SGI® Altix® UV CMC Controller Software User's Guide

007-5636-003

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.

New Features in This Guide

Major Documentation Changes

- Added “Using the hwcfg Command” on page 20.
- Updated information in “Upgrading System BIOS” on page 31.
- Added a new section called “CMC Command Targets” on page 33.
- Updated information in “power” on page 41.

Contents

Figures

About This Guide

This guide describes how to use the controller commands on your chassis manager controller (CMC) to monitor and manage the following systems:

- SGI Altix UV 100 systems
- SGI Altix UV 1000 systems

SGI Management Center (SMC) software running on the system management node (SMN) provides a robust graphical interface for system configuration, operation, and monitoring. This manual describes commands that can be used on systems without an SMN or not running the SMC. For more information on the SMC, see *SGI Management Center System Administrator's Guide*.

The *SGI Altix UV System Management Node Administrator's Guide* describes the system management node (SMN) for SGI Altix UV 1000 and SGI Altix UV 100 series systems. It provides information on how to install, configure, and use software on the SMN to manage and monitor SGI Altix UV systems.

Note: The UV controller commands described in this manual do not apply to SGI Altix UV 10 systems. For information on the SGI Altix UV 10 system, see the *SGI Altix UV 10 System User's Guide*.

The following topics are covered in this guide:

- Chapter 1, "Introducing Altix UV System Control Topology"
- Chapter 2, "Using the Altix UV CMC Software Commands"
- Chapter 3, "Altix UV CMC Software Commands"

Related Publications

The following publications contain additional information that may be helpful:

- *SGI Altix UV 10 System User's Guide* provides an overview of the Altix UV 10 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Altix UV 100 System User's Guide* provides an overview of the Altix UV 100 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Altix UV 1000 System User's Guide* provides an overview of the Altix UV 1000 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Performance Suite 1.0 Start Here* provides information about the SGI Performance Suite 1.0 release including information about major new features, software installation, and product support.
- S*GI Altix UV Systems Linux Configuration and Operations Guide* provides information for people who manage the operation of SGI UV systems running SGI Performance Suite software. It explains how to perform general system configuration and operations under the Linux operating system used with SGI UV systems.
- *SGI Altix UV System Management Node Administrator's Guide* describes the system management node (SMN) for SGI Altix UV 1000 and SGI Altix UV 100 series systems. It provides information on how to install, configure, and use software on the SMN to manage and monitor SGI Altix UV systems.
- *SGI Management Center Installation and Configuration* is intended for system administrators. It describes how to install and configure the SGI Management Center. A companion manual, SGI Management Center System Administrator's Guide, describes general cluster administration.
- *SGI Management Center System Administrator's Guide* describes how you can monitor and control a cluster using the SGI Management Center. A companion manual, SGI Management Center Installation and Configuration Guide, describes installing and configuring the SGI Management Center.

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at http:/ / docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- You can view release notes on your system by accessing the `README.txt` file for the product. This is usually located in the `/usr/share/doc/productname` directory, although file locations may 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
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names.
GUI element	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact SGI. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, 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:

  Technical Publications
  SGI
  46600 Landing Parkway
  Fremont, CA 94538

SGI values your comments and will respond to them promptly.

Introducing Altix UV System Control Topology

This manual describes controller software commands on SGI Altix UV 100 and SGI Altix UV 1000 systems.

Note: This manual does not apply to SGI Altix UV 10 systems. For information, see the *SGI Altix UV 10 System User's Guide.*

Altix UV 1000 Overview

The SGI Altix UV 1000 system is a blade-based, cache-coherent non-uniform memory access (ccNUMA), computer system that is based on the Intel Xeon 7500 series processor. The UV 1000 system scales, as follows:

- From 32 to 2048 threads in a single system image (SSI)
- A maximum of 2048 processor cores with hyper-threading turned off
- A maximum of 4096 processor threads (2048 processor cores) with hyper-threading turned on

Note: Each processor core supports two threads. A processor with hyper-threading enabled is treated by the operating system as two processors instead of one. This means that only one processor is physically present but the operating system sees two logical processors, and shares the workload between them. At initial release, the maximum SSI supported by the Linux operating system is 2048.

