SGI® UV™ CMC Software User Guide

007–5636–006

New Features in this Guide

This revision includes the following changes:

- Support for SGI® UV™ 2000. See the following:
  - Figure 2-1 on page 5
  - "SGI UV 2000 CMC Ethernet Ports" on page 53
  - "SGI UV 2000 Diagrams" on page 54
- Information about the following commands:
  - The `hel` command to access the hardware error logs. See "`hel`" on page 43.
  - The `leds` command to display system LED values. See "`leds`" on page 45.

Record of Revision

Version	Description
001	June 2010 Initial release.
002	June 2010 Added information for SGI Altix UV 100 systems.
003	October 2010 Updated to support the SGI Foundation Software 2.2 release.
004	February 2011 Updated to support SGI Foundation Software 2.3 release.
005	April 2011 Updated to support SGI Foundation Software 2.4 release.
006	November 2012 Updated for distribution with the SGI Foundation Software 2.7 release.

Contents

Tables

About This Guide

You can use the chassis manager controller (CMC) commands to monitor and manage the following SGI® UV™ systems:

- SGI UV 2000 systems
- SGI UV 1000 systems
- SGI UV 100 systems

You can issue CMC commands from the CMC itself, or you can issue CMC commands from the system management node (SMN). The SGI UV 2000 and SGI UV 1000 systems include an SMN. The SGI UV 100 system can include an SMN as an option. If your SGI UV system includes an SMN, you have an additional option for system management, which is the SGI Management Center (SMC) graphical user interface. The SMC interface runs on an SMN.

This guide describes how to use the CMC commands to monitor and manage SGI UV systems. This guide assumes that you are familiar with the information in the SGI UV system hardware guides.

Note: The SGI UV CMC commands described in this manual do not apply to SGI UV 10 systems or SGI UV 20 systems.

Related Publications

The hardware guides contain information about SGI UV system components, safety and regulatory specifications, setting up and operating the system, powering up and powering down the system, and basic troubleshooting. The hardware guides are as follows:

- *SGI UV 2000 System User Guide*
- *SGI Altix UV 1000 System User's Guide*
- *SGI Altix UV 100 System User's Guide*

The following publications contain additional information that is related to CMC operations:

- *SGI Performance Suite x.x Start Here*
- *SGI Foundation Software x.x Start Here*
- *SGI UV Systems Linux Configuration and Operations Guide*
- *SGI UV System Management Node Administrator Guide*
- *SGI Management Center (SMC) Installation and Configuration*
- *SGI Management Center (SMC) System Administrator Guide*

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 release notes on your system by accessing the README file for the product. This is usually located in the `/usr/share/doc/`*productname* directory, although file locations can vary.

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

Conventions

The following conventions are used throughout this publication:

Convention	Meaning
[ ]	Brackets enclose optional portions of a command or directive line.
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.

...	Ellipses indicate that a preceding element can be repeated.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names.
`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.
variable	Italic typeface denotes variable entries and words or concepts being defined.

Reader Comments

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

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

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

- Send mail to the following address:

  SGI
  Technical Publications
  46600 Landing Parkway
  Fremont, CA 94538

SGI values your comments and will respond to them promptly.

Overview

This chapter includes the following topics:

System Management Component Overview

SGI® UV™ system components provide a single system control network. The network is the control point for system power up, initialization, booting, and maintenance. The components are as follows:

- The baseboard management controller (BMC) on the node boards. This is a standard component for all SGI UV systems.
- The chassis management controller (CMC) in the rear of the individual rack unit (IRU). This is a standard component for all SGI UV systems. This manual describes the CMC commands.
- The system management node (SMN). This is a standard component on SGI UV 2000 and SGI UV 1000 systems. This component is optional on SGI UV 100 systems. The SMN is a gateway between the SGI UV system control network and the other networks at your site.

  The SGI Management Center (SMC) graphical user interface runs only on the SMN. You can configure, operate, and monitor your SGI UV system through the SMC interface. On systems with an SMN, you can use the SMC software as an alternative to, or in addition to, the CMC commands. For more information, see the *SGI Management Center System Administrator Guide*.

Note: The system control network is a private, closed network. Do not reconfigure it from the standard SGI UV installation. Do not directly connect it to any other network. The SGI UV system control network does not accommodate additional network traffic, routing, address naming other than its own schema, and DCHP controls other than its own configuration. The system control network is not security hardened and is not tolerant of heavy network traffic, so it is vulnerable to denial-of-service attacks.

Chassis Management Controller (CMC) Overview

The CMC performs several functions. Many functions are common across both IRUs and routers, but some functions are specific to the type of enclosure. The CMC functions are as follows:

- Manages power control and sequencing
- Provides environmental control and monitoring
- Initiates system resets
- Stores identification and configuration information
- Provides a console interface for diagnostics and scans

The CMC accepts direction from the SMN and sends operational requests to the BMC on each compute blade upon request. A CMC can communicate with the blade BMCs and other CMCs in an IRU if they are linked together as a single system image (SSI), also called a *partition*.

On SGI UV 2000 systems, a port from each CMC connects to a dedicated Ethernet switch that provides a synchronous clock signal to all of the CMCs and the SSI. On SGI UV 1000 systems and on SGI UV 100 systems, a port from each CMC connects the CMCs together in an ethernet string.

Each CMC shares its information with the SMN as well as other CMCs within the SSI. The SMN, optional mass storage units, and PCIe expansion enclosures do not have a CMC installed.

Chassis Management Controller Tasks

You can use chassis management controller (CMC) commands to perform the following tasks:

Accessing the CMC on an SGI UV 2000 System Through a Serial Connection

You can use the procedure in this topic to connect to the CMC on an SGI UV 2000 system if the following conditions exist:

- You do not know the address of the CMC and there is no SMN.

- The SMN is down or unavailable.

The following procedure explains how to establish a serial connection from a dumb terminal to a CMC in an SGI UV 2000 system.

Procedure 2-1 To establish a serial connection to an SGI UV 2000 system

1. Locate the CMC on the SGI UV 2000 system.

   Figure 2-1 on page 5 shows the location of the CMC.

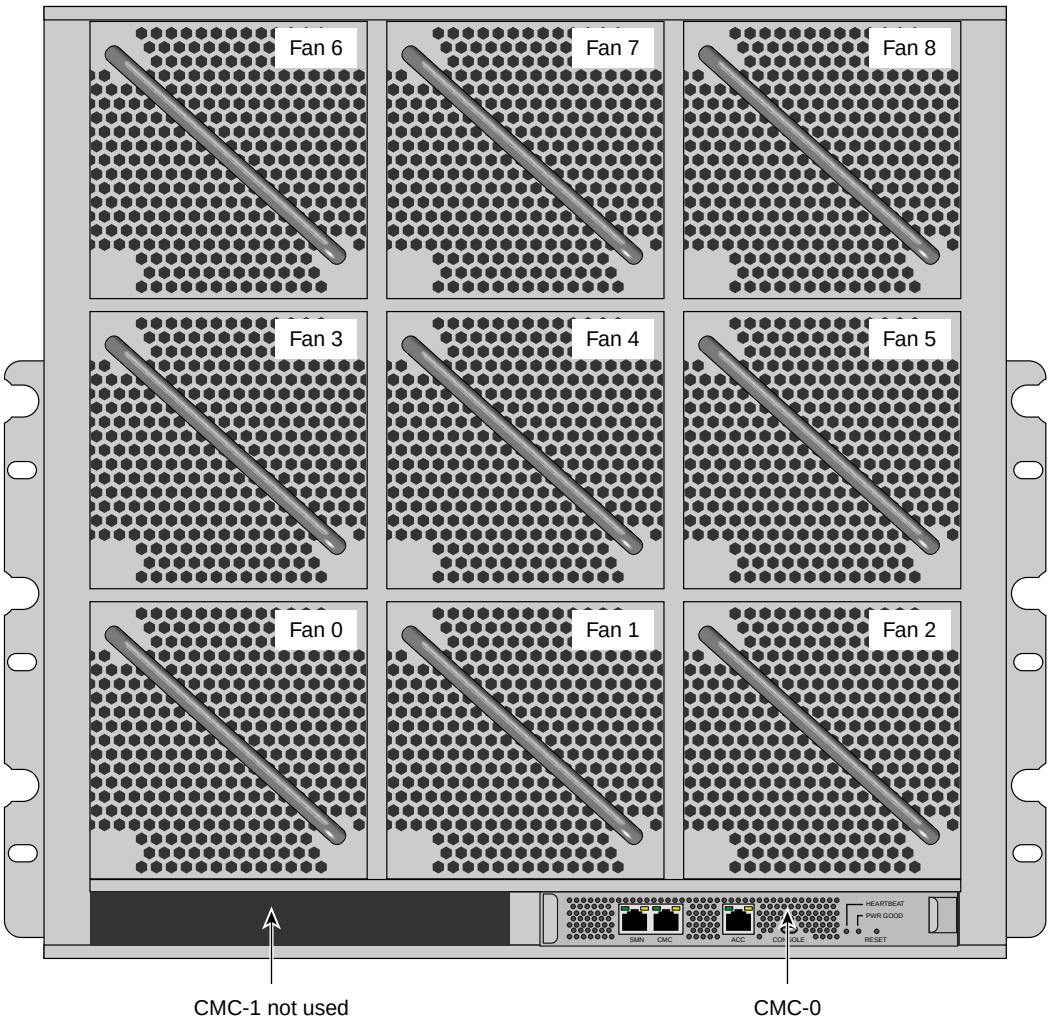

Figure 2-1 SGI UV 2000 CMC Location

2. Use a micro-USB serial cable to connect the terminal to the CONSOLE port connector on the CMC board of the IRU.

   Typically, you connect a serial console to the first (bottom) IRU in any single rack configuration. Figure 2-2 on page 6 shows the CONSOLE port.

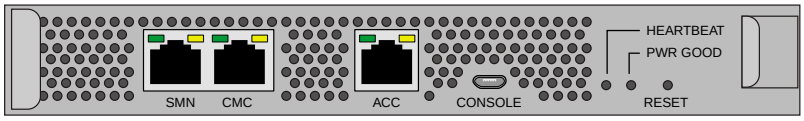

Figure 2-2 SGI UV 2000 Ports

3. Set the terminal to the following functional modes:
   - Baud rate of 115,200
   - 8 data bits
   - One stop bit, no parity
   - No hardware flow control (RTS/CTS)

Accessing the CMC on an SGI UV 1000 System Through a Serial Connection

You can use the procedure in this topic to connect to the CMC on an SGI UV 1000 system if the following conditions exist:

- You do not know the address of the CMC and there is no SMN.
- The SMN is down or unavailable.

The following procedure explains how to establish a serial connection from a dumb terminal to a CMC in an SGI UV 1000 system.

Procedure 2-2 To establish a serial connection to an SGI UV 1000 system

1. Locate the CMC on the SGI UV 1000 system.

   Figure 2-3 on page 7 shows the location of the CMC.

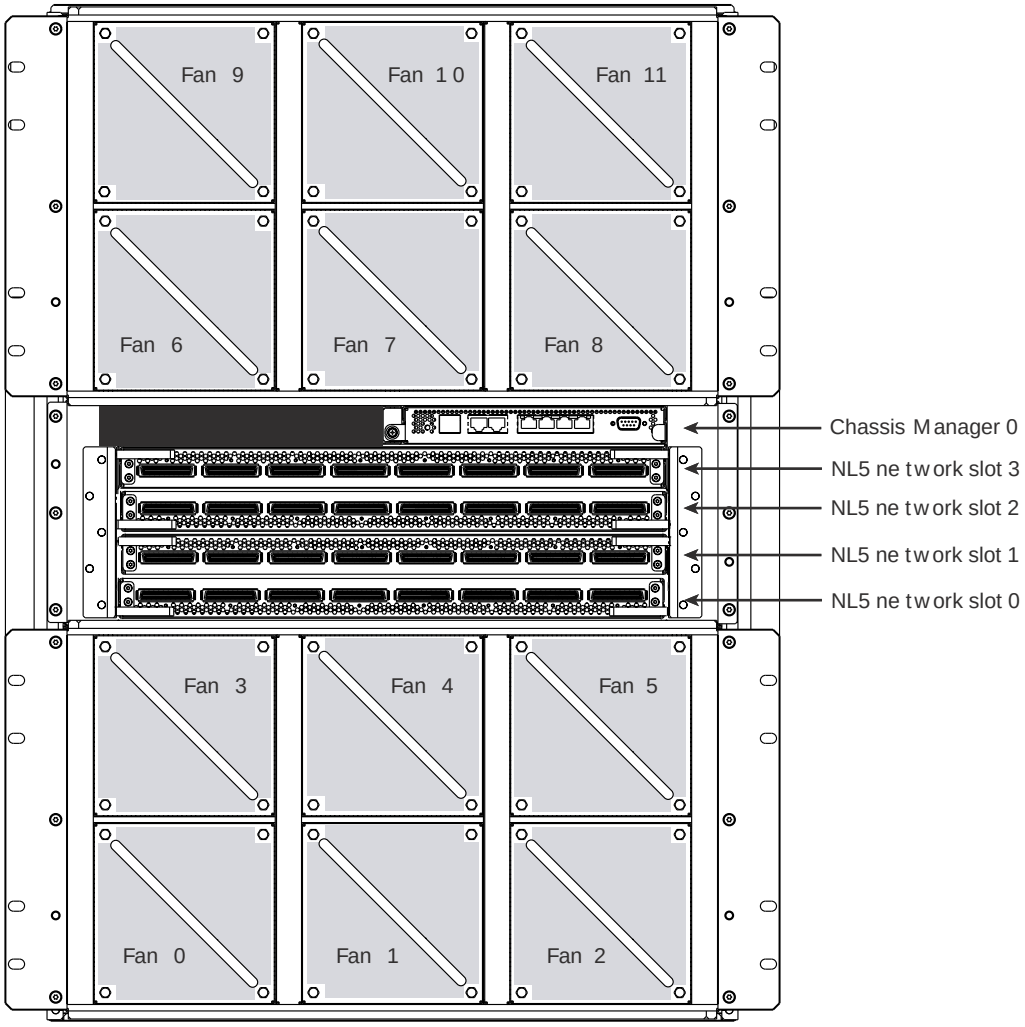

Figure 2-3 SGI UV 1000 CMC Location

2. Use a serial cable to connect the terminal to the (DB-9) RS-232-style console port connector on the CMC.

   Typically, you connect a serial console to the first (bottom) IRU in any single rack configuration. Figure 2-4 on page 8 shows the `CONSOLE` port.

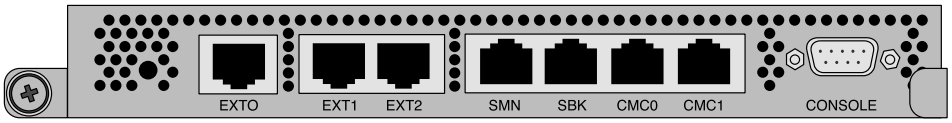

Figure 2-4 SGI UV 1000 Ports

3. Set the terminal to the following functional modes:
   - Baud rate of 115,200
   - 8 data bits
   - One stop bit, no parity
   - No hardware flow control (RTS/CTS)

Accessing the CMC on an SGI UV 100 System Through a Serial Connection

You can use the procedure in this topic to connect to the CMC on an SGI UV 100 system if the following conditions exist:

- You do not know the address of the CMC and there is no SMN.
- The SMN is down or unavailable.

The following procedure explains how to establish a serial connection from a dumb terminal to a CMC in an SGI UV 100 system.

Procedure 2-3 To establish a serial connection to an SGI UV 100 system

1. Locate the CMC and the `Console` port on the SGI UV 100 system.

   Figure 2-5 on page 9 shows the location of the CMC and the port.

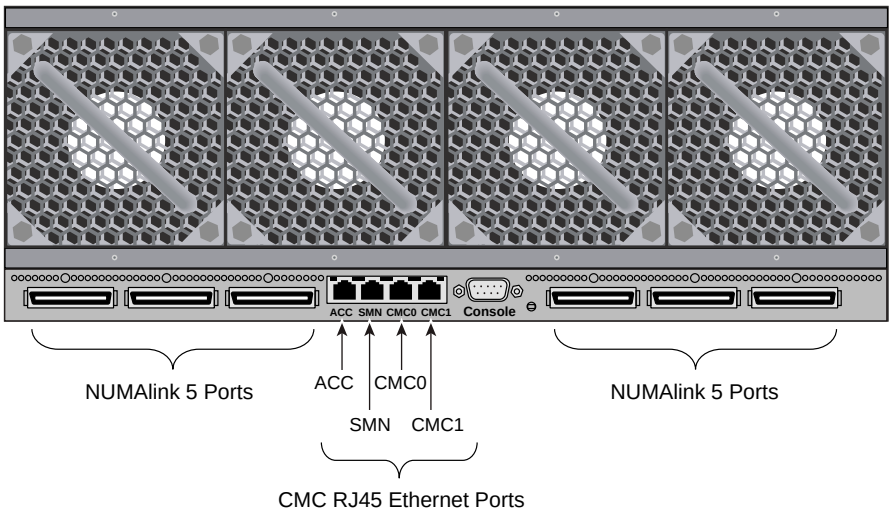

Figure 2-5 SGI UV 100 System CMC and Ports

2. Use a serial cable to connect the terminal to the (DB-9) RS-232-style console port connector on the CMC.

   Typically, you connect a serial console to the first (bottom) IRU in any single rack configuration.

3. Set the terminal to the following functional modes:

   - Baud rate of 115,200
   - 8 data bits
   - One stop bit, no parity
   - No hardware flow control (RTS/CTS)

Accessing the CMC Through a Network Connection and Logging In

Each SGI UV 2000 system or SGI UV 1000 system has only one SMN, but each of these systems can have more than one CMC. When there are multiple CMCs, some might be attached to the SMN, and some might not be attached to the SMN. In a

troubleshooting situation, your SMN might be down. An SMN is optional on an SGI UV 100 System.

If an SMN is present, the CMC is configured to request an IP address from the SMN via dynamic host configuration protocol (DHCP). This is the default behavior.

If an SMN is not present, the CMC might be configured with a static IP address.

The following notes pertain to the procedures in this topic:

- If your SGI UV system is configured to use `telnet`(1), you can use use either `ssh`(1) or `telnet`(1) to log in. The procedures and examples generally use only the `ssh`(1) command.
- The CMC password is always `root`. Do not change this password.
- You can log into the CMC by using the CMC's IP address or by using the CMC's hostname. If you want to use the CMC's hostname to log in, make sure to set the hostname in the `/etc/hosts` file on your PC or workstation. The procedures and examples generally use the IP address.

The following procedures explain how to establish a network connection and log in to a CMC when an SMN is available and when an SMN is not available.

Procedure 2-4 To connect to a CMC from an SMN

1. Use the `ssh`(1) command to log in to the SMN as the root user.

   Obtain the SMN's address from your system administrator or the network administrator.

   For example:

   `# ssh root@`*ip_address*

2. Type the `cmclist` command to retrieve the IP addresses of the CMCs that are attached to this SMN.

   The SMN acts as both a DNS and a DHCP server to the CMCs.

   For example, the following output shows one CMC, and the CMC's IP address is `172.19.1.1`:

```
[root@uv48-smn]# cmclist
# SYSCO discovered CMC(s)
# Hostname                IP Address
s UV-00000048:r001i01c     172.19.1.1
```

3. (Optional) Type the `config -v` command to retrieve the identifiers for other CMCs attached to this SMN.

   For example, the following output shows 4 CMCs in the SGI UV system, but only one is attached to the SMN:

   ```
   [root@uv48-smn]# config -v
   SSN: UV-00000048

   CMCs:           16
    r001i01c UV1000 SMN
    r001i23c UV1000
    r001q42c UVQCR
    r001q43c UVQCR
   ...
   output truncated for inclusion in this manual
   ```

   The preceding output identifies the CMCs on this system according to rack number and IRU. For example, `r001i01c` refers to rack `001`, IRU `01`, and `c` identifies a CMC.

4. Use the `ssh`(1) command to connect to the CMC with the IP address shown in the `cmclist` output.

   For example:

   ```
   [root@uv48-smn]# ssh 172.19.1.1
   ```

   The root user is the only user configured on the CMC, so log in as `root`. Type `root` for the CMC password.

The following procedure explains how to establish a network connection to a CMC without logging into an SMN first. Use this procedure if there is no SMN, if the SMN is down, or if you want to log into a CMC from another CMC.

Procedure 2-5 To connect to a CMC without an SMN

1. Determine the IP address of the CMC to which you want to connect.

   Use one of the following methods:

   - Obtain the CMC's address from your system administrator or network administrator.

     If you do not have the CMC's address, you need to make a serial connection to the CMC. See one of the following topics:

- "Accessing the CMC on an SGI UV 2000 System Through a Serial Connection" on page 3
- "Accessing the CMC on an SGI UV 1000 System Through a Serial Connection" on page 6
- "Accessing the CMC on an SGI UV 100 System Through a Serial Connection" on page 8

2. Use the `ssh`(1) command to log in to the CMC.

   For example:

   `# ssh` *ip_address*

   The root user is the only user configured on the CMC, so log in as `root`. Type `root` for the CMC password.

Powering On and Booting an SGI UV System From a Complete Power Off

The following procedure explains how to power on an SGI UV system from a complete power off.

Procedure 2-6 To power on or boot an SGI UV system

1. Visually inspect the system and make sure that the power breakers are on.
2. Establish a connection to the CMC and log in.

   Use one of the following procedures:

   - "Accessing the CMC on an SGI UV 2000 System Through a Serial Connection" on page 3
   - "Accessing the CMC on an SGI UV 1000 System Through a Serial Connection" on page 6
   - "Accessing the CMC on an SGI UV 100 System Through a Serial Connection" on page 8
   - "Accessing the CMC Through a Network Connection and Logging In" on page 9

3. Type the `power on` command to power up the SGI UV system.

For example:

```
CMC:r001i01c> power on
```

4. Type the `console` command to open a second window and monitor the power-on process.

   Depending upon the size of your system, it can take 5 to 10 minutes for the SGI UV system to power on.

5. When the power-on process completes, type `CTRL-] q` to exit the console.

6. When the `shell>` prompt appears, type `fs0:` to access the boot partition.

   For example:

   ```
   shell> fs0:
   ```

7. Boot the system.

   Use one of the following commands:

   - On Red Hat Enterprise Linux CMCs, type the following command:

     ```
     fs0:\> EFI\redhat\grub
     ```

   - On SUSE Linux Enterprise Server CMCs, type the following command:

     ```
     fs0:\> \efi\SuSE\elilo
     ```

Powering Off an SGI UV System

The following procedure explains how to power off an SGI UV system.

Procedure 2-7 To power off an SGI UV system

1. Type the `power off` command to power off the SGI UV system.

   For example:

   ```
   CMC:r1i1c> power off
   ==== r001i01c (PRI) ====
   ```

2. Type the `power status` command to verify that the system has powered down.

For example:

```
CMC:r1i1c> power status
==== r001i01c (PRI) ====
on: 0, off: 32, unknown: 0, disabled: 0
```

Access the Kernel Debugger

To send a nonmaskable interrupt (NMI) signal and invoke the kernel debugger (KDB), enter the following:

```
CMC:r1i1c> power nmi
Entering kdb (current=0xffff8aa3fe11c040, pid 0) on processor 7 due to NonMaskable Interrupt @ 0xffffffff8100ad42
     r15 = 0x0000000000000000      r14 = 0x0000000000000000
     r13 = 0x0000000000000000      r12 = 0x0000000000000000
      bp = 0xffffffff81927380       bx = 0xffff8ac1ff11dfd8
     r11 = 0xffffffff8101a2c0      r10 = 0xffff88000beefd18
      r9 = 0x00000000ffffffff       r8 = 0x0000000000000000
      ax = 0x0000000000000000       cx = 0x0000000000000000
      dx = 0x0000000000000000       si = 0xffff8ac1ff11dfd8
      di = 0xffffffff81a2b308  orig_ax = 0xffffffffffffffff
      ip = 0xffffffff8100ad42       cs = 0x0000000000000010
   flags = 0x0000000000000246       sp = 0xffff88000bee7ff0
      ss = 0x0000000000000018 &regs = 0xffff88000bee7f58
[7]kdb>
```

The `kdb>` prompt indicates that the KDB is available.

Enable the Auto-Power Capability

The auto-power capability allows your system to power up automatically when power is applied after a power outage:

- To determine if the auto-power capability is enabled on your system, use the `autopower` command without options. For example, the following output shows that the feature is currently disabled:

  ```
  uv44-cmc CMC:r001i01c> autopower
  ==== r001i01c (PRI) ====
  auto-power on is disabled
  ```

- To enable auto-power, use the `-e` option:

```
uv44-cmc CMC:r001i01c> autopower -e
==== r001i01c (PRI) ====
auto-power on enabled (120 second delay)
```

- To disable auto-power, use the `-d` option:

```
uv44-cmc CMC:r001i01c> autopower -d
==== r001i01c (PRI) ====
auto-power on disabled
```

The default delay before a `power on` command is issued after auxiliary power is supplied to CMCs and BMCs is two minutes. This allows for the system controller configuration to stabilize (complete CMC and BMC detection).

Only the lowest numbered (rack and u-position) CMC initiates the power-on sequence. Because this is a CMC-based feature, the maximum size of the supported system configuration supported is 16 or fewer racks.

Note: On SGI UV 1000 and SGI UV 100 systems, the auto-power capability requires CMC 1.2.7 or later firmware. You can use the `version` command from the CMC prompt to determine the CMC firmware version, as follows:

```
uv44-cmc CMC:r001i01c> version
SGI Chassis Manager Controller, Firmware Rev. 1.3.16   [Bootloader 0.6.0]
```

Flashing the CMC 1.2.7 or later firmware only adds support for the auto-power capability; it does not enable it. To enable it, type the `autopower -e` command at the CMC prompt after the system controller configuration has stabilized.

For more information, see "`autopower`" on page 33.

View the System Configuration

To view your system configuration, enter the following:

```
CMC:r1i1c> config -v
```

Example output on an SGI UV 2000 system:

```
CMC:r001i01c> config -v

SSN: UV2-00000082

CMCs:            2
 r001i01c UV2000
 r001i11c UV2000

BMCs:           16
 r001i01b00 IP109-BASEIO
 r001i01b01 IP109
 r001i01b02 IP109
 r001i01b03 IP109
 r001i01b04 IP109
 r001i01b05 IP109
 r001i01b06 IP109
 r001i01b07 IP109
 r001i11b00 IP109-BASEIO     IORISER-DISABLED
 r001i11b01 IP109
 r001i11b02 IP109
 r001i11b03 IP109
 r001i11b04 IP109
 r001i11b05 IP109
 r001i11b06 IP109
 r001i11b07 IP109

Partitions:      1
 partition000 BMCs:   16

Accessories:     0
```

Example output on an SGI UV 1000 system:

```
CMC:r1i1c> config -v

CMCs:            2
        r001i01c UV1000
        r001i02c UV1000

BMCs:           32
        r001i01b00 IP93-BASEIO
        r001i01b01 IP93-DISK
```

```
        r001i01b02 IP93-EXTPCIE
        r001i01b03 IP93-EXTPCIE
        r001i01b04 IP93
        r001i01b05 IP93
        r001i01b06 IP93
        r001i01b07 IP93
        r001i01b08 IP93
        r001i01b09 IP93
        r001i01b10 IP93
        r001i01b11 IP93
        r001i01b12 IP93
        r001i01b13 IP93
        r001i01b14 IP93
        r001i01b15 IP93
        r001i02b00 IP93-BASEIO
        r001i02b01 IP93-EXTPCIE
        r001i02b02 IP93-DISK
        r001i02b03 IP93-EXTPCIE
        r001i02b04 IP93-EXTPCIE
        r001i02b05 IP93-EXTPCIE
        r001i02b06 IP93-EXTPCIE
        r001i02b07 IP93-EXTPCIE
        r001i02b08 IP93-INTPCIE
        r001i02b09 IP93-INTPCIE
        r001i02b10 IP93-INTPCIE
        r001i02b11 IP93-INTPCIE
        r001i02b12 IP93-INTPCIE
        r001i02b13 IP93-INTPCIE
        r001i02b14 IP93-INTPCIE
        r001i02b15 IP93-INTPCIE

Partitions:      1
        partition000 BMCs:   32
```

Note: `r001i01b00` refers to rack 0, IRU 1, and blade 0 (see Figure A-4 on page 59).

Set Hardware Configuration Overrides

The `hwcfg` command lets you view and set hardware configuration overrides.

Note: Many of the hardware overrides available on early SGI UV systems are no longer necessary due to firmware enhancements and hardware changes. Most of these overrides are still available but have been suppressed. To expose them, use the `-h` or `--hidden` flag.

- To see a list of current override settings, use the `hwcfg` command without options. This shows all overrides set on any blades in the system. If any overrides are set on some blades and not on others, the output shows a count of blades where the override is set. For example:

```
CMC:r001i01c> hwcfg
DEBUG_SW=0x4
IORISER_DISABLE=yes ..................................... 1/2 BMC(s)
NL6_ENABLE=0x808
```

- To see a list of blades where each override is set, use the `-v` option. For example:

```
harp10-cmc CMC:r001i01c> hwcfg -v
DEBUG_SW=0x4
        all targeted BMC(s)
IORISER_DISABLE=yes ..................................... 1/2 BMC(s)
        r001i02b00
NL6_ENABLE=0x808
        all targeted BMC(s)
```

- To see an individual list of blades and their overrides, use the `-vv` option. For example:

```
harp10-cmc CMC:r001i01c> hwcfg -vv
==== r001i01b00 ====
DEBUG_SW=0x4
NL6_ENABLE=0x808
==== r001i02b00 ====
IORISER_DISABLE=yes
DEBUG_SW=0x4
NL6_ENABLE=0x808
```

- To set one or more overrides, use the following command:

  `hwcfg` *name=value* [*name=value* ...]

  For example:

```
CMC:r001i01c> hwcfg DEBUG_SW=0x4
```

- To clear overrides, use the `-c` option. For example:

```
CMC:r001i01c> hwcfg -c
```

- To clear all overrides, including hidden overrides, use both the `-c` and `-a` options.

```
CMC:r001i01c> hwcfg -c -a
```

- To clear one or more specific variables, use the following command:

```
hwcfg -c name [... name ]
```

- To show a list of `hwcfg` variables available, use the `--list` option.

Example for an SGI UV 2000 system:

```
CMC:r001i01c> hwcfg -l
==== 16/16 BMC(s) ====
SOCKET_DISABLE=yes|no| Partition number for this blade

MAX_CORES= Maximum number of cores allowed (per socket)

BLADE_DISABLE=yes|no
 Disable this blade

IORISER_DISABLE=yes|no
 Disable the I/O riser on this blade

DEBUG_SW=<32-bit value>
 Software debug switches  (see "hwcfg --help DEBUG_SW" for details)

BIOS_FILE=Alternate BIOS image file

NL6_ENABLE=yes|no| Enable NL6 links

ROUTER_TYPE=ordinary|repeater|meta0|meta1|meta2|meta3
 Override the default NL6R Router Type
```

Example for an SGI UV 1000 system:

```
CMC:r1i1c> hwcfg --list
==== 4/4 BMC(s) ====
```

```
SOCKET_DISABLE=yes|no|<socket bitmask>
        Socket 0 disable

PARTITION=<numeric value 0-65535>
        Partition number for this blade

SMT_ENABLE=yes|no
        SMT (HyperThread) enable

MAX_CORES=<numeric value 0-255, 0=no limit>
        Maximum number of cores allowed (per node)

BLADE_DISABLE=yes|no
        Disable this blade

IORISER_DISABLE=yes|no
        Disable the I/O riser on this blade

ICH_DISABLE=yes|no
        Disable ICH10 on this BaseIO

DEBUG_SW=<32-bit value>
        Software debug switches  (see "hwcfg --help DEBUG_SW" for details)

HUB_CORE_SPEED=320|367|375|383|400
        Clock frequency of the HUB

NL5_ENABLE=yes|no|<NL5 port bitmask>
        Enable NL5 links

NL5_RATE=1.25|2.5|3.125|5.0|6.25
        NL5 transfer rate

NL5_CABLE_ENABLE=yes|no
        Enable cabled NL5 links

NL5_NEAR_LB=yes|no
        Configure NL5 channels in near loopback

NL5_SCRAMBLE=yes|no|<NL5 port bitmask>
        Enable scramble mode on NL5 links
```

```
NL5_HUB2_WAR=yes|no
        Enable the NL5 PHY/BIST war for Hub2.0
```

- To show all variables, regardless of state, use the `--all` option. For example, for an SGI UV 1000 system:

```
CMC:r1i1c> hwcfg --all
BLADE_DISABLE=no
DEBUG_SW=0x0
HUB_CORE_SPEED=375
ICH_DISABLE=no
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes
NL5_ENABLE=yes
NL5_HUB2_WAR=yes
NL5_NEAR_LB=no
NL5_RATE=6.25
NL5_SCRAMBLE=0x0
PARTITION=0
SMT_ENABLE=no
SOCKET_DISABLE=no
```

- To show a list of hidden hardware overrides:

```
uv44-cmc:~ # hwcfg --hidden --all
```

For example, for SGI UV 2000:

```
CMC:r001i01c> hwcfg --hidden --all
BACKPLANE_TYPE=default
BIOS_FILE=/common/bios.fd
BLADE_DISABLE=no
DEBUG_SW=0x4
IORISER_DISABLE=no ..................................... 1/2 BMC(s)
IORISER_DISABLE=yes .................................... 1/2 BMC(s)
MAX_CORES=0
NL6_ENABLE=0x808
PARTITION=0
ROUTER_TYPE=ordinary
SOCKET_DISABLE=no
```

```
{HIDDEN} BIOS_HOLD=no
{HIDDEN} BMCINIT0=
{HIDDEN} BMCINIT1=
{HIDDEN} BMC_RESET_DBG=0x0
{HIDDEN} BOOTMODE=QPI
{HIDDEN} CLOCK_MUX_SPREAD_SPECTRUM=no
{HIDDEN} CONSOLE_TO_DB9=no
{HIDDEN} FORCE_LOCAL_CLOCK=no
{HIDDEN} HARP_MEM_REPAIR_DISABLE=no
{HIDDEN} HARP_RESET_OPTS=
{HIDDEN} HARP_VDD_MARGIN=default
{HIDDEN} HUB_CORE_SPEED=375
{HIDDEN} HUB_DISABLE=no
{HIDDEN} HUB_XDP=no
{HIDDEN} IORISER_FORCE=no
{HIDDEN} IO_XDP=no
{HIDDEN} NL6INIT_OPTS=
{HIDDEN} NL6_8B10B=no
{HIDDEN} NL6_NEAR_LB=no
{HIDDEN} NL6_RATE=12.5
{HIDDEN} NL6_SPICO_FILE=
{HIDDEN} NL6_TUNE_ENABLE=yes
{HIDDEN} NL6_TX_EMPHASIS=0x0
{HIDDEN} QPI_SPEED=8.0
{HIDDEN} QPI_SPICO_FILE=
{HIDDEN} QPI_TUNE_ENABLE=yes
{HIDDEN} SNBCFG_OPTS=
```

Upgrade System BIOS

To upgrade the compute blade basic input/output system (BIOS), perform the following steps:

1. Display the current PROM level:

   ```
   CMC:r1i1c> bios
   ```

For example, for SGI UV 1000:

```
CMC:r1i1c> bios
Flashed on Sat May  1 14:14:45 UTC 2010 was bios.latest.fd (20100429_1603)
```

2. Get the newest PROM image from Supportfolio™ Online at http://support.sgi.com/

Note: Upgrading to SGI UV BIOS 2.00 release (or later) from a pre-2.00 SGI UV BIOS release requires the set-up variables to be cleared using either of the following CMC commands:

```
flashbios -e
power -c reset
```

These commands also clear any site-specific settings, such as M/N values and Extensible Firmware Interface (EFI) boot menu changes. These site-specific settings must be reconfigured at the 2.00 boot menu.

---

3. Copy the latest BIOS to a directory on the CMC in `/work/bmc/common/`. For example:

```
CMC:r1i1c> ls
bios.latest.fd flashbios
```

4. Use the `flashbios` command to rewrite (or *flash*) the compute blade BIOS. For example:

```
CMC:r1i1c> flashbios
Using default bios: bios.latest.fd
Checking processor status on all nodes....
Done. System is read for BIOS flash update
Flashing bios bios.lastest.fd (20100429_1603) This will take several minutes.
...
```

The following firmware flashing commands are available on the SMN for flashing an entire SGI UV system:

`flashcmc`
`flashbmc`
`flashiobmc`

For more information on how to use these commands, see the section about updating firmware in the *SGI UV Systems Linux Configuration and Operations Guide.*

Adding an Accessory

SGI supports the following CMC accessories:

- Magma™ PCI Express® PCIe Expansion chassis

  Power control only affects I/O accessories, such as this accessory. When the IRU chassis power is turned on, off, or cycled, the accessories are also turned on, off, or cycled. If necessary, use the `noio` option on the CMC `power on|off|cycle` commands to exclude the I/O accessories from the power operation.

- Eaton ePDU® enclosure-based power distribution unit

The following procedure explains how to add an accessory to an SGI UV CMC.

Procedure 2-8 To connect an accessory

1. Use the accessory's documentation to configure the accessory to use DHCP.

   When in use, the DHCP server on the CMC assigns an IP address to the accessory.

2. (Optional) Use the accessory's documentation to specify the accessory's physical location.

   The CMC uses simple network management protocol (SNMP). Specify the location information in the `sysLocation.0` string.

3. Connect a cable from the accessory's SNMP port to an open accessory jack on the CMC.

   The accessory jacks are labeled as follows:

   - On an SGI UV 2000 system, the accessory jack is labeled `ACC`. For a diagram that shows this jack, see the following:

     Figure 2-2 on page 6

   - On an SGI UV 1000 system or an SGI UV 100 system, the accessory jacks are labeled `EXT0`, `EXT1`, and `EXT2`. For diagrams that show these jacks, see the following:

     Figure 2-4 on page 8

     Figure 2-5 on page 9

   If you have more accessories than available jacks, use an external switch.

4. Type the `config -v` command to display information about the accessory.

After you attach the accessory, the CMC queries the accessory's SMNP `sysName.0` object identifier (OID) to determine the accessory type. The content of the OID is assumed to identify the accessory, as follows:

- `Magma Chassis` is assumed to be a Magma PCIe expansion chassis.
- `Monitored ePDU` is assumed to be an Eaton ePDU.

The CMC's DCHP server assigned an IP address to the accessory in the range of `10.`*RACK*`.`*UPOS*`.100` — `10.`*RACK*`.`*upos*`.199` range, where *rack* and *UPOS* are the rack and u-position of the CMC. This is the CMC's `VACC` virtual local area network (VLAN).

For example:

```
uv14-cmc CMC:r1i1c>  config -v
  CMCs:            1
      r001i01c UV1000
  BMCs:            4
      r001i01b00 IP93-BASEIO
      r001i01b01 IP93-DISK
      r001i01b02 IP93
      r001i01b03 IP93
  Partitions:      1
      partition000 BMCs:    4
  Accessories:     1
      undefined      10.1.1.101 (Magma PCIE Expansion)
```

5. Specify the accessory's location.

   To specify the location, type the `config` command in the following format:

   `config -acc` *ip_addr@rack.upos*

   The arguments are as follows:

ip_addr	Specify the IP address of the accessory.
rack	Specify the rack number. Racks are numbered sequentially with a three-digit number starting at `001`. A rack contains IRU enclosures. In a single compute rack system, the rack number is always `001`.
upos	For an Eaton ePDU unit, specify `0` for left or `1` for right.

For a Magma PCI Express PCIe Expansion chassis, specify the individual rack unit (IRU) position. IRU locations within a rack are identified by the bottom unit in which the IRU resides. For example, in a 42U rack, specify `1` as the *upos* for an IRU in the range U01 through U10.

For example:

```
uv14-cmc CMC:r1i1c> config --acc 10.1.1.101@1.30
  ==== r001i01c (PRI) ====
  10.1.1.101 (Magma Chassis) configured as r001u30io
```

6. Use the `config -v` command to verify the accessory's location.

   For example:

```
uv14-cmc CMC:r1i1c>  config -v
  CMCs:            1
      r001i01c UV1000
  BMCs:            4
      r001i01b00 IP93-BASEIO
      r001i01b01 IP93-DISK
      r001i01b02 IP93
      r001i01b03 IP93
  Partitions:      1
      partition000 BMCs:    4
  Accessories:     1
      r001u30io        10.1.1.101 (Magma PCIE Expansion)
```

   The preceding output shows `r001u30io` as the location for the accessory. For an Eaton ePDU, the format is r*rack*`pdu0` or r*rack*`pdu1`. For a Magma PCIe Expansion chassis, the format is r*rack*u*upos*`io`.

Enable Hyper-Threading

Threading in a software application splits instructions into multiple streams so that multiple processors can act on them.

Intel® Hyper-threading (HT) technology provides thread-level parallelism on each processor, resulting in more efficient use of processor resources, higher processing throughput, and improved performance. One physical CPU can appear as two logical

CPUs by having additional registers to overlap two instruction streams or a single processor can have dual-cores executing instructions in parallel.

- To enable HT, enter the following:

  ```
  CMC:r001i01c> hwcfg SMT_ENABLE=yes
  ```

- To disable HT, enter the following:

  ```
  CMC:r001i01c> hwcfg SMT_ENABLE=no
  ```

For more information about using HT, see the information about using cpusets with hyper-threads in the *Linux Resource Administration Guide*.

Chassis Management Controller (CMC) Command Reference

This chapter discusses the following:

CMC Commands Overview

The chassis management controller (CMC) commands perform the following functions:

- Control and monitor individual rack unit (IRU) and router fan speeds
- Read system identification (ID) PROMs
- Monitor voltage levels and reports failures
- Monitor and controls warning LEDs on the enclosure
- Create multiple system partitions running their own operating systems
- Flash system BIOS

You can use the CMC commands from the CMC command line or from the SGI Management Node (SMN) command line. The commands typically use the following format:

command [*options*] [*targets*]

The commands are located in `/sysco/bin` on both the SMN and CMC. For a list of available commands, enter `help` at the SMN/CMC prompt. For example:

```
CMC:r001i01c> /sysco/bin/help
available commands are:

        auth       authenticate SSN/APPWT change
        autopower  autopower status/management
        bios       perform bios actions
        bmc        access BMC shell
        cmc        access CMC shell
        config     show system configuration
        console    access system consoles
        help       list available commands
        hel        access hardware error logs
        hwcfg      access hardware configuration variable
        leds       display system LED values
        log        display system controller logs
        power      access power control/status
        uvcon      access system consoles
```

Note: The `uvcon` command is an alias to the `console` command.

Most of the commands (`bios`, `bmc`, `cmc`, `config`, `hwcfg`, `log`, `power`, and `console`) are available at both the CMC prompt (as user **`root`**) and at the SMN prompt (as user **`sysco`**).

For options available with each command, type:

command `--help`

Chassis Management Controller (CMC) and Baseboard Management Controller (BMC) Command Targets

Many CMC commands accept a `TARGET` argument that you can use to direct the command to operate on a group of CMCs or BMCs. The syntax lets you specify one target, multiple targets, or a range of targets. If you do not specify a target, the command operates on all BMCs or CMCs, as appropriate for a command.

The table in this topic shows how to specify groups of targets. You might need to quote some `TARGET`s to avoid wild card expansion by the SMN/CMC shell. The table uses the following notation:

Notation	Meaning
Any	In the table, “Any” refers to all racks, all U-positions, or all slots. “Any” also appears if you can use the command for all BMC types and for all CMC types.
upos	The U-position of the component in the rack.
rack	The rack in which the component is housed.
*	All values in that position. For example, `r*` references all racks, and `i*` references all IRUs. You can use a decimal number to narrow the target selection in place of any asterisk. The commands do not support ranges, but you can specify a space-separated list of targets.
BMC Type	One of the following: • Any. Applies to all BMC types. • IRUCOMP. On an SGI UV 2000 or SGI UV 1000, IRUCOMP describes IRU computer blade slots 0 through 15. On an SGI UV 100, IRUCOMP describes computer blade slots 0 through 1.

- IRUIO. IRU computer blades with an IO riser. On an SGI UV 2000 or SGI UV 1000, IRUIO describes IRU computer blade slots 0 through 15. On an SGI UV 100, computer blade slots 0 through 2.
- IRURTR. IRU router blades (router slots 0 through 3).
- QCRRTR. Quad Compact Router (QCR) router blades (slots 0 through 3).

CMC Type — One of the following:

- Any. Applies to all CMC types.
- QCR. Quad Compact Router.
- IRU. Applies to an SGI UV 2000, an SGI UV 1000, or SGI UV 100 system.

Table 3-1 CMC Command TARGET Specifications

TARGET Syntax	BMC *rack*, *upos*, *slot* — BMC Type	CMC *rack*, *upos* — CMC Type
`*, all`	Any, Any, Any — Any	Any, Any — Any
`r*`	*rack*, Any, Any — Any	*rack*, Any — Any
`r*i*`	*rack*, *upos*, Any — IRURTR, IRUCOMP	*rack*, *upos* — IRU
`r*q*`	*rack*, *upos*, Any — QCRRTR	*rack*, *upos* — QCR
`r*i*b*`	*rack*, *upos*, *slot* — IRUCOMP	N / A
`r*i*r*`	*rack*, *upos*, *slot* — IRURTR	N / A
`r*q*r*`	*rack*, *upos*, *slot* — QCRRTR	N / A
`*c`	N / A	Any, Any — Any
`r*i*c`	N / A	*rack*, *upos* — IRU
`r*q*c`	N / A	*rack*, *upos* — QCR
`allb`, `allc`	Any, Any, Any — IRUCOMP	N / A
`allbi`, `allci`	Any, Any, Any — IRUIO	N / A
`allr`	Any, Any, Any — IRURTR, QCRRTR	N / A

TARGET Syntax	BMC *rack, upos, slot* — BMC Type	CMC *rack, upos* — CMC Type
allri	Any, Any, Any — IRUCOMP, IRURTR	N/A
allrq	Any, Any, Any — QCRRTR	N/A
p*	All IRUCOMPs with matching partitions	N/A

Chassis Management Controller (CMC) and Baseboard Management Controller (BMC) Commands

The following topics describe the CMC and BMC commands.

auth

The auth command sets the average peak performance in weighted teraflops (APPWT) limit for export compliance. You can obtain the key and reset the limit if needed.

To see the usage statement:

```
CMC:r014i01c> auth --help

usage: auth  [-ard] [-s <SSN>] <key> [<key>]... [--help] [TARGET]...
-s, --ssn=<SSN>           system serial number change
-a, --appwt               APPWT change
-r, --reset               set learn mode
-d, --disable             disable authentication
<key>                     authentication keys
--help                    display this help and exit
```

autopower

The autopower command allows your system to power up automatically when power is applied after a power outage. Issuing the autopower command without any argument acts as a query, with the results displaying the current state

(enabled / disabled) and (if enabled) the current delay time and the results of the last attempted automatic power on.

To determine if autopower is enabled:

```
CMC:r001i01c> autopower
==== r001i01c (PRI) ====
auto-power on is enabled (120 second delay), aborted (power already on).
```

To disable auto-power:

```
uv44-cmc CMC:r001i01c> autopower -d
==== r001i01c (PRI) ====
auto-power on disabled
```

To see the usage statement:

```
CMC:r001i01c> autopower --help

usage: autopower  [-edc] [-t <seconds>] [--help]
-e, --reset               enable auto-power on
-t, --time=<seconds>      time (seconds) to delay auto-power on
-d, --disable             disable auto-power on
-c, --cancel              cancelauto-power on
--help                    display this help and exit
```

bios

The `bios` command displays BIOS information for the system after it is powered on.

To see the usage statement:

```
CMC:r1i1c> bios --help

usage: bios  [-rsuv] [--help] [TARGET]...
-r, --revision            display last BIOS revision banner
-s, --state               display current BIOS state
-u, --uptime              display time since last BIOS reset
-v, --verbose             verbose output
--help                    display this help and exit
```

Example 1. The following shows `bios` command usage on an SGI UV 2000:

```
CMC:r001i01c> bios
==== 8/8 BMC(s) ====
h0: Last booted: SGI BIOS 2.1.0 built in romley0tmp by habeck on Oct  1 2012 at 11:52:44
h1: Last booted: SGI BIOS 2.1.0 built in romley0tmp by habeck on Oct  1 2012 at 11:52:44

harp35-cmc CMC:r001i01c>
harp35-cmc CMC:r001i01c>
harp35-cmc CMC:r001i01c>
CMC:r001i01c> bios -s
==== 7/8 BMC(s) ====
h0: [S:0x02,DF:0x001e01] Sleeping    - Handoff (SlaveHandoff)
h1: [S:0x02,DF:0x001e01] Sleeping    - Handoff (SlaveHandoff)
==== 1/8 BMC(s) [r001i01b00] ====
h0: [S:0x04,DF:0x00aa00] EFI Shell   - Shell
h1: [S:0x02,DF:0x001e01] Sleeping    - Handoff (SlaveHandoff)
harp35-cmc CMC:r001i01c>
```

Example 2. The following shows `bios` command usage on an SGI UV 1000:

```
CMC:r1i1c> bios
==== 4/4 BMC(s) ====
Last booted: SGI BIOS Version 2 Revision 0 built in 20101012_1502 by ajm on Oct 12 2010 at 15:08:29
```

bmc

The `bmc` command sends a command to one or more BMCs. It is a shell command similar in behavior to the `ssh` command.

To see the usage statement:

```
CMC:r1i1c> bmc --help

usage: bmc [-t] exec <command> [--timeout=<seconds>] [TARGET]...
exec                      executes command on BMC(s)
<command>                 command to execute
--timeout=<seconds>       timeout value
-t, --terse               terse mode

usage: bmc list [TARGET]...
list                      list active shells on BMC(s)
```

```
usage: bmc kill [TARGET]...
kill                    kill all active shells on BMC(s)

usage: bmc [-v] runtime [TARGET]...
runtime                 show daemon run time
-v, --verbose           verbose mode

usage: bmc reboot [TARGET]...
reboot                  initiates controller reboot

usage: bmc --help
--help                  display this help and exit
```

cmc

The cmc command sends a command to one or more CMCs. It is a shell command similar in behavior to the ssh command.

To see the usage statement:

```
CMC:r1i1c> cmc --help

usage: cmc [-t] exec <command> [--timeout=<seconds>] [TARGET]...
exec                    executes command on CMC(s)
<command>               command to execute
--timeout=<seconds>     timeout value
-t, --terse             terse mode

usage: cmc list [TARGET]...
list                    list active shells on CMC(s)

usage: cmc kill [TARGET]...
kill                    kill all active shells on CMC(s)

usage: cmc [-v] runtime [TARGET]...
runtime                 show daemon run time
-v, --verbose           verbose mode

usage: cmc reboot [TARGET]...
reboot                  initiates controller reboot
```

```
usage: cmc --help
--help                    display this help and exit
```

config

The `config` command shows your system configuration. For example:

```
CMC:r1i1c> config -v

CMCs:            1
        r001i01c UV1000 SMN

BMCs:            4
        r001i01b00 IP93-BASEIO
        r001i01b01 IP93-DISK
        r001i01b02 IP93
        r001i01b03 IP93

Partitions:      1
        partition000 BMCs:    4

Accessories:     0
```

To see the usage statement:

```
CMC:r1i1c> config --help
usage: config  [-tvd] [-a <ip>@<location>] [--pcheck] [--plist] [<TARGET>]... [--help]
-t, --terse               terse output
-v, --verbose             verbose output
-d, --debug               debug output
-a, --acc=<ip>@<location>
        configure accessory location
        <location> is <rack>.<uposition> for IO
        <location> is <rack>.<0|1> for PDUs (0=left, 1=right)

--pcheck                  check for targets crossing partitions
--plist                   list partitions targeted
--help                    display this help and exit
```

The `-t`/`--terse` option produces a terse configuration report:

- On an SGI UV 2000, it produces a terse configuration report similar to the following:

```
CMC:r001i01c> config -t

SSN: UV2-00000010, CMCs: 2, BMCs 2/2, Partitions: 1, Accessories: 0
r001i01c b-00-
r001i02c b-00-
```

- On an SGI UV 100 system with two IRUs, it produces a terse configuration report similar to the following:

```
SSN: UVL-00000001, CMCs: 2, BMCs 4/4, Partitions: 1, Accessories: 0
r001i01c b-00-01
r001i02c b-00-01
```

- For a single SGI UV 1000 IRU without blades in slots 0 through 7 (or router blades), this output shows the “holes” for blades not present, similar to the following:

```
SSN: UV-00000021, CMCs: 1, BMCs 8/8, Partitions: 1, Accessories: 2
r001i01c b-  -  -  -  -  -  -  -  -08-09-10-11-12-13-14-15 r-  -  -  -
```

  On VT100–compliant terminals, it will use the following character attributes:

```
Reverse red - blade is connected, but not configured

Reverse yellow - blade is disabled (see hwcfg command)
```

- On an SGI UV 100 system with eight IRUs, it produces a terse configuration report similar to the following:

```
SSN: UV-00000048, CMCs: 16, BMCs 192/192, Partitions: 5, Accessories: 0
r001i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r001i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r001q42c r-00-01-02-03
r001q43c r-00-01-02-03
r002i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r002i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r002q42c r-00-01-02-03
r002q43c r-00-01-02-03
r003i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r003i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r003q42c r-00-01-02-03
r003q43c r-00-01-02-03
```

```
r004i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r004i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15 r-00-01-02-03
r004q42c r-00-01-02-03
```

console

The `console` command opens a console window on the CMC, BMC, or compute blade. Use `CTRL -] q` to exit the console.

Note: The `console` command supersedes the `uvcon` command. The `uvcon` command is an alias to the `console` command.

To see the usage statement for SGI UV 2000:

```
CMC:r001i01c> console --help

usage: console [-bnd23] [--steal] [--spy] [--kill] [--notty] [--nocache] [--clear] [--full] [--quiet] [TARGET]
-b, --baseio                    specifies baseio bmc console
-n, -0, --normal                specifies node BMC console (normal channel)
-d, -1, --debug                 specifies node BMC console (debug channel)
-2, --chan2                     specifies node BMC console (channel 2)
-3, --chan3                     specifies node BMC console (channel 3)
--steal                         steal the console
--spy                           spy the console
--kill                          kill all other uvcon sessions
--notty                         disables tty interventions
--nocache                       don't return cached output
--clear                         clear cached output
--full                          full output (default is since last reset)
--quiet                         quiet mode (only system I/O, implies --notty --nocache)
TARGET                          console target

 escape codes:
    ctrl-] s    steal console
    ctrl-] r    release console
    ctrl-] c    connection status
    ctrl-] ?|h  this help

usage: console -d[band23cCft] [-l ] [TARGET]...
```

```
-d, --dump                    dump cached console output
-b, --baseio                  specifies baseio bmc console
-a, --all                     all node BMC consoles
-n, -0, --normal              specifies node BMC console (normal channel)
-d, -1, --debug               specifies node BMC console (debug channel)
-2, --chan2                   specifies node BMC console (channel 2)
-3, --chan3                   specifies node BMC console (channel 3)
-l, --lines=<count>           limit output to last  lines
-c, --clear                   clear cached output after dumping
-C, --CLEAR                   clear cached output without dumping
-f, --full                    full dump (default is since last reset)
-t, --tag                     prefix each line with source
TARGET                        console target

usage: console -i  [TARGET]...
-i, --input=<input>           backdoor baseio console input
TARGET                        console target

usage: console --help
--help                        display this help and exit
```

To see a usage statement for SGI UV 100 or SGI UV 1000:

```
CMC:r014i01c> console --help

usage: console [-bnd23] [--steal] [--spy] [--kill] [--notty] [--nocache] [--clear] [TARGET]
-b, --baseio                  specifies baseio bmc console
-n, -0, --normal              specifies node BMC console (normal channel)
-d, -1, --debug               specifies node BMC console (debug channel)
-2, --chan2                   specifies node BMC console (channel 2)
-3, --chan3                   specifies node BMC console (channel 3)
--steal                       steal the console
--spy                         spy the console
--kill                        kill all other uvcon sessions
--notty                       disables tty interventions
--nocache                     don’t return cached output
--clear                       clear cached output
TARGET                        console target

 escape codes:
    ctrl-] s    steal console
    ctrl-] r    release console
```

```
    ctrl-] c    connection status
    ctrl-] ?|h  this help

usage: console -d[band23cCft] [-l <count>] [TARGET]...
-d, --dump                      dump cached console output
-b, --baseio                    specifies baseio bmc console
-a, --all                       all node BMC consoles
-n, -0, --normal                specifies node BMC console (normal channel)
-d, -1, --debug                 specifies node BMC console (debug channel)
-2, --chan2                     specifies node BMC console (channel 2)
-3, --chan3                     specifies node BMC console (channel 3)
-l, --lines=<count>             limit output to last <count> lines
-c, --clear                     clear cached output after dumping
-C, --CLEAR                     clear cached output without dumping
-f, --full                      full dump (default is since last reset)
-t, --tag                       prefix each line with source
TARGET                          console target

usage: console [-bnd23] -i <input> [TARGET]...
-i, --input=<input>             backdoor console input
-b, --baseio                    specifies baseio bmc console
-n, -0, --normal                specifies node BMC console (normal channel)
-d, -1, --debug                 specifies node BMC console (debug channel)
-2, --chan2                     specifies node BMC console (channel 2)
-3, --chan3                     specifies node BMC console (channel 3)
TARGET                          console target

usage: console --help
--help                          display this help and exit
```

Note: The following character sequences are available interactively after a `console` session is started:

```
console: escape codes:
console:    ctrl-] s    steal console
console:    ctrl-] r    release console
console:    ctrl-] c    connection status
console:    ctrl-] ?|h  this help
```

flashbios

The `flashbios` command rewrites (flashes) the latest BIOS located in the `/work/bmc/common` directory on the CMC.

For an example of how this command is used, see "Upgrade System BIOS" on page 22.

Following is information about `flashbios`:

```
NAME
        flashbios -- Flash UV BIOS from CMC

SYNOPSIS
        flashbios [-e] [-n] [-p] [-o] <flashfile> [<target> [...<target>]]
        flashbios -r <flashfile>

DESCRIPTION
        The file specified by flashfile is flashed into the flash devices
        on all nodes of the UV system.  By default the system will be
        automatically reset after the flash completes.

        The BIOS image file <flashfile> must be located in the directory
        /work/bmc/common on the CMC.

        By default, BIOS is updated on all compute blades in the system.
        The flash can be restricted to a subset of the full system by
        specifying one or more targets. Some examples of blade targets:

                r1i23b4         blade 4 in the IRU @ rack 1, u-position 23
                r2i1b*          all blades in IRU @ rack 2i1
                r4i*            all blades in all IRUs in rack 4
                p2              all blades in partition 2

        (These are the same targets accepted by other CMC commands).
        Any number or combination of targets can be used.


        The following options are available:

        -e      Erase BIOS variables and BIOS scratch space while
                resetting the system.  This option will be ignored
```

```
        if '-n' is also specified.

-n      Do not automatically reset the system.

-o      Allow flashing across partitions in a multiple partition system

-p      Power cycle after flashing.

-r      Do not flash, instead report the Revision (BIOS banner) string
        from the BIOS image file.
```

hel

The `hel` command displays the hardware error logs.

To get a usage statement:

```
 CMC:r001i01c> hel --help

usage: hel  [-cax] [--help] [TARGET]...
-c, --clear               clear log
-a, --all                 show all errors (default is since last reset)
-x, --hex                 hexidecimal dump
--help                    display this help and exit
```

hwcfg

The `hwcfg` command displays and sets hardware override variables.

For information about how to use the `hwcfg` command to partition an SGI UV system, see the *SGI UV Systems Linux Configuration and Operations Guide.*

For example, for SGI UV 1000:

```
CMC:r1i1c> hwcfg -a -v
BLADE_DISABLE=no
DEBUG_SW=0x0
HUB_CORE_SPEED=400
ICH_DISABLE=no
```

```
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes
NL5_ENABLE=yes
NL5_HUB2_WAR=no
NL5_NEAR_LB=no
NL5_RATE=6.25
NL5_SCRAMBLE=0x0
PARTITION=0
SMT_ENABLE=no
SOCKET_DISABLE=no
```

For example, for SGI UV 1000 showing a system with four partitions:

```
CMC:r1i1c> hwcfg -a -v
NL5_RATE=5.0
PARTITION=1 ..................................................... 16/64 BMC(s)
PARTITION=2 ..................................................... 16/64 BMC(s)
PARTITION=3 ..................................................... 16/64 BMC(s)
PARTITION=4 ..................................................... 16/64 BMC(s)
```

You can use `hwcfg -c` to clear the four partitions, as follows:

```
CMC:r1i1c> hwcfg -c partition
PARTITION=0 <PENDING RESET>
```

For example, for SGI UV 2000:

```
CMC:r001i01c> hwcfg -a -v
BIOS_FILE=/common/bios.fd
        all targeted BMC(s)
BLADE_DISABLE=no
        all targeted BMC(s)
DEBUG_SW=0x4
        all targeted BMC(s)
IORISER_DISABLE=no ............................................. 1/2 BMC(s)
        r001i01b00
IORISER_DISABLE=yes ............................................ 1/2 BMC(s)
        r001i02b00
MAX_CORES=0
        all targeted BMC(s)
NL6_ENABLE=0x808
        all targeted BMC(s)
```

```
PARTITION=0
        all targeted BMC(s)
ROUTER_TYPE=ordinary
        all targeted BMC(s)
SOCKET_DISABLE=no
        all targeted BMC(s)
```

To see the usage statement:

```
CMC:r1i1c> hwcfg --help

usage: hwcfg  [-lhadcv] [var[=val]]... [--help] [TARGET]...

var[=val]                variable [and value to set]
-l, --list               list variables available
-h, --hidden             include hidden variables
-a, --all                show variable(s) regardless of override state
-d, --default            show variable(s) default value if overriden
-c, --clear              clear overridden variable(s) to default value
-v, --verbose            show per BMC output, repeat for maximum verbosity
--help                   display this help and exit, use with variable(s) to get specific help
```

leds

The `leds` command displays system LED values.

To see the usage statement:

```
CMC:r001i01c> leds --help

usage: leds  [-sv] [-c <index>] [-d <value>] [--help] [TARGET]...

-c, --cpu=<index>         index of cpu to display
-d, --delay=<value>       sample delay value in seconds
-s, --summary             show shorter LED history
-v, --verbose             show longer LED history
--help                    display this help and exit
```

log

The log command provides a log of various operations performed on the CMC.

To see the usage statement:

```
CMC:r001i01c> log --help

usage: log  [-pacewnidmr] [--help] [TARGET]...
-p, --panic               select log type
-a, --alert               select log type
-c, --critical            select log type
-e, --error               select log type
-w, --warning             select log type
-n, --notice              select log type
-i, --info                select log type
-d, --debug               select log type
-m, --merge               prevent merging of selected logs
-r, --reset               reset log
--help                    display this help and exit
```

power

The power command lets you power on, power off, reset, cycle, get status, and invoke the kernel debugger (KDB).

When you use the power command, you do not have to power up the IRU. When the power command runs, it checks to see if the IRU is powered on. If the IRU is not powered on, it powers up the IRU, and it then powers up the compute blades.

To see the usage statement:

```
CMC:r1i1c> power --help

usage: power [-vchosw] on|up [bmc] [--noio] [TARGET]...
on|up                     turn power on
bmc                       turn aux power on
--noio                    do not power on accessories (IO)
-v, --verbose             verbose output
-c, --clear               clear EFI variables (system/partition targets only)
-h, --hold                hold in reset
-o, --override            override partition check
```

```
-s, --single              single node boot
-w, --watch               watch boot progress

usage: power [-vo] off|down [bmc] [--noio] [--nochassis] [TARGET]...
off|down                  turn power off
bmc                       turn aux power off
--noio                    do not power off accessories (IO)
--nochassis               do not power off chassis power
-v, --verbose             verbose output
-o, --override            override partition check

usage: power [-vchosw] reset [bmc|iobmc] [TARGET]...
reset                     system reset
bmc|iobmc                 BMC reset
-v, --verbose             verbose output
-c, --clear               clear EFI variables (system/partition targets only)
-h, --hold                hold in reset
-o, --override            override partition check
-s, --single              single node boot
-w, --watch               watch boot progress

usage: power [-vchosw] cycle [bmc] [--noio] [--nochassis] [TARGET]...
cycle                     cycle power off on
bmc                       cycle aux power
--noio                    do not power cycle accessories (IO)
--nochassis               do not power cycle chassis power
-v, --verbose             verbose output
-c, --clear               clear EFI variables (system/partition targets only)
-h, --hold                hold in reset
-o, --override            override partition check
-s, --single              single node boot
-w, --watch               watch boot progress

usage: power [-v10ud] [status] [TARGET]...
status                    show power status
-v, --verbose             verbose output
-1, --on                  show only blades with on status
-0, --off                 show only blades with off status
-u, --unknown             show only blades with unknown status
-d, --disabled            show only blades with disabled status
```

```
usage: power [-ov] nmi|debug [TARGET]...
nmi|debug                 issue NMI
-o, --override            override partition check
-v, --verbose             verbose output

usage: power [-v] margin [high|low|norm|<value>] [TARGET]...
margin                    power margin control
high|low|norm|<value>     margin state
-v, --verbose             verbose output

usage: power cancel [TARGET]...
cancel                    cancel outstanding power action

usage: power --help
--help                    display this help and exit
```

sensor

The `sensor` command is normally used remotely by SMN-based applications. However, you can use the `sensor` command to get system temperatures, fan speed, and voltage information, as follows:

```
CMC:r1i1c> sensor
```

Use the command from the `cmc` or `bmc` shell, as follows:

```
cmc sensor
bmc sensor
```

The BMCs on SGI UV 2000 systems do not support the `sensor` command.

version

The `version` command displays the CMC firmware version. For example:

```
uv44-cmc CMC:r001i01c> version
SGI Chassis Manager Controller, Firmware Rev. 1.3.16   [Bootloader 0.6.0]
```

You can use the `version` command to determine the BMC firmware version, as follows:

```
uv44-cmc CMC:r001i01c> bmc version
==== r001i01b00 ====
SGI UV BMC, Firmware Rev. 1.3.20   [Bootloader 0.5.0, FPGA Rev. 0x41]
==== r001i01b01 ====
SGI UV BMC, Firmware Rev. 1.3.20   [Bootloader 0.5.0, FPGA Rev. 0x41]
```

You can access the BMC using `ssh` and then use the `version` command, as follows:

```
uv44-cmc CMC:r001i01c> ssh bmc0

SGI UV BMC, Firmware Rev. 1.3.20

BMC:r001i01b00> version
SGI UV BMC, Firmware Rev. 1.3.20   [Bootloader 0.5.0, FPGA Rev. 0x41]
```

You can access the Base I/O blade BMC using `ssh` and then use the `version` command, as follows:

```
uv44-cmc CMC:r001i01c> ssh ibmc0

SGI UV BMC, Firmware Rev. 1.3.9
r001i01b00i> version
SGI Chassis Manager Controller, Firmware Rev. 1.3.9   [Bootloader 0.3.0]
```

Supplemental Hardware Information

This appendix contains additional hardware information that might be helpful when performing chassis management controller (CMC) procedures. The topics in this appendix are as follows:

Determining Rack Numbers

This section discusses the following:

SGI UV 2000 Bay and Rack Numbers

For SGI UV 2000, bays (or units) in the racks are numbered using standard units. A *standard unit (SU)* or *unit (U)* is equal to 1.75 inches (4.445 cm). Because IRUs occupy multiple standard units, IRU locations within a rack are identified by the bottom unit in which the IRU resides. For example, in a 42U rack, an IRU positioned in the range U01 through U10 is identified as U01.

Each rack is numbered with a three-digit number sequentially beginning with 001. A rack contains IRU enclosures, optional mass storage enclosures, and potentially other options. In a single compute rack system, the rack number is always 001.

SGI UV 1000 Rack Numbers

The system controller network has strict requirements for rack numbering. The requirements minimize the amount of information that must be manually configured for each CMC when it is plugged into an IRU. Only the rack number and unit position (*u-position*) of the IRU must be set. The u-position is the physical location of the IRU in the rack. The rack and u-position values are found in the `/etc/sysconfig/module_id` file. Besides uniquely identifying the physical location of the CMCs, the values are used to generate several IP addresses for the various VLANs on the CMC and are used by any software interacting with the system controller network to target operations.

For large SGI UV 1000 configurations, a building block consists of four racks with two IRUs in each rack with the CMCs in those IRUs interconnected via their **CMC0** and **CMC1** jacks. In order for racks to be considered part of the same building block, their rack numbers must be consecutive and satisfy the following equation:

(rack - 1) MOD 4 = 0, 1, 2 or 3

or

(rack - 1) DIV 4 = the same value for all racks in the building block

For example, a system with four racks numbered 1, 2, 3, and 4 has one building block. Similarly, a system with four racks number 9, 10, 11, and 12 has one building block.

A system with racks numbered 10, 11, 12, 13 would have two building blocks with 10, 11 and 12 in one building block; 13 is in a second building block. The system controller network must be cabled appropriately for each configuration.

A super block consists of four building blocks. Two primary CMCs in each building block are used to interconnect the building blocks via their **SBK** jacks. For racks to be considered part of the same super block, their rack numbers must be consecutive and satisfy the following equation:

(rack - 1) MOD 16 = 0,1,2,... 15

or

(rack - 1) DIV 16 = the same value for all racks in the super blocks

In summary, a single super block can support up to four building blocks (16 racks).

CMC Ethernet Ports

This section discusses the CMC Ethernet ports:

SGI UV 2000 CMC Ethernet Ports

On an SGI UV 2000 CMC, the RJ45 Ethernet ports are as follows:

Port	Description
SMN	Connects to the SMN
CMC	Connects multiple IRUs' CMCs together via a dedicated external ethernet switch
ACC	Connects miscellaneous accessory devices (for example, smart power distribution units) to the CMC network

CONSOLE supports a serial channel connection directly to the CMC for system maintenance.

SGI UV 1000 CMC Ethernet Ports

On an SGI UV 1000 CMC, the RJ45 Ethernet ports are as follows:

Port	Description
SMN	Connects to the SMN.
SBK	Connects one super block to another super block. A *super block* consists of four building blocks. Each 16–rack group is a super block.
CMC0, **CMC1**	Connects multiple IRUs' CMCs within a building block together in a string. A *building block* consists of four racks.

EXT0, **EXT1**, **EXT2**	Connects to external devices such as I/O chassis and smart power distribution units.

CONSOLE supports a serial channel connection directly to the CMC for system maintenance.

SGI UV 100 CMC Ethernet Ports

On an SGI UV 100 CMC, the RJ45 Ethernet ports are as follows:

Port	Description
ACC	Connects miscellaneous accessory devices (for example, smart power distribution units) to the CMC network. The ACC port provides network access to the CMC. It corresponds to `eth0` on the CMC and can be configured with either a static IP address or a dynamic IP address with DHCP.
SMN	Connects to the SMN.
CMC0, **CMC1**	Connects multiple IRUs to form a string topology.

CONSOLE supports a serial channel connection directly to the CMC for system maintenance.

SGI UV 2000 Diagrams

For complete details, see the *SGI UV 2000 System User Guide.*

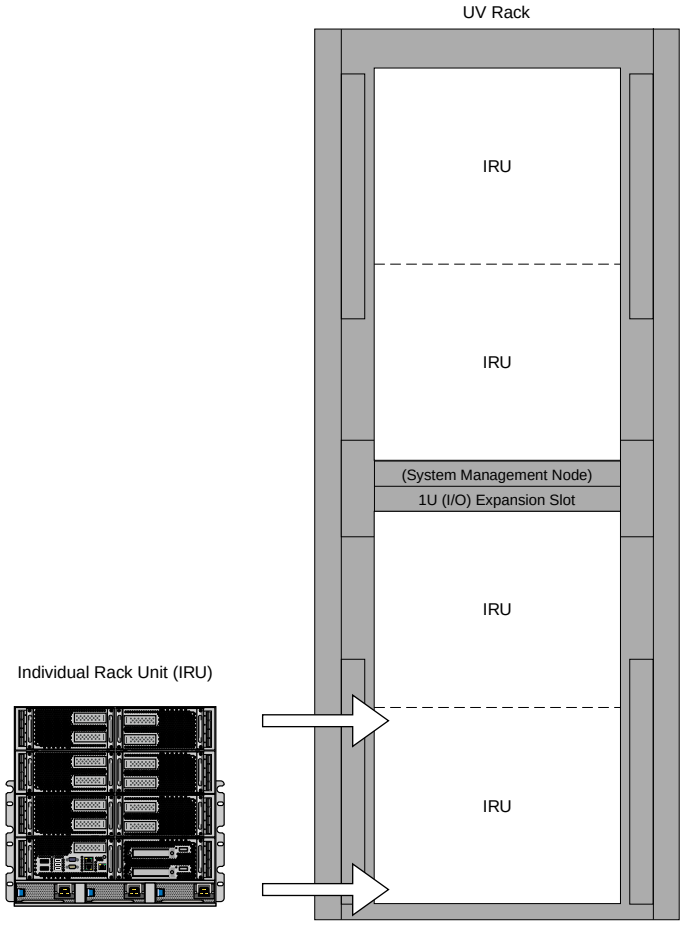

Figure A-1 SGI UV 2000 IRU

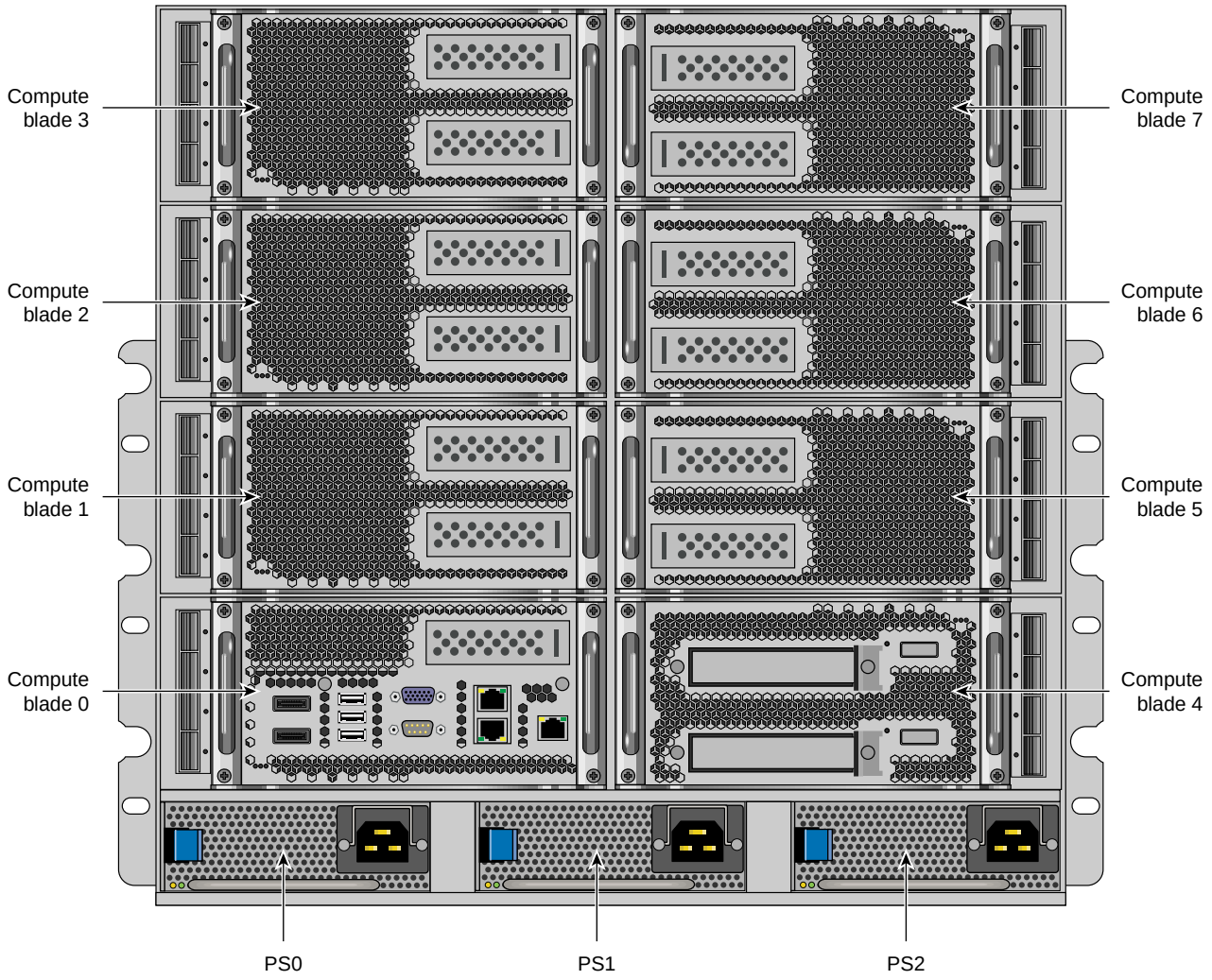

Figure A-2 SGI UV 2000 IRU Front View

SGI UV 1000 Diagrams

For complete details, see the *SGI Altix UV 1000 System User's Guide.*

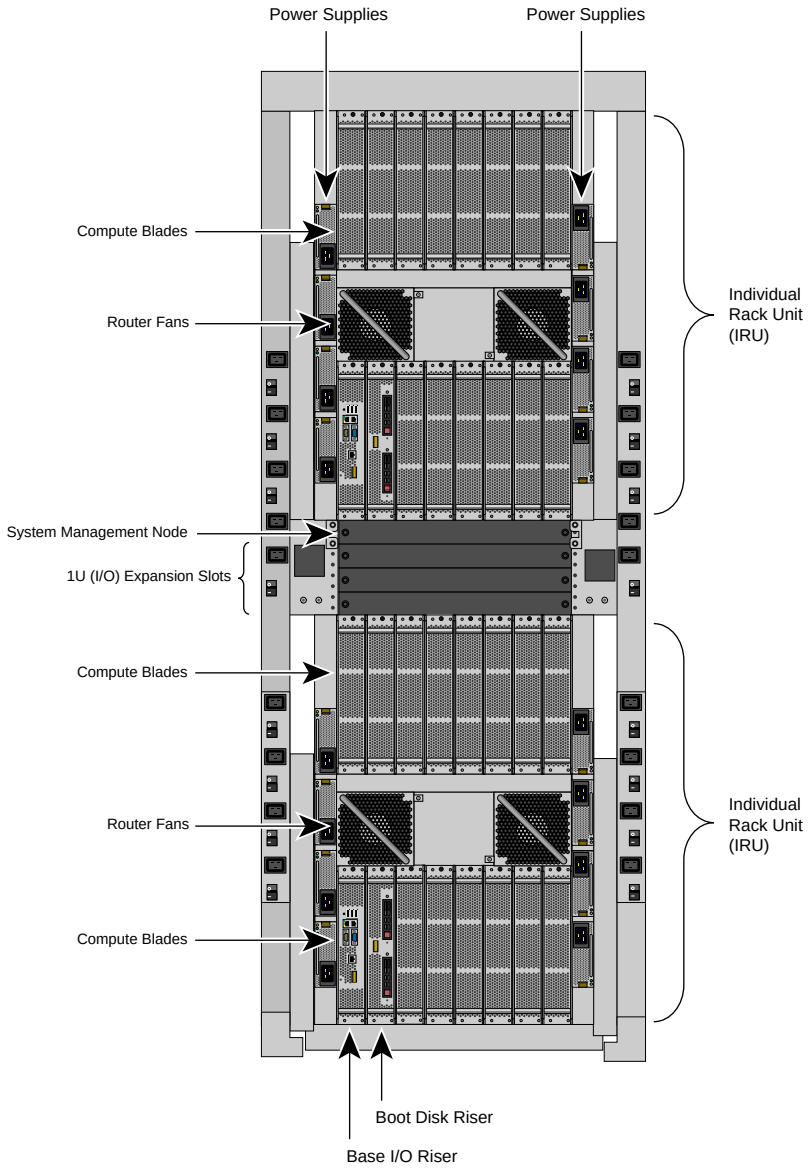

Figure A-3 SGI UV 1000 Full System Rack

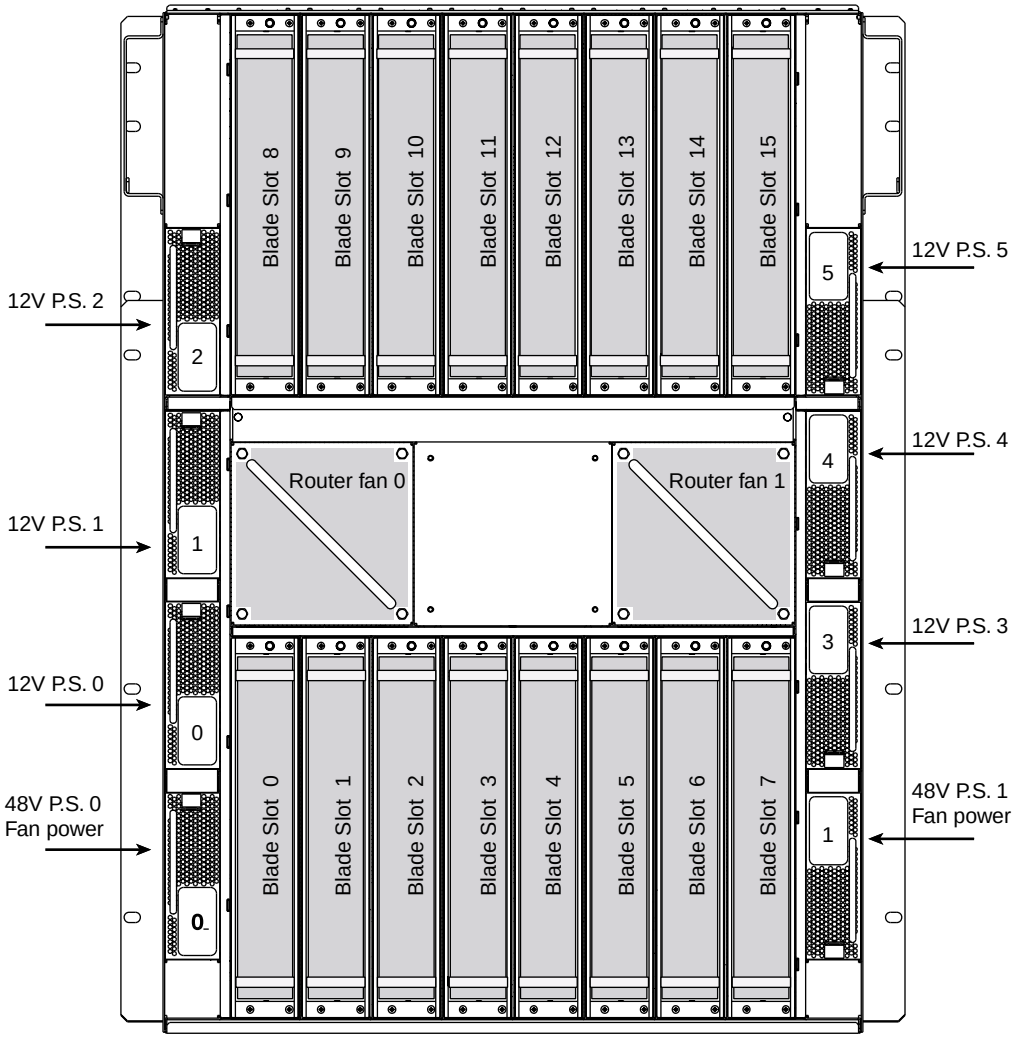

Figure A-4 SGI UV 1000 IRU

SGI UV 100 Diagram

For complete details, see the *SGI Altix UV 100 System User's Guide.*

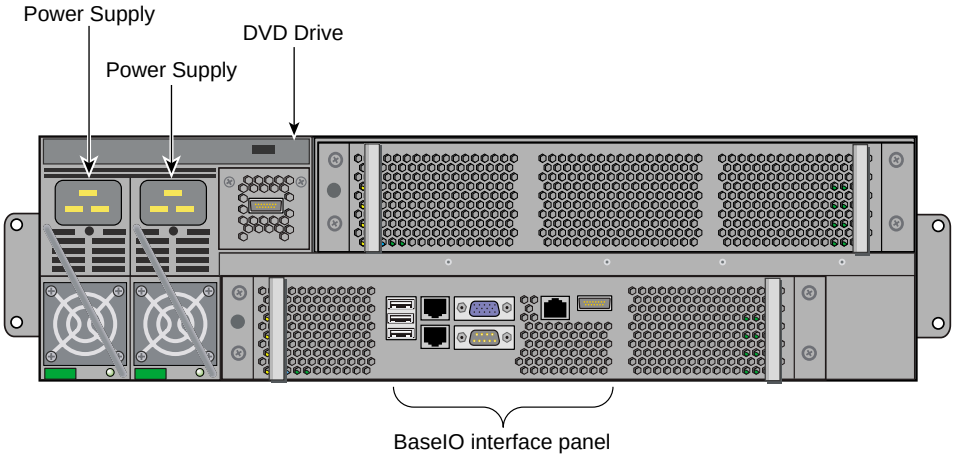

Figure A-5 SGI UV 100 IRU Front View

Index