, so 3300000 KHz is the nominal frequency.

   You can also obtain the nominal frequency by typing the following command and examining the information in the model name field:

   # **cat /proc/cpuinfo**

10. Set the CPU frequency on the nodes.

    The commands to accomplish this task differ depending on the number of nodes, as follows:

    - For one compute node, use the `cpupower` command to set the frequency to the nominal frequency of 3.3 GHz plus 1 MHz.

That is, specify a frequency of 3301 MHz. For example:

```
# cpupower frequency-set -u 3301MHz
```

- For multiple compute nodes, use the pdsh(1) command to set the frequency. The pdsh(1) command operates on more than one node at a time.

  Example 1. The following command sets the CPU frequency on the RLCs:

  ```
  # pdsh -g leader ice-compute cpupower frequency-set -u 3301MHz
  ```

  Example 2. The following command sets the CPU frequency on all the compute nodes within a cluster:

  ```
  # pdsh -g ice-compute cpupower frequency-set -u 3301MHz
  ```

  For more information about the pdsh(1) command, see the following:

  SGI Management Center (SMC) Administration Guide for Clusters

Later, if you want to disable turbo mode, type the following command to set the maximum frequency back to the nominal frequency:

```
# cpupower frequency-set -u 3300MHz
```

CPU Frequency Scaling for SGI UV Servers

The following topics explain how to configure CPU frequency scaling on SGI UV servers:

- "Configuring the powersave Setting on SGI UV Servers That Include the intel_pstate Directory" on page 18
- "Enabling CPU Frequency Scaling on SGI UV Servers That do not Include the intel_pstate Directory" on page 18
- "Changing the Governor Setting and Configuring Turbo Mode on SGI UV Servers That do not Include the intel_pstate Directory" on page 19

Configuring the powersave Setting on SGI UV Servers That Include the intel_pstate Directory

By default, the CPU frequency setting on SGI UV servers that include the intel_pstate directory is performance. During non-peak production times, you might want to configure the powersave setting.

For information about the intel_pstate directory, see the following:

"Determining Your System's Possible CPU Frequency Settings" on page 11

The following procedure explains how to configure the powersave setting.

Procedure 3-3 To configure the powersave setting

1. Log in as root to the system you want to configure.
2. Type the following command:

   # **cpupower frequency-set -g powersave**
3. Type the following command:

   # **cpupower frequency-info**

   Verify that the powersave setting appears in the command output in the current policy field.
4. (Optional) Use a text editor to edit the /etc/init.d/after.local file and add the following line:

   ```
   cpupower frequency-set -g powersave
   ```

   The preceding line ensures that after each boot, the system sets the powersave setting.

Enabling CPU Frequency Scaling on SGI UV Servers That do not Include the intel_pstate Directory

The procedure in this topic explains how to enable or disable CPU frequency scaling on SGI UV systems that do not include the intel_pstate directory.

Procedure 3-4 To enable CPU frequency scaling

1. Log in as root to the system you want to configure.

2. Use a text editor to open file `/etc/sysconfig/x86config`, and verify or change the system setting from within this file.

   This file contains the settings that enable or disable CPU frequency scaling.

   To enable CPU frequency scaling, set `UV_DISABLE_CPU_FREQUENCY_SCALING=no`.

   To disable CPU frequency scaling, set `UV_DISABLE_CPU_FREQUENCY_SCALING=yes`.

3. Type the following command to propagate the new system setting:

   # **/usr/sbin/x86config**

4. Type one of the following commands to restart services:

   - On RHEL platforms, type the following:

     # **service cpuspeed restart**

   - On SLES 12 platforms, type the following:

     # **modprobe acpi_cpufreq**

   - On SLES 11 platforms, type the following:

     # **service haldaemon restart**

5. Change the CPU frequency governor setting and configure turbo mode.

   Proceed to the following:

   "Changing the Governor Setting and Configuring Turbo Mode on SGI UV Servers That do not Include the `intel_pstate` Directory" on page 19

Changing the Governor Setting and Configuring Turbo Mode on SGI UV Servers That do not Include the `intel_pstate` Directory

The default CPU frequency governor setting can inhibit the system's performance. Use the procedure in this topic to change the governor setting and, optionally, to configure turbo mode. When you enable turbo mode, you enable the CPU frequency to exceed its nominal level for short periods of time, depending on the processor, temperature, current, power, and other factors. For general information about turbo mode, see the following website:

https://www-ssl.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html

The following procedure explains how to set the CPU frequency governor appropriately and how to configure turbo mode.

Procedure 3-5 To change the governor setting and configure turbo mode

1. Make sure that CPU frequency is enabled.

   For information, see "Enabling CPU Frequency Scaling on SGI UV Servers That do not Include the `intel_pstate` Directory" on page 18.

2. Decide which governor setting is suitable for your site.

   `ondemand` is the the default setting. SGI recommends that you change this to a site-specific setting. SGI recommends that you configure the governor to `performance`.

   The possible power governor settings are as follows:

***governor* Setting**	**Effect**
`ondemand`	Dynamically switches between the available CPUs if at 95% of CPU load. Default. SGI does not recommend this setting. Consider using the `performance` setting.
`performance`	Runs the CPUs at the maximum frequency. SGI recommends this setting.
`conservative`	Dynamically switches between the available CPUs if at 75% of CPU load.
`powersave`	Runs the CPUs at the minimum frequency.
`userspace`	Runs the CPUs at user-specified frequencies.

3. Use one of the following platform-specific methods to change the setting:

   - On RHEL 6 platforms, complete the following steps:

     1. Open file `/etc/sysconfig/cpuspeed`.
     2. Search for the `GOVERNOR=` string.
     3. Edit the setting, adding the *governor* setting you chose in the previous step.

4. Save and close the file.
5. Type the following command:

   # **service cpuspeed restart**

- On RHEL 7, SLES 12, and SLES 11 platforms, complete the following steps:

  1. Type the following command:

     # **cpupower frequency-set -g *governor***

     For *governor*, specify the setting you chose in the previous step.

  2. Type the following command:

     # **cpupower frequency-info**

  3. Verify that the *governor* setting you specified appears in the command in the output in the `current policy` field.

  4. Use a text editor to edit the `/etc/init.d/after.local` file and add the following line:

     `cpupower freqency-set -g` *governor*

     The preceding line ensures that after each boot, the system sets the *governor* setting you specified.

  Complete the rest of this procedure if you want to configure turbo mode. If your goal was to configure a nondefault governor setting, you do not need to perform the remaining steps in this procedure.

4. Use the `cat`(1) command to retrieve the list of available frequencies. For example:

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
3301000 3300000 3200000 3100000 3000000 2900000 2800000 2700000 2600000
2500000 2400000 2300000 2200000 2100000 2000000 1900000 1800000 1700000
1600000 1500000 1400000 1300000 1200000
```

   The preceding output shows the available frequencies, listed in order from the highest, 3301000 KHz, to the lowest, 1200000 KHz.

   On SGI systems, the second frequency listed is always the processor's nominal frequency. This is a 3.3 GHz processor, so 3300000 KHz is the nominal frequency.

You can also obtain the nominal frequency by typing the following command and examining the information in the model name field:

```
# cat /proc/cpuinfo
```

5. Use the `cpupower` command to set the frequency to the nominal frequency of 3.3 GHz plus 1 MHz.

   That is, specify a frequency of 3301 MHz. For example:

   ```
   # cpupower frequency-set -u 3301MHz
   ```

   Later, if you want to disable turbo mode, type the following command to set the maximum frequency back to the nominal frequency:

   ```
   # cpupower frequency-set -u 3300MHz
   ```

Monitoring Main Memory Health

This chapter includes the following topics:

About Main Memory Health Monitoring

The SGI `memlog`(8) utility monitors the overall system health of each dual inline memory module (DIMM) on your SGI system. The `memlog` utility is configured for your system when the SGI Foundation Software (SFS) is installed.

To verify that `memlog` is running, type the following command:

```
# service memlog status
```

For more information about `memlog`, see the `memlog`(8) man page.

Retrieving Main Memory Health Information

SGI recommends that you check your SGI computer system periodically to determine whether the `memlog`(8) utility has reported any hardware errors.

The `memlog` utility verifies and diagnoses problems with the DIMMs. The utility's messages appear in `/var/log/messages`. If you are using SGI Management Center (SMC), you can retrieve reports that include MEMLOG information through the Nagios interface.

The following topics explain how to access information from the `memlog` utility through the monitoring tools or by using commands:

Accessing `memlog`(8) Messages With Nagios

Nagios scans the MEMLOG entries for each node, and it issues escalation messages when it detects MEMLOG messages that contain certain keywords. For example, Nagios escalates MEMLOG messages that contain the keywords `uncorrected`, `error`, or `warning`. It issues a critical message when it detects MEMLOG messages that include the strings `uncorrected` or `error`. It issues a warning message when it detects MEMLOG messages that include the keyword `warning`.

The memory management system logs that Nagios monitors reside in `/var/log/nagios-memlog`. The log files are rotated.

The following procedure explains how to retrieve a report that includes Nagios's messages for MEMLOG.

Procedure 4-1 To retrieve MEMLOG information

1. Log into the admin node as the root user.
2. Open a web browser, and type the following URL in the address bar:

   `http://`*admin_hostname*`/nagios`

   For example, to run Nagios on a cluster named `gazelle`, type the following:

   ```
   http://gazelle/nagios
   ```

3. Provide the userid and password credentials when prompted.

   It is possible that the login credentials have been customized for your site. The default username is `nagiosadmin`. The default password is `sgisgi`.

   The following manual contains a procedure that explains how to change the login credentials:

   SGI Management Center (SMC) Administration Guide for Clusters

4. In the left pane, click **Services** to create the Service Status report.

   Figure 4-1 on page 25 shows an example Service Status report. The **Status** column indicates that Nagios has detected a MEMLOG message.

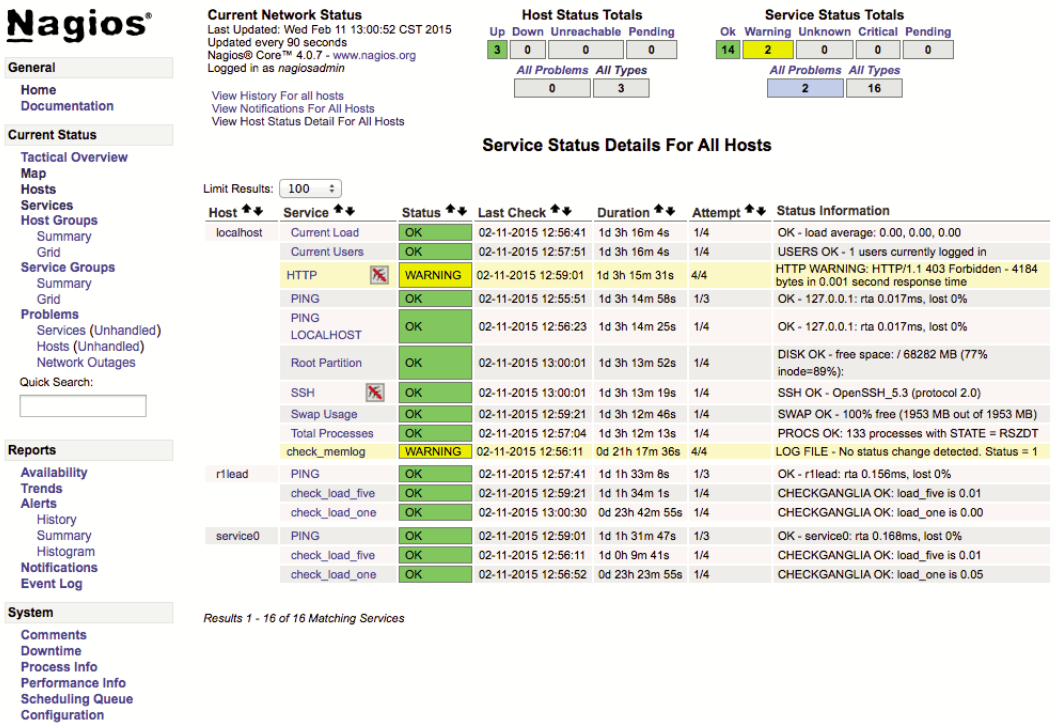

Figure 4-1 MEMLOG Messages in the Service Status Report

5. In the Service Status report, click **check_memlog** to retrieve more information about the MEMLOG information in the Service Status report.

   Figure 4-2 on page 26 shows an example Service report for MEMLOG.

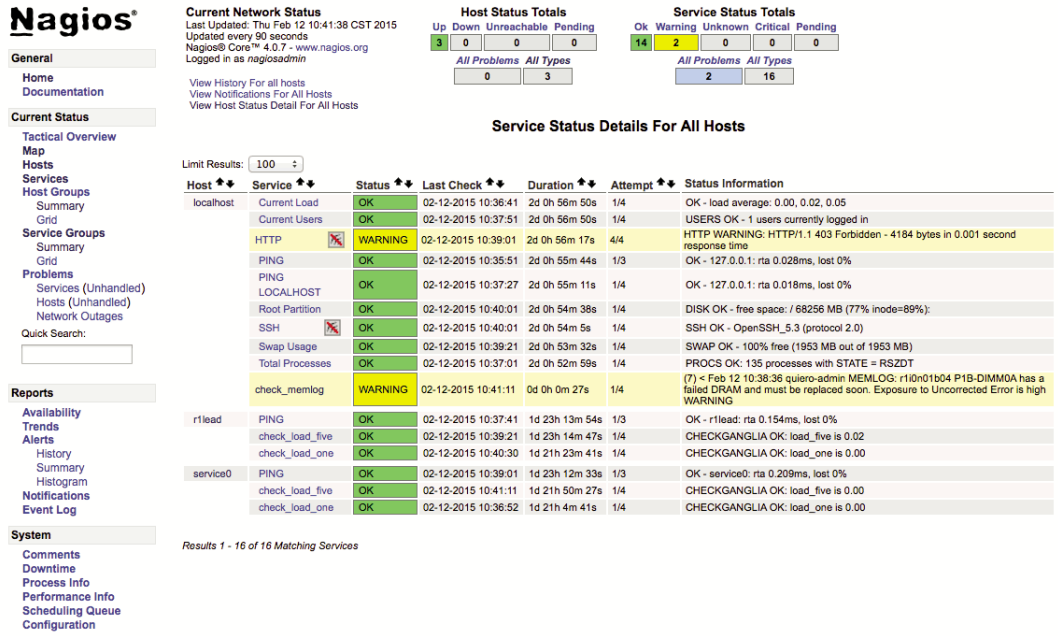

Figure 4-2 Service Information for MEMLOG

Accessing `memlog`(8) Messages With Commands

The following are two ways to use commands to retrieve information about memory problems or memory health:

- Scan the system log for entries that contain the string `MEMLOG`. If problems arise with any of the DIMMs on your system, `memlog` writes a message to `/var/log/messages`. To retrieve these messages, type the following command:

```
# grep MEMLOG /var/log/messages

r1i0n0:Dec  9 07:29:45 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 DRAM U9 DQ4 Temp = 21C
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]:  P1-DIMM1A has a failed DRAM and must be replaced soon.
     Exposure to Uncorrected Error is high
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 Bank 0 Row 0x0 Col 0x8 Temp = 21C
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 DRAM U9 DQ4 Temp = 21C
r1i0n0:Dec  9 07:30:12 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 Temp = 22C
r1i0n0:Dec  9 07:30:12 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 DRAM U9 DQ4 Temp = 22C
```

```
r1i0n0:Dec  9 07:30:25 r1i0n0 MEMLOG[4595]:  P1-DIMM3A has a failed DRAM and must be replaced soon.
     Exposure to Uncorrected Error is high
r1i0n0:Dec  9 07:30:25 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 Bank 0 Row 0x0 Col 0x8 Temp = 22C
```

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

- Use the `memlog` command to retrieve a report. The report lists all the DIMMS in the system and contains an error summary for each DIMM. To obtain this report, type the following command:

```
r1i0n0:~ # memlogd -c
user config match for X9DRT-Dakota
found 2 sockets, highest socket number 1, deviceID Ivybridge, mem ctlrs/socket 1.
  P1-DIMM1A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 405031E4 Part HMT31GR7EFR4C-RD    1867
 Tue Dec  9 07:28:48 2014 Rank 0 Dram U9 Bank 0 Row 0x0 Col 0x8 multiaddress C DQ4  Temp = 21C hits 19
 Tue Dec  9 07:31:31 2014 Rank 1 Dram U9B Bank 0 Row 0x0 Col 0x0 single DQ4  Temp = 21C hits 1
  P1-DIMM2A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 409031CA Part HMT31GR7EFR4C-RD    1867
  P1-DIMM3A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 405031DE Part HMT31GR7EFR4C-RD    1867
 Tue Dec  9 07:30:12 2014 Rank 0 Dram U9 Bank 0 Row 0x0 Col 0x8 multiaddress C DQ4  Temp = 22C hits 2
  P1-DIMM4A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 40C031C7 Part HMT31GR7EFR4C-RD    1867
  P2-DIMM1A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 402031AA Part HMT31GR7EFR4C-RD    1867
  P2-DIMM2A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 407031A8 Part HMT31GR7EFR4C-RD    1867
  P2-DIMM3A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 407031E7 Part HMT31GR7EFR4C-RD    1867
  P2-DIMM4A Size 8192MB Width 4 Rank 2 Row 15 Col 11 Bank 8 Serial 40C031E8 Part HMT31GR7EFR4C-RD    1867
```

The preceding example output includes 8 DIMMs. Note the following in this output:

- Information about the first DIMM, `P1-DIMM1A`, is on the first line. The two lines that follow are the DIMM repair tag, which shows that this DIMM has been encountering corrected errors.
- The third DIMM, `P1-DIMM3A`, has also encountered corrected errors.
- The last number on each line of inventory is `1867`. This number is the channel speed that the memory controller set at boot time for that DIMM.

Monitoring System Performance

This chapter includes the following topics:

About the System Monitoring Software

You can use Linux utilities, SFS utilities, and open source utilities to monitor system performance. Some of these utilities are as follows:

- Linux utilities, such as `w`(1), `ps`(1), `top`(1), `vmstat`(8), `iostat`(1), and `sar`(1). For information about these commands, see their `man`(1) pages or the following SGI publication:

  Linux Application Tuning Guide for SGI X86-64 Based Systems

- SFS utilities, such as `hubstats`(1), `linkstat-uv`(1), `nodeinfo`(1), and `topology`(1). This chapter describes how to use these commands.

`hubstats`(1) Command (SGI UV Systems Only)

The `hubstats`(1) command monitors NUMAlink traffic, directory cache operations, and global reference unit (GRU) traffic statistics on SGI UV systems. For more information, see the `hubstats`(1) man page.

`linkstat-uv`(1) Command (SGI UV Systems Only)

The `linkstat-uv`(1) command monitors NUMAlink traffic and error rates on SGI UV systems. The `linkstat-uv`(1) command returns information about packets and

Mbytes sent/received on each NUMAlink in the system. It also returns information about error rates. It is useful as a performance monitoring tool and as a tool for helping you to diagnose and identify faulty hardware. For more information, see the `linkstat-uv`(1) man page.

Note that this command is specific to SGI UV systems and does not return the same information as the `linkstat`(1) command.

`nodeinfo` Command (SGI UV Systems Only)

`nodeinfo`(1) is a tool for monitoring per-node NUMA memory statistics on SGI UV systems. The `nodeinfo` tool reads `/sys/devices/system/node/*/meminfo` and `/sys/devices/system/node/*/numastat` on the local system to gather NUMA memory statistics.

Sample memory statistics from the `nodeinfo`(1) command are as follows:

```
uv44-sys:~ # nodeinfo
Memory Statistics  Tue Oct 26 12:01:58 2010
uv44-sys
      ---------------------- Per Node KB -----------------------  ------ Preferred Alloc -------  -- Loc/Rem ---
node       Total        Free     Used   Dirty    Anon        Slab     hit    miss foreign interlv   local  remote
   0    16757488    16277084   480404      52   34284       36288   20724       0       0       0   20720       4
   1    16777216    16433988   343228      68    6772       17708    4477       0       0       0    3381    1096
   2    16777216    16438568   338648      76    6908       12620    1804       0       0       0     709    1095
   3    16760832    16429844   330988      56    2820       16836    1802       0       0       0     708    1094
   4    16777216    16444408   332808      88   10124       13588    1517       0       0       0     417    1100
   5    16760832    16430300   330532      72    1956       17304    4546       0       0       0    3453    1093
   6    16777216    16430788   346428      36    3236       15292    3961       0       0       0    2864    1097
   7    16760832    16435532   325300      44    1220       14800    3971       0       0       0    2877    1094
 TOT   134148848   131320512  2828336     492   67320      144436   42802       0       0       0   35129    7673
 Press "h" for help
```

From an interactive `nodeinfo` session, enter **h** for a help statement. For example:

```
Display memory statistics by node.
        q   quit
        +   Increase starting node number. Used only if more nodes than will
            fit in the current window.
        -   Decrease starting node number. Used only if more nodes than will
            fit in the current window.
```

```
        b   Start output with node 0.
        e   Show highest node number.
        k   show sizes in KB.
        m   show sizes in MB.
        p   show sizes in pages.
        t   Change refresh rate.
        A   Show/Hide memory policy stats.
        H   Show/Hide hugepage info.
        L   Show/Hide LRU Queue stats.

Field definitions:
        hit - page was allocated on the preferred node
        miss - preferred node was full. Allocation occurred on THIS node
               by a process running on another node that was full

        foreign - Preferred node was full. Had to allocate somewhere
                  else.

        interlv - allocation was for interleaved policy

        local - page allocated on THIS node by a process running on THIS node
        remote - page allocated on THIS node by a process running on ANOTHER node


  (press any key to exit from help screen)
```

`topology`(1) Command (SGI UV Systems Only)

The `topology`(1) command provides topology information about your system. Application programmers can use the `topology`(1) command to help optimize execution layout for their applications.

The `topology`(1) command includes many options. For more information, type `topology --help` on the command line.

Example 1. The following `topology`(1) command shows the system summary.

```
uv-sys:~ # topology
System type: UV2000
System name: harp34-sys
Serial number: UV2-00000034
```

```
Partition number: 0
       8 Blades
     256 CPUs
      16 Nodes
  235.82 GB Memory Total
   15.00 GB Max Memory on any Node
       1 BASE I/O Riser
       2 Network Controllers
       2 Storage Controllers
       2 USB Controllers
       1 VGA GPU
```

Example 2. The following `topology`(1) command explicitly requests the system summary and also shows node and CPU information.

```
uv-sys:~ # topology --summary --nodes --cpus
System type: UV2000
System name: harp34-sys
Serial number: UV2-00000034
Partition number: 0
       8 Blades
     256 CPUs
      16 Nodes
  235.82 GB Memory Total
   15.00 GB Max Memory on any Node
       1 BASE I/O Riser
       2 Network Controllers
       2 Storage Controllers
       2 USB Controllers
       1 VGA GPU

Index      ID         NASID CPUS     Memory
-------------------------------------------
    0 r001i11b00h0      0   16      15316 MB
    1 r001i11b00h1      2   16      15344 MB
    2 r001i11b01h0      4   16      15344 MB
    3 r001i11b01h1      6   16      15344 MB
    4 r001i11b02h0      8   16      15344 MB
    5 r001i11b02h1     10   16      15344 MB
    6 r001i11b03h0     12   16      15344 MB
    7 r001i11b03h1     14   16      15344 MB
    8 r001i11b04h0     16   16      15344 MB
```

```
    9 r001i11b04h1     18   16      15344 MB
   10 r001i11b05h0     20   16      15344 MB
   11 r001i11b05h1     22   16      15344 MB
   12 r001i11b06h0     24   16      15344 MB
   13 r001i11b06h1     26   16      15344 MB
   14 r001i11b07h0     28   16      15344 MB
   15 r001i11b07h1     30   16      15344 MB

CPU        Blade PhysID CoreID APIC-ID Family Model Speed L1(KiB) L2(KiB) L3(KiB)
------------------------------------------------------------------------------
  0 r001i11b00h0     00     00       0      6    45  2599 32d/32i     256   20480
  1 r001i11b00h0     00     01       2      6    45  2599 32d/32i     256   20480
  2 r001i11b00h0     00     02       4      6    45  2599 32d/32i     256   20480
  3 r001i11b00h0     00     03       6      6    45  2599 32d/32i     256   20480
  4 r001i11b00h0     00     04       8      6    45  2599 32d/32i     256   20480
  5 r001i11b00h0     00     05      10      6    45  2599 32d/32i     256   20480
  6 r001i11b00h0     00     06      12      6    45  2599 32d/32i     256   20480
  7 r001i11b00h0     00     07      14      6    45  2599 32d/32i     256   20480
  8 r001i11b00h1     01     00      32      6    45  2599 32d/32i     256   20480
  9 r001i11b00h1     01     01      34      6    45  2599 32d/32i     256   20480
 10 r001i11b00h1     01     02      36      6    45  2599 32d/32i     256   20480
 11 r001i11b00h1     01     03      38      6    45  2599 32d/32i     256   20480
.
.
.
```

Example 3. The following topology(1) command shows the interrupt requests assigned to devices.

```
uv300a-sys:~ # topology --irq

Index Location    NASID  PCI Address           IRQ(s)    Device
-----------------------------------------------------------------
    0 r001i01s00      0  0000:00:1f.2             519   Intel SATA RAID Controller
    .          .      .  0000:02:00.0       1529-1532   Intel I210 Gigabit Network Connection
    .          .      .  0000:06:00.0             255   Matrox G200eR2
    4 r001i06s01      8  0001:01:00.0    56,1511-1526   LSI SAS2308 Fusion-MPT SAS-2
    4 r001i06s02      8  0001:02:00.0    64,1480-1510   Intel P3700 Non-Volatile Memory Controller
    4 r001i06s03      8  0001:03:00.0    66,1527,1533-1562   Intel P3700 Non-Volatile Memory Controller
    5 r001i06s05     10  0002:02:00.0    88,1563-1593   Intel P3700 Non-Volatile Memory Controller
    5 r001i06s06     10  0002:03:00.0    90,1594-1624   Intel P3700 Non-Volatile Memory Controller
    6 r001i06s07     12  0003:01:00.0   104,1625-1655   Intel P3700 Non-Volatile Memory Controller
```

```
    6 r001i06s08     12  0003:02:00.0   106,1656-1686   Intel P3700 Non-Volatile Memory Controller
    7 r001i06s10     14  0004:01:00.0   128,1687-1717   Intel P3700 Non-Volatile Memory Controller
    7 r001i06s11     14  0004:02:00.0   130,1718-1748   Intel P3700 Non-Volatile Memory Controller
   12 r001i16s01     24  0005:01:00.0   152,2493-2508   LSI SAS2308 Fusion-MPT SAS-2
   12 r001i16s02     24  0005:02:00.0   160,1749-1779   Intel P3700 Non-Volatile Memory Controller
   12 r001i16s03     24  0005:03:00.0   162,1780-1810   Intel P3700 Non-Volatile Memory Controller
   13 r001i16s05     26  0006:02:00.0   184,1811-1841   Intel P3700 Non-Volatile Memory Controller
   13 r001i16s06     26  0006:03:00.0   186,1842-1872   Intel P3700 Non-Volatile Memory Controller
   14 r001i16s07     28  0007:01:00.0   200,1873-1903   Intel P3700 Non-Volatile Memory Controller
   14 r001i16s08     28  0007:02:00.0   202,1904-1934   Intel P3700 Non-Volatile Memory Controller
   15 r001i16s10     30  0008:01:00.0   224,1935-1965   Intel P3700 Non-Volatile Memory Controller
   15 r001i16s11     30  0008:02:00.0   226,1966-1996   Intel P3700 Non-Volatile Memory Controller
   20 r001i28s01     40  0009:01:00.0            2558   NVIDIA GK110BGL [Tesla K40m]
   20 r001i28s02     40  0009:02:00.0   256,1997-2027   Intel P3700 Non-Volatile Memory Controller
   20 r001i28s03     40  0009:03:00.0   258,2028-2058   Intel P3700 Non-Volatile Memory Controller
   21 r001i28s04     42  000a:01:00.0            2557   NVIDIA GK110BGL [Tesla K40m]
   21 r001i28s05     42  000a:02:00.0   280,2059-2089   Intel P3700 Non-Volatile Memory Controller
   21 r001i28s06     42  000a:03:00.0   282,2090-2120   Intel P3700 Non-Volatile Memory Controller
   22 r001i28s07     44  000b:01:00.0   296,2121-2151   Intel P3700 Non-Volatile Memory Controller
   22 r001i28s08     44  000b:02:00.0   298,2152-2182   Intel P3700 Non-Volatile Memory Controller
   22 r001i28s09     44  000b:03:00.0            2560   NVIDIA GK110BGL [Tesla K40m]
   23 r001i28s10     46  000c:01:00.0   320,2183-2213   Intel P3700 Non-Volatile Memory Controller
   23 r001i28s11     46  000c:02:00.0   322,2214-2244   Intel P3700 Non-Volatile Memory Controller
   23 r001i28s12     46  000c:03:00.0            2559   NVIDIA GK110BGL [Tesla K40m]
   28 r001i38s01     56  000d:01:00.0   344,2509-2524   LSI SAS2308 Fusion-MPT SAS-2
   28 r001i38s02     56  000d:02:00.0   352,2245-2275   Intel P3700 Non-Volatile Memory Controller
   28 r001i38s03     56  000d:03:00.0   354,2276-2306   Intel P3700 Non-Volatile Memory Controller
   29 r001i38s05     58  000e:02:00.0   376,2307-2337   Intel P3700 Non-Volatile Memory Controller
   29 r001i38s06     58  000e:03:00.0   378,2338-2368   Intel P3700 Non-Volatile Memory Controller
   30 r001i38s07     60  000f:01:00.0   392,2369-2399   Intel P3700 Non-Volatile Memory Controller
   30 r001i38s08     60  000f:02:00.0   394,2400-2430   Intel P3700 Non-Volatile Memory Controller
   31 r001i38s10     62  0010:01:00.0   416,2431-2461   Intel P3700 Non-Volatile Memory Controller
   31 r001i38s11     62  0010:02:00.0   418,2462-2492   Intel P3700 Non-Volatile Memory Controller
```

Example 4. The following `topology`(1) command uses the `-v` option, which includes interrupt count information.

```
uv300a-sys:~ # topology --irq -v

Index Location    NASID  PCI Address           IRQ(s)     INTCNT     Device
-------------------------------------------------------------------------------
    0 r001i01s00      0  0000:00:1f.2             519     703608    Intel SATA RAID Controller
```

```
    .          .      .  0000:02:00.0       1529-1532   11088420    Intel I210 Gigabit Network Connection
    .          .      .  0000:06:00.0             255          0    Matrox G200eR2
    4 r001i06s01      8  0001:01:00.0    56,1511-1526          0    LSI SAS2308 Fusion-MPT SAS-2
    4 r001i06s02      8  0001:02:00.0    64,1480-1510          0    Intel P3700 Non-Volatile Memory Controller
    4 r001i06s03      8  0001:03:00.0    66,1527,1533-1562           0    Intel P3700 Non-Volatile Memory Controller
    5 r001i06s05     10  0002:02:00.0    88,1563-1593          0    Intel P3700 Non-Volatile Memory Controller
    5 r001i06s06     10  0002:03:00.0    90,1594-1624          0    Intel P3700 Non-Volatile Memory Controller
    6 r001i06s07     12  0003:01:00.0   104,1625-1655          0    Intel P3700 Non-Volatile Memory Controller
.
.
.
```

Example 5. The following `topology`(1) command shows local CPU and node information for each device. You can use the output from this command to help you place applications close to their I/O device for better direct memory access performance.

```
uv300a-sys:~ # topology --io -v --nox

Index Location    NASID  PCI Address   Node    Local CPUS     Device
-------------------------------------------------------------------------
    0 r001i01s00      0  0000:00:1f.2     0    0-14,480-494   Intel SATA RAID Controller
    .          .      .  0000:02:00.0     0    0-14,480-494   Intel I210 Gigabit Network Connection
    .          .      .  0000:06:00.0     0    0-14,480-494   Matrox G200eR2
    4 r001i06s01      8  0001:01:00.0     4   60-74,540-554   LSI SAS2308 Fusion-MPT SAS-2
    4 r001i06s02      8  0001:02:00.0     4   60-74,540-554   Intel P3700 Non-Volatile Memory Controller
    4 r001i06s03      8  0001:03:00.0     4   60-74,540-554   Intel P3700 Non-Volatile Memory Controller
    5 r001i06s05     10  0002:02:00.0     5   75-89,555-569   Intel P3700 Non-Volatile Memory Controller
    5 r001i06s06     10  0002:03:00.0     5   75-89,555-569   Intel P3700 Non-Volatile Memory Controller
    6 r001i06s07     12  0003:01:00.0     6  90-104,570-584   Intel P3700 Non-Volatile Memory Controller
    6 r001i06s08     12  0003:02:00.0     6  90-104,570-584   Intel P3700 Non-Volatile Memory Controller
    7 r001i06s10     14  0004:01:00.0     7 105-119,585-599   Intel P3700 Non-Volatile Memory Controller
    7 r001i06s11     14  0004:02:00.0     7 105-119,585-599   Intel P3700 Non-Volatile Memory Controller
   12 r001i16s01     24  0005:01:00.0    12 180-194,660-674   LSI SAS2308 Fusion-MPT SAS-2
   12 r001i16s02     24  0005:02:00.0    12 180-194,660-674   Intel P3700 Non-Volatile Memory Controller
   12 r001i16s03     24  0005:03:00.0    12 180-194,660-674   Intel P3700 Non-Volatile Memory Controller
   13 r001i16s05     26  0006:02:00.0    13 195-209,675-689   Intel P3700 Non-Volatile Memory Controller
   13 r001i16s06     26  0006:03:00.0    13 195-209,675-689   Intel P3700 Non-Volatile Memory Controller
   14 r001i16s07     28  0007:01:00.0    14 210-224,690-704   Intel P3700 Non-Volatile Memory Controller
   14 r001i16s08     28  0007:02:00.0    14 210-224,690-704   Intel P3700 Non-Volatile Memory Controller
   15 r001i16s10     30  0008:01:00.0    15 225-239,705-719   Intel P3700 Non-Volatile Memory Controller
```

```
15 r001i16s11     30  0008:02:00.0    15 225-239,705-719   Intel P3700 Non-Volatile Memory Controller
20 r001i28s01     40  0009:01:00.0    20 300-314,780-794   NVIDIA GK110BGL [Tesla K40m]
```

Partitioning an SGI UV Server

This chapter includes the following topics:

About Partitioning

You can divide a single SGI UV server into multiple, distinct systems, each with its own console, root filesystem, and IP network address. Each of these software-defined groups of processor cores constitutes a *partition*. You can reboot, install software, power down, and upgrade each partition independently. The partitions can communicate with each other over an SGI NUMAlink connection, and this is called *cross-partition communication*. Collectively, all of these partitions compose a single, shared-memory cluster.

Partition discovery software allows all of the partitions to know about each other, and partition firewalls provide memory protection for each partition. The system software uses firewall code to open up a portion of memory so that it can be accessed by CPU cores in other partitions.

You can configure differently sized partitions. For example, you can configue a 128-processor system into four partitions of 32 CPU cores each, or you can configure the same 128–processor system into two partitions of 64 CPU cores each. The partition sizes and the number of partitions affect fault containment and scalability. For example, you might want to dedicate all 64 CPU cores of a system to a single large application during the night. During the day, you can partition the system into two 32–processor systems for separate and isolated use.

Note: The terms *single system image* (SSI) and *partition* can be used interchangeably, so you might see both used in SGI documentation.

The following sections provide more information about partitioning:

Partitioning Advantages

Partitioning provides the following capabilities:

- The ability to create a large, shared-memory cluster.

  You can use SGI's NUMAlink technology to create a very low latency, very large, shared-memory cluster for optimized use of Message Passing Interface (MPI) software and logically shared, distributed memory access (SHMEM) routines. MPI and SHMEM exploit the globally addressable, cache coherent, shared memory to deliver high performance.

- Fault containment.

  In most cases, you can bring down a single partition, for any reason, without affecting the rest of the system. Hardware memory protections prevent any unintentional accesses to physical memory on a different partition from reaching and corrupting that physical memory. For current fault containment caveats, see "Partitioning Limitations" on page 38.

Partitioning Limitations

Partitioning can increase the reliability of a system because power failures and other hardware errors can be contained within a particular partition. However, there still can be cases in which the whole shared memory cluster is affected. For example, this can occur during hardware upgrades that multiple partitions share.

If a partition is sharing its memory with other partitions, the loss of that partition may take down all other partitions that were accessing its memory. This is currently

possible when an MPI or SHMEM job is running across partitions using the XPC kernel module.

Failures are usually contained within a partition even when memory is being shared with other partitions. XPC is invoked using normal shutdown commands such as `reboot`(8) and `halt`(8) to ensure that all memory shared between partitions is revoked before the partition resets. This is also done if you use the `rmmod`(8) command to remove the XPC kernel modules. Unexpected failures such as kernel panics or hardware failures almost always force the affected partition into the kernel debugger or the crash dump utility. These tools also invoke XPC to revoke all memory shared between partitions before the partition resets. XPC cannot be invoked for unexpected failures such as power failures and spontaneous resets (not generated by the operating system), and thus all partitions sharing memory with the partition might also reset.

About Using the Message Passing Toolkit (MPT) on a Partitioned System

If you enable the XPC kernel module, you enable direct memory access between partitions, which is referred to as *global shared memory*. When XPC is enabled, processes in one partition can access physical memory located on another partition.

If the system issues the following message when your application runs, you need to enable the kernel modules:

```
MPT ERROR from do_cross_gets/xpmem_get, rc = -1, errno = 22
```

For more information about the benefits of global shared memory and how to enable the XPC kernel module, see the *SGI MPI and SGI SHMEM User Guide*.

Partitioning Requirements

The following topics describe the system requirements for partitioning SGI UV systems:

SGI UV 300 System Partitioning Requirements

An SGI UV 300 partition can include as few as one chassis or as many as eight chassis. Each individual partition needs to be equipped with the following:

- The infrastructure to run as a standalone system. This infrastructure includes a system disk and console connection.
- A base I/O BMC that belongs to the partition. The base I/O BMC cannot be shared by two partitions.

Peripherals, such as dual-ported disks, can be shared the same way two nodes in a cluster can share peripherals.

If you partition an SGI UV 300 system, note the following:

- SGI supports the `ipmitool` command on unpartitioned SGI UV 300 systems only. The `ipmitool` command does not function as expected on a partitioned SGI UV 300 system.
- The power commands are `power on`, `power off`, `power cycle`, and `power reset`. Power operations performed on one partition need to complete before you perform a power operation on another partition. Specifically, this means that you need to perform power operations in the following order:
  - If you type the `reset` command on one partition, wait for that partition to reset completely before you issue any power commands on a different partition.
  - If one partition is powered off, power that partition back on before you issue any power commands on a different partition.

SGI UV 2000 System Partitioning Requirements

An SGI UV 2000 system includes one or more individual rack units (IRUs). Each IRU is a 10U high enclosure that includes eight compute blades and one chassis management controller (CMC). For each partition, you can incude the following:

- As few as two and as many as 128 compute blades.
- A maximum of 2048 physical processor cores. When you enable Hyper-Threading, the maximum number of processor threads is 4096.

  For information about how to enable Hyper-Threading, see the *SGI UV CMC Software User Guide*.

The following list describes additional requirements for a single partition:

- Each partition must have the infrastructure to run as a standalone system. This infrastructure includes a system disk and console connection.
- An I/O blade belongs to the partition to which the attached IRU belongs. I/O blades cannot be shared by two partitions.
- Peripherals, such as dual-ported disks, can be shared the same way two nodes in a cluster can share peripherals.
- Partitions must be contiguous in the topology. For example, the route between any two nodes in the same partition must be contained within that partition and not route through any other partition. This allows intrapartition communication to be independent of other partitions.
- Partitions must be fully interconnected. That is, for any two partitions, there must be a direct route between those partitions without passing through a third. This is required to fulfill true isolation of a hardware or software fault to the partition in which it occurs.

SGI UV 1000 System Partitioning Requirements

An SGI UV 1000 system includes at least two and up to 128 compute blades (16 to 2048 cores). The following list describes additional requirements for a single partition:

- One base I/O blade per partition.
- As few as two and up to 128 compute blades.
- 16 to 2048 physical processor cores, for a maximum of 4096 threads with Hyper-Threading enabled.

Creating Partitions

The following topics explain how to create partitions:

- "Creating Partitions on an SGI UV 300 System" on page 42
- "Creating Partitions on an SGI UV 2000, SGI UV 1000, or SGI UV 100 System" on page 47

Creating Partitions on an SGI UV 300 System

The following procedure explains how to partition a system, and it includes a running example that divides a system into two partitions.

Note: SGI supports the `ipmitool` command on unpartitioned SGI UV 300 systems only. The `ipmitool` command does not function as expected on a partitioned SGI UV 300 system.

Procedure 6-1 To partition an SGI UV 300 system

1. Verify that SGI Performance Suite is installed on your SGI UV system.

   Partitioning requires the following products from the SGI Performance Suite: SGI Accelerate and SGI MPI. To verify that these products are installed, type one of the following commands:

   - On RHEL platforms, type the following command:

     ```
     # yum grouplist | grep SGI
        SGI Accelerate
        SGI Foundation Software
        SGI MPI
        SGI REACT
     ```

   - On SLES platforms, type the following command:

```
# zypper search -C -i SGI

Loading repository data...
Reading installed packages...

S | Name                               | Summary                               | Type
--+------------------------------------+---------------------------------------+--------
i | SGI-Accelerate                     | SGI Accelerate                        | pattern
i | SGI-Accelerate-trace-modules       | SGI Accelerate Trace Kernel Modules | pattern
i | SGI-MPI                            | SGI MPI                               | pattern
```

   The preceding example output shows that both the RHEL and SLES systems include the appropriate software packages.

2. Log into the rack management controller (RMC) as the root user.

For example:

```
# ssh root@uv-rmc
```

3. Type the following command to retrieve information about the SGI UV system:

```
> config -v

SSN: UV300-00000007

RMCs:            1
        r001i01c UV300


BMCs:            4
        r001i01b IP127-BASEIO
        r001i06b IP127-BASEIO     BASEIO-DISABLED
        r001i11b IP127
        r001i16b IP127


Partitions:      1
        partition000 BMCs:    4
```

Your goal is to determine the following:

- If the system is partitioned currently.
- The number of chassis and the number of chassis with base I/O risers. You need at least one chassis with one one base I/O riser per partition.

The preceding output shows the following:

- This system has four chassis in total. There are two chassis with base I/O risers. One is `r001i01b`, and the other is `r001i16b`. The only base I/O riser enabled at this time is on `r001i01b`.
- Only one base I/O riser can be enabled per partiton. This machine has only one partition at this time, so only one base I/O riser is enabled. This procedure shows how to enable the other base I/O riser and how to create two partitions.

4. Type the following command to retrieve information about the BASEIO_DISABLE variable on this SGI UV 300 system:

```
> hwcfg -av
BASEIO_DISABLE=no ..................................... 3/4 BMC(s)
        r001i01b
        r001i11b
        r001i16b
BASEIO_DISABLE=yes ..................................... 1/4 BMC(s)
        r001i06b
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION=0
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

5. Enable additional base I/O risers, as needed.

   In this procedure's example, there are two base I/O risers on the SGI UV system, but currently, one base I/O riser is disabled because the system is configured as a single, large partiton. Type the following command to enable the additional base I/O riser:

```
> hwcfg BASEIO_DISABLE=no r001i06b
BASEIO_DISABLE=default [no] <PENDING RESET>
BASEIO_DISABLE=no (this is the BMC default, override cleared)
```

6. Type the following command to verify the base I/O riser(s) that you enabled:

```
> hwcfg -av
BASEIO_DISABLE=default [no] <PENDING_RESET> ............... 1/4 BMC(s)
        r001i06b
```

```
BASEIO_DISABLE=no ..................................... 3/4 BMC(s)
        r001i01b
        r001i11b
        r001i16b
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION=0
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

In the previous output, the `<PENDING RESET>` line shows that base I/O riser `r001i06b` is set to be enabled. The actual enablement occcurs later in the procedure, after the power cycle.

7. Use the `hwcfg` command to create two or more partitions.

   The format for the command is as follows:

   `hwcfg PARTITION=`*partition_num BMC_num BMC_num* [*BMC_num* ...]

   The variables in the preceding command are as follows:

   - For *partition_num*, specify the number for one of the partitions. SGI recommends that you start partition numbers at `1`. For example, you can have partitions `1` and `2`, or you can have partitions `1` and `2` and `3`.

     You can use `0` for a partition number only if you want to configure the system as a single, large partition. A single partition should have all chassis in partition `0`.

   - For *BMC_num*, specify the BMC identifier for one or more chassis.

For example, to create two partitions, type the following commands:

```
> hwcfg PARTITION=1 r001i01b r001i11b
PARTITION=1 <PENDING RESET>
> hwcfg PARTITION=2 r001i06b r001i16b
PARTITION=2 <PENDING RESET>
```

It is customary to list the base I/O BMC first, as the preceding lines show, but it is not necessary.

8. Verify that the system registered the information from the preceding step's `hwcfg` commands.

   For example, type the following command and verify its output:

```
> hwcfg -av
BASEIO_DISABLE=default [no] <PENDING_RESET> ..... 1/4 BMC(s)
        r001i06b
BASEIO_DISABLE=no ..................................... 3/4 BMC(s)
        r001i01b
        r001i11b
        r001i16b
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION=1 <PENDING RESET> ............................. 2/4 BMC(s)
        r001i01b
        r001i11b
PARTITION=2 <PENDING RESET> ............................. 2/4 BMC(s)
        r001i06b
        r001i16b
SOCKET_DISABLE=no
        all targeted BMC(s)
```

The preceding output shows that the base I/O riser in the second chassis is ready to be enabled. The output also shows the four chassis associated with the two partitions you created.

9. Type the following command to power cycle the chassis:

   ```
   > power -o -c cycle
   ```

10. Wait for the power cycling to complete.

    On a large system, this could take several minutes.

11. Periodically, type the `power status` command to affirm that all the chassis are powered up.

    For example, the output from the following command shows that all chassis are powered up:

    ```
    > power status
    ==== r001i01c ====
    chassis - on: 4, off: 0, unknown: 0, disabled: 0
    ```

    After all the blades power up, proceed to the next step.

12. Use the `uvcon` command to access the shell.

    For example, to get to the shell for partition 1 or partition 2, type one of the following commands:

    ```
    > uvcon p1
    ```

    OR

    ```
    > uvcon p2
    ```

13. Proceed to the following:

    "Installing the Operating System on a Partition" on page 54

Creating Partitions on an SGI UV 2000, SGI UV 1000, or SGI UV 100 System

The following procedure explains how to partition a system, and it includes a running example that divides a system into two partitions.

Procedure 6-2 To partition an SGI UV 2000, SGI UV 1000, or SGI UV 100 system

1. Verify that SGI Performance Suite is installed on your SGI UV system.

   Partitioning requires the following products from the SGI Performance Suite: SGI Accelerate and SGI MPI. To verify that these products are installed, type one of the following commands:

   - On RHEL platforms, type the following command:

     ```
     # yum grouplist | grep SGI
        SGI Accelerate
        SGI Foundation Software
        SGI MPI
        SGI REACT
     ```

   - On SLES platforms, type the following command:

```
# zypper search -C -i SGI

Loading repository data...
Reading installed packages...

S | Name                               | Summary                               | Type
--+------------------------------------+---------------------------------------+--------
i | SGI-Accelerate                     | SGI Accelerate                        | pattern
i | SGI-Accelerate-trace-modules       | SGI Accelerate Trace Kernel Modules | pattern
i | SGI-MPI                            | SGI MPI                               | pattern
```

   The preceding example output shows that both the RHEL and SLES systems include the appropriate software packages.

2. Log into the chassis management controller (CMC) as the root user.

   For example:

   ```
   # ssh root@uv-cmc
   ```

3. Type the following command to retrieve information about the SGI UV system:

   ```
   > config -v

   SSN: UV2-00000048

   CMCs:            1
   ```

```
        r001i01c UV2000

BMCs:           8
        r001i01b00 IP109-BASEIO
        r001i01b01 IP109
        r001i01b02 IP109
        r001i01b03 IP109
        r001i01b04 IP109-BASEIO     IORISER-DISABLED
        r001i01b05 IP109
        r001i01b06 IP109
        r001i01b07 IP109

Partitions:     1
        partition000 BMCs:    8

Accessories:    0
```

Your goal is to determine the following:

- If the system is partitioned currently.
- The number of blades, base I/O risers, and base I/O BMCs that are available. You need at least one compute blade with a base I/O riser per partition.

The preceding output shows the following:

- This system has eight compute blades. There are two blades that are base I/O BMCs: one is r001i01b00, and the other is r001i01b04. Only base I/O BMC r001i01b00 is enabled at this time.
- Only one base I/O riser can be enabled per partiton. This machine has only one partition at this time, so only one base I/O riser is enabled. This procedure shows how to enable the other base I/O riser and how to create two partitions.

4. Type the following command to retrieve information about the SGI UV system:

```
> hwcfg -av
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=no ..................................... 7/8 BMC(s)
```

```
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
        r001i01b05
        r001i01b06
        r001i01b07
IORISER_DISABLE=yes ..................................... 1/8 BMC(s)
        r001i01b04
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=0
       all targeted BMC(s)
ROUTER_TYPE=ordinary
       all targeted BMC(s)
SOCKET_DISABLE=no
       all targeted BMC(s)
```

Your goal is to verify the IORISER_DISABLE and PARTITION variables.

5. Enable additional base I/O risers, as needed.

   In this procedure's example, there are two base I/O risers on the SGI UV system, but currently, one base I/O riser is disabled because the system is configured as a single, large partiton. Type the following command to enable the additional base I/O riser:

```
> hwcfg IORISER_DISABLE=no r001i01b04
IORISER_DISABLE=default [no] <PENDING RESET>
IORISER_DISABLE=no (this is the BMC default, override cleared)
```

6. Type the following command to verify the base I/O riser(s) that you enabled:

```
> hwcfg -av
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=default [no] <PENDING RESET> ..... 1/8 BMC(s)
        r001i01b04
IORISER_DISABLE=no ....................................... 7/8 BMC(s)
```

```
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
        r001i01b05
        r001i01b06
        r001i01b07
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=0
        all targeted BMC(s)
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

In the previous output, the `<PENDING RESET>` line shows that base I/O riser `r001i01b04` is set to be enabled. The actual enablement occcurs later in the procedure, after the power cycle.

7. Use the `hwcfg` command to create two or more partitions.

   The format for the command is as follows:

   `hwcfg PARTITION=`*partition_num BMC_num BMC_num* `[`*BMC_num* `...]`

   The variables in the preceding command are as follows:

   - For *partition_num*, specify the number for one of the partitions. SGI recommends that you start partition numbers at `1`. For example, you can have partitions `1` and `2`, or you can have partitions `1` and `2` and `3`.

     You can use `0` for a partition number only if you want to configure the system as a single, large partition. A single partition should have all blades in partition `0`.

   - For *BMC_num*, specify the BMC identifier for one or more blades.

   For example, to create two partitions, type the following commands:

```
> hwcfg PARTITION=1 r001i01b00 r001i01b01 r001i01b02 r001i01b03
PARTITION=1 <PENDING RESET>
> hwcfg PARTITION=2 r001i01b04 r001i01b05 r001i01b06 r001i01b07
PARTITION=2 <PENDING RESET>
```

It is customary to list the base I/O BMC first, as the preceding lines show, but it is not necessary.

8. Verify that the system registered the information from the preceding step's `hwcfg` commands.

   For example, type the following command and verify its output:

```
> hwcfg -av
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=default [no] <PENDING RESET> ..... 1/8 BMC(s)
        r001i01b04
IORISER_DISABLE=no ..................................... 7/8 BMC(s)
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
        r001i01b05
        r001i01b06
        r001i01b07
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=1 <PENDING RESET> ............................ 4/8 BMC(s)
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
PARTITION=2 <PENDING RESET> ............................ 4/8 BMC(s)
        r001i01b04
        r001i01b05
        r001i01b06
        r001i01b07
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

The preceding output shows the second base I/O Riser is ready to be enabled. The output also shows the eight blades associated with the two partitions you created.

9. Type the following command to power cycle the blades:

   ```
   > power -o -c cycle
   ```

10. Wait for the power cycling to complete.

    On a large system, this could take several minutes.

11. Periodically, type the following command to affirm that all the blades are powered up:

    ```
    > power status
    ```

    The following example output shows that all blades are powered up:

    ```
    ==== r001i01c (PRI) ====
    blades - on: 8, off: 0, unknown: 0, disabled: 0
    chassis - on: 1, off: 0
    accio - on: 0, off: 0
    ```

    After all the blades power up, proceed to the next step.

12. Use the uvcon command to access the shell.

    For example, to get to the shell for partition 1 or partition 2, type one of the following commands:

    ```
    > uvcon p1
    ```

    OR

    ```
    > uvcon p2
    ```

13. Proceed to the following:

    "Installing the Operating System on a Partition" on page 54

Installing the Operating System on a Partition

The procedure you use to install an operating system on an individual partition is the same procedure that you use to install an operating system as a single-system image on an entire SGI UV system. The following platform-specific information describes prerequisites:

- If you have an SGI UV 300 system, your system includes a DVD drive. You need to obtain the software installation media.
- If you have an SGI UV 2000, SGI UV 1000, or SGI UV 100 system, the installation procedure assumes that you have a system management node (SMN) attached to your SGI UV system. If you do not have an SMN, you need the software media and an extra DVD drive. You need to plug the DVD drive into each base I/O blade of the partition and install the operating system manually.

The following procedure explains how to install a copy of the operating system and the SGI Foundation Software on a partition.

Procedure 6-3 To install software on a partition

1. Notify users that they need to log off.

   The software installation process includes a step to reboot the system. If any users are logged in at this time, their sessions end abruptly.

2. Log into the rack management controller (RMC) or the chassis management controller (CMC) as the root user.

   Example 1. On an SGI UV 300 system, type the following command, and specify the hostname of the RMC for *hostname*:

   # **ssh root@*hostname***

   Example 2. On an SGI UV 2000, SGI UV 1000, or SGI UV 100 system, type the following command:

   # **ssh root@uv-cmc**

3. Type the `config -v` command to retrieve the name for the base I/O BMC for each partition.

   Example 1. On an SGI UV 300, type the following command:

   ```
   uv300-rmc-rmc RMC:r001i01c> config -v
   ```

```
SSN: UV300-00000021

RMCs:            1
        r001i01c UV300


BMCs:            4
        r001i01b IP123-BASEIO    P001
        r001i06b IP123-BASEIO    P001  BASEIO-DISABLED
        r001i11b IP123-BASEIO    P002
        r001i16b IP123-BASEIO    P002  BASEIO-DISABLED


Partitions:      2
        partition001 BMCs:    2
        partition002 BMCs:    2
```

The preceding output shows two partitions, `P001` and `P002`. The lines that contain `BASEIO` show the base I/O BMC for each partition. Note the chassis names for the base I/O BMCs for each partition, which are `r001i06b` for `P001` and `r001i16b` for `P002`.

Example 2. On an SGI UV 2000, SGI UV 1000, or SGI UV 100 system, type the following command:

```
uv2000-cmc CMC:r001i01c> config -v

SSN:UV2-00000048

CMCs:             1
         r001i01c UV2000

BMCs:             8
         r001i01b00 IP109-BASEIO    P001
         r001i01b01 IP109           P001
         r001i01b02 IP109           P001
         r001i01b03 IP109           P001
         r001i01b04 IP109-BASEIO    P002
         r001i01b05 IP109           P002
         r001i01b06 IP109           P002
         r001i01b07 IP109           P002
```

```
Partitions:      2
         partition001 BMCs:    4
         partition002 BMCs:    4

Accessories:     0
```

The rightmost column in the preceding example output, which contains P*xxx* entries, shows two partitions, `P001` and `P002`. The lines that contain `BASEIO` show the base I/O BMC for each partition. Note the names for the base I/O BMCs for each partition, which are `r001i01b00` for `P001` and `r001i01b04` for `P002`.

4. Create a connection to the SGI UV server.

   On SGI UV 300 systems, you create this connection to the chassis that contains the base I/O riser.

   On SGI UV 2000, SGI UV 1000, and SGI UV 100 systems, you create this connection to the base I/O blade of the partition.

   For information about how to connect to the SGI UV server, see the following:

   SGI UV System Software Installation and Configuration Guide

5. Install the operating system.

   For information about how to install the operating system on the partition, see the following:

   SGI UV System Software Installation and Configuration Guide

---

Note: The procedures in the *SGI UV System Software Installation and Configuration Guide* contain some conditional statements that pertain to either an SGI UV that you want to install as large, single-system image or to a partition image, so make sure to note these conditional statements when you perform the installation.

Disabling Partitions

The following topics explain how to disable partitions:

- "Disabling Partitions on an SGI UV 300 System" on page 57

- "Disabling Partitions on an SGI UV 2000, SGI UV 1000, or SGI UV 100 System" on page 60

Disabling Partitions on an SGI UV 300 System

The following procedure explains how to remove the partitions from a partitioned system, and it includes a running example that changes a system from a two-partition system to a one-partition system.

Procedure 6-4 To remove partitions

1. Log into the rack management controller (RMC) as the root user.

   For example:

   # **ssh root@uv-rmc**

2. Type the following command to retrieve the system's current partitioning information:

```
> hwcfg -av
BASEIO_DISABLE=no
        all targeted BMC(s)
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION=1 ..................................................... 2/4 BMC(s)
        r001i01b
        r001i11b
PARTITION=2 ..................................................... 2/4 BMC(s)
        r001i06b
        r001i16b
SOCKET_DISABLE=no
        all targeted BMC(s)
```

The preceding output shows two partitions.

3. Remove all the partitions you configured.

   It is possible to remove only one partition, but subsequent system operations can be confusing. To make operations more straightforward, remove all partitions if you want to remove one.

   To remove a partition, type the `hwcfg` command in the following format:

   `hwcfg -c PARTITION=`*id*

   For *id*, type the partition identifier as shown in the previous step's `hwcfg -av` output.

   For example, to remove the two partitons :

   > **hwcfg -c PARTITION**

4. Type the following command to verify that the partition(s) that you want to remove are set to be removed:

```
> hwcfg -av
BASEIO_DISABLE=no
        all targeted BMC(s)
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION= default [0] <PENDING RESET>
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

The preceding output shows `PARTITION=default [0] <PENDING RESET>`, which means that the BMCs are set for the change when the next power cycle occurs.

5. Disable the base I/O risers that are associated with the partitions you want to remove.

   This step is important because the SGI UV system cannot be reset if it is configured to have only one partition but it is configured to have multiple chassis with base I/O risers that are enabled.

   Use the `hwcfg` command in the following format to disable the base I/O risers:

   `hwcfg BASEIO_DISABLE=yes` *id*

   For *id*, specify the identifier of the chassis with the base I/O riser that you want to disable.

   For example:

```
> hwcfg BASEIO_DISABLE=yes r001i06b
BASEIO_DISABLE=yes <PENDING RESET>
```

6. Type the `hwcfg -av` command again to retrieve the system status.

   For example:

```
> hwcfg -av
BASEIO_DISABLE=no ..................................... 3/4 BMC(s)
        r001i01b
        r001i11b
        r001i16b
BASEIO_DISABLE=yes <PENDING RESET> ..................... 1/4 BMC(s)
        r001i06b
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
CHASSIS_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IIO_DISABLE=no
        all targeted BMC(s)
MEMRISER_DISABLE=no
        all targeted BMC(s)
```

```
NL_ENABLE=yes
        all targeted BMC(s)
PARTITION=default [0] <PENDING RESET>
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

7. Type the following command to power cycle the blades:

   > **power -o -c cycle**

8. Wait for the power cycling to complete.

   On a large system, this could take several minutes.

9. Periodically, type the `power status` command to affirm that all the blades are powered up.

   For example, the output from the following command shows that all chassis are powered up:

   ```
   > power status
   ==== r001i01c ====
   chassis - on: 4, off: 0, unknown: 0, disabled: 0
   ```

Disabling Partitions on an SGI UV 2000, SGI UV 1000, or SGI UV 100 System

The following procedure explains how to remove the partitions from a partitioned system, and it includes a running example that changes a system from a two-partition system to a one-partition system.

Procedure 6-5 To remove partitions

1. Log into the chassis management controller (CMC) as the root user.

   For example:

   # **ssh root@uv-cmc**

2. Type the following command to retrieve the system's current partitioning information:

   ```
   > hwcfg -av
   BIOS_FILE=/common/bios.fd
           all targeted BMC(s)
   ```

```
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=no
        all targeted BMC(s)
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=1 ..................................................... 4/8 BMC(s)
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
PARTITION=2 ..................................................... 4/8 BMC(s)
        r001i01b04
        r001i01b05
        r001i01b06
        r001i01b07
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

The preceding output shows two partitions.

3. Remove all the partitions you configured.

   It is possible to remove only one partition, but subsequent system operations can be confusing. To make operations more straightforward, remove all partitions if you want to remove one.

   To remove a partition, type the `hwcfg` command in the following format:

   `hwcfg -c` *partition_id*

   For *partition_id*, type the partition identifier as shown in the previous step's `hwcfg -av` output.

   For example, to remove the two partitons :

   ```
   > hwcfg -c PARTITION
   ```

4. Type the following command to verify that the partition(s) that you want to remove are set to be removed:

```
> hwcfg -av
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=no
        all targeted BMC(s)
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=default [0] <PENDING RESET>
        all targeted BMC(s)
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

   The preceding output shows `PARTITION=default [0] <PENDING RESET>`, which means that the BMCs are set for the change when the next power cycle occurs.

5. Disable the base I/O risers that are associated with the partitions you want to remove.

   This step is important because the SGI UV system cannot be reset if it is configured to have only one partition but it is configured to have multiple base I/O riser blades with each blade's base I/O riser enabled.

   Use the `hwcfg` command in the following format to disable the base I/O risers:

   `hwcfg IORISER_DISABLE=yes` *blade_id*

   For example:

```
> hwcfg IORISER_DISABLE=yes r001i01b04
IORISER_DISABLE=yes <PENDING RESET>
```

6. Type the `hwcfg -av` command again to retrieve the system status.

For example:

```
> hwcfg -av
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x0
        all targeted BMC(s)
IORISER_DISABLE=no ..................................... 7/8 BMC(s)
        r001i01b00
        r001i01b01
        r001i01b02
        r001i01b03
        r001i01b05
        r001i01b06
        r001i01b07
IORISER_DISABLE=yes <PENDING RESET> ..................... 1/8 BMC(s)
        r001i01b04
NL6_ENABLE=yes
        all targeted BMC(s)
PARTITION=default [0] <PENDING RESET>
        all targeted BMC(s)
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

7. Type the following command to power cycle the blades:

   ```
   > power -o cycle
   ```

8. Wait for the power cycling to complete.

   On a large system, this could take several minutes.

9. Periodically, type the following command to affirm that all the blades are powered up:

   ```
   > power status
   ```

The following example output shows that all blades are powered up:

```
==== r001i01c (PRI) ====
blades - on: 8, off: 0, unknown: 0, disabled: 0
chassis - on: 1, off: 0
accio - on: 0, off: 0
```

Enabling Remote Services

SGI Remote Services provides secure, proactive, 24 x 7 support monitoring. Whether you are focused on HPC solutions or SGI® UV™ for SAP HANA®, our secure remote services can help your business run smoothly.

By allowing us to connect with your system, access to key configuration and diagnostic information is available to SGI Customer Support. The result can be significantly reduced time to issue resolution. Additional benefits include greater operational efficiency, maximized service levels, and improved uptime and productivity.

SGI enables basic Remote Services features by default. For more information, see the SFS release notes or the following websites:

- https://support.sgi.com/content_request/220019/index.html
- www.sgi.com/remoteservices

You can send questions to `remoteservices@sgi.com`.

Fixing Broken Weak Updates Links

This chapter includes the following topics:

About Weak Updates Links

When you install new kernel packages, either from SGI or from one of the operating system distributors, the kernel update package RPM installer creates a link to the existing kernel module files that are provided by other RPMs. These links to existing packages are called *weak-updates* links.

During some installations, the installer fails to create all the weak-updates links. The `sgi-upgrade-utils` package contains the following commands that can create links to existing kernel module files:

- The `sgi-pre-upgrade` command. Use this command before you apply a kernel upgrade. It repairs existing weak-updates links that are broken. SGI supports this command only on RHEL platforms.
- The `sgi-post-upgrade` command. Use this command if weak-updates links become broken during an upgrade or if the installer fails to create weak-updates links.

You can use the commands in the `sgi-upgrade-utils` package to fix weak-updates links that result from the following types of upgrades:

- An operating system kernel upgrade
- An SGI kernel modules upgrade

For example, assume that your system has installed the following SGI kernel module RPM:

```
sgi-hwperf-kmp-default-1.0_3.0.76_0.11-sgi710r3.sles11sp3
```

On the system at this time, kernel `sgi-hwperf-kmp-default-1.0_3.0.76_0.11-sgi710r3.sles11sp3` includes the `hwperf.ko` kernel module in the following directory:

```
/lib/modules/3.0.76-0.11-default/updates/hwperf.ko
```

Assume that you want to update your kernel. Specifically, you want to install the following new kernel RPM:

```
kernel-default-3.0.101-0.15.1.x86_64.rpm
```

When you install the new kernel, the installer creates directories for new kernel modules in the following directory:

```
/lib/modules/3.0.101-0.15-default/dirs
```

The new kernel RPM does not include an update to kernel module `.../hwperf.ko`, so the new kernel does not contain new software for that module. When the new kernel installs, a script creates a link from the new kernel's modules directory to the old `hwperf.ko` kernel module that resides in `/lib/modules/3.0.76-0.11-default/updates/hwperf.ko`.

The following command shows the link after a correct installation:

```
# ls -l /lib/modules/3.0.101-0.15-default/weak-updates/updates/hwperf.ko
lrwxrwxrwx 1 root root 50 Apr 7 2014 /lib/modules/3.0.101-0.15-default/weak-updates/updates/hwperf.ko
-> /lib/modules/3.0.76-0.11-default/updates/hwperf.ko
```

If the installer fails to create a weak-updates link, or if the installer breaks an existing weak-updates link, you can use the `sgi-post-upgrade` command to to fix the link.

Using the `sgi-upgrade-utils` Package Tools

The following procedure explains how to run the `sgi-pre-upgrade` command and the `sgi-post-upgrade` command.

Procedure 8-1 To fix broken weak-updates links

1. Make sure that the SGI upgrade repositories are in place.
2. (Conditional — RHEL platforms only) Test and repair the existing weak-updates links.

   Complete this step only if you want to repair existing weak-updates links.

Type the following command:

/usr/sbin/sgi-pre-upgrade [-r]

Use the `-r` parameter if you want to perform a test run of this command. When you include the `-r` parameter, no actions are taken, and you can examine the output to see how the command will operate on your system.

3. Install the new kernel.
4. (Optional) Type the following command, in a test run, to repeat the addition of all kernel modules:

   # **/usr/sbin/sgi-post-upgrade -1**

   This command's output lists the SGI kernel module RPMs that would be changed.

   This command does not actually perform the re-addition of all the kernel modules.

   Examine the output from this command to ensure it is as expected before you continue to the next step in this procedure.
5. Type the following command to repeat the addition of all kernel modules:

   # **/usr/sbin/sgi-post-upgrade -r -1**

   Note that the preceding command includes the `-r` parameter, which runs the command.
6. (Optional) Type the following command, in a test run, to display the directories in which the kernel releases reside:

   # **/usr/sbin/sgi-post-upgrade -1** ***kernel_release***

   For *kernel_release*, specify the release string identifier of the kernel you just installed. For example:

   # **/usr/sbin/sgi-post-upgrade -1 3.0.101-0.46-default**

   This command's output lists the available kernel releases available in `/lib/modules`.

   This command does not actually destroy and recreate the links.

   Examine the output from this command to ensure it is as expected before you continue to the next step in this procedure.

If necessary, type the following command to retrieve a list of kernel releases to specify for the *kernel_release* argument to the `sgi-post-upgrade` command:

```
# ls /lib/modules
```

7. Type the following command to destroy, and then recreate, all kernel modules' weak-updates links in *kernel_release*:

   ```
   # /usr/sbin/sgi-post-upgrade -r -1 kernel_release
   ```

   For *kernel_release*, specify the release directory of the kernel.

   Note that the preceding command includes the `-r` parameter, which runs the command.

For more information about the `sgi-pre-upgrade` command and the `sgi-post-upgrade` command, display each command's help output. To retrieve the help output, type the command name with the `-h` parameter.

Additional SFS Utilities

This appendix section includes information about additional SFS commands and utilities that typically require no user involvement. SGI technical support staff members might guide you in the use of these commands when troubleshooting or tuning.

These utilities are as follows:

- The online diagnostics commands `field_diags_licensed_ice-x86`(1) and `field_diags_licensed_xe_x86_cluster`(1). These diagnostic utilities are optional, but SGI systems that ship from the factory include these utilities by default.
- The `sgi-base-configuration` package. This is a collection of configuration scripts for SGI UV systems and SGI Rackable systems.
- The `sgi-ha-stonith-plugins-uv` RPM provides STONITH agents to implement fencing on SGI UV systems through the RMC or CMC.
- The `sgi_irqbalance`(8) utility controls interrupt requests (IRQ) affinity on SGI UV systems and SGI Rackable systems. The daemon starts when a system boots. If a device generates IRQs, `sgi_irqbalance` attempts to distribute the interrupts to the CPUs that are on the same chassis (or node) upon which the interrupt originated.

  By default, this utility starts every two minutes. To change that interval, set `SGI_IRQBALANCE_SLEEPTIME` in `/etc/sysconfig/sgi_irqbalance` to the desired number of seconds and restart `sgi_irqbalance` or reboot your system.

  The `sgi-base-configuration` package configures the `sgi_irqbalance`(8) utility. On SGI UV systems, `sgi-base-configuration` configures `sgi_irqbalance`(8) automatically. On SGI Rackable systems, you have the option to use the operating system's IRQ balancer. To use the operating system version, edit the `X86_DISABLE_SGI_IRQBALANCE` environment variable in `/etc/sysconfig/x86config`.

  For more information, see the `sgi_irqbalance`(8) man page.
- The `sgi-kdump` RPM replaces the RHEL 6 and SLES 11 distributions' standard `crash`(8), `makedumpfile`(8), and `kexec`(8) binaries with binaries that support SGI UV large-memory systems. SGI UV systems that run RHEL 7 and SLES 12 use

the standard `crash`(8), `makedumpfile`(8), and `kexec`(8) binaries included in the operating system distributions.

- The `sgtools` package enables you to interact with SCSI storage devices, such as disks, JBOD enclosures, and CD/DVD drives, using the Linux SCSI generic driver. For more information about these tools, see the following website:

  http://sg.danny.cz/sg/tools.html

Index