The main component is an 18U-high individual rack unit (IRU) shown in Figure 1-1 that supports 16 compute blades and is configurable to support multiple topology options.

The compute blades in the IRU are interconnected using NUMAlink 5 technology. NUMAlink 5 has a peak aggregate bi-directional bandwidth of 15 GB/s. Multiple IRUs are also interconnected with NUMAlink 5 technology.

A maximum of two IRUs can be placed into a custom 42U rack as shown in Figure 1-2. Each rack supports a maximum of 512 processor cores; therefore, the largest SSI system requires four racks. A maximum of 128 four rack cells can be interconnected to create a 512 rack system (256K processor cores).

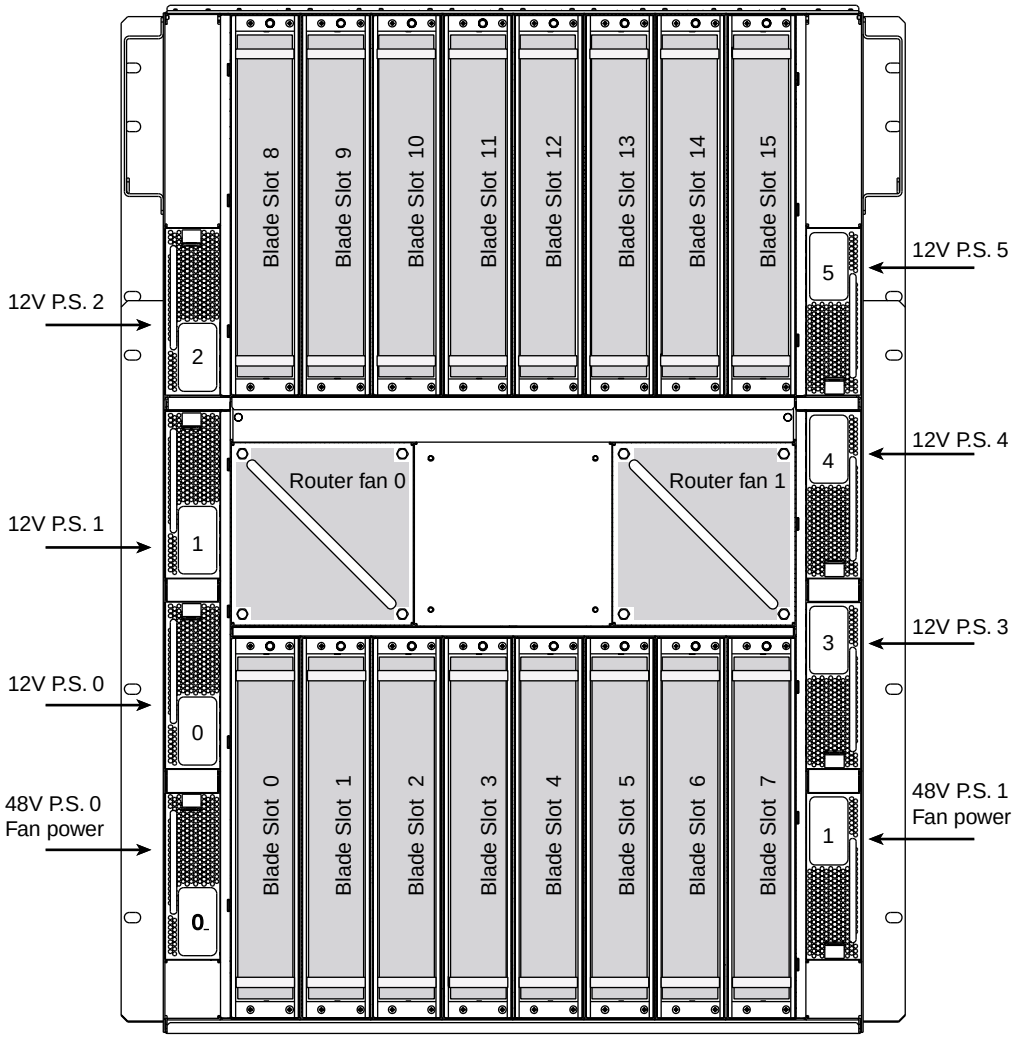

Figure 1-1 Individual Rack Unit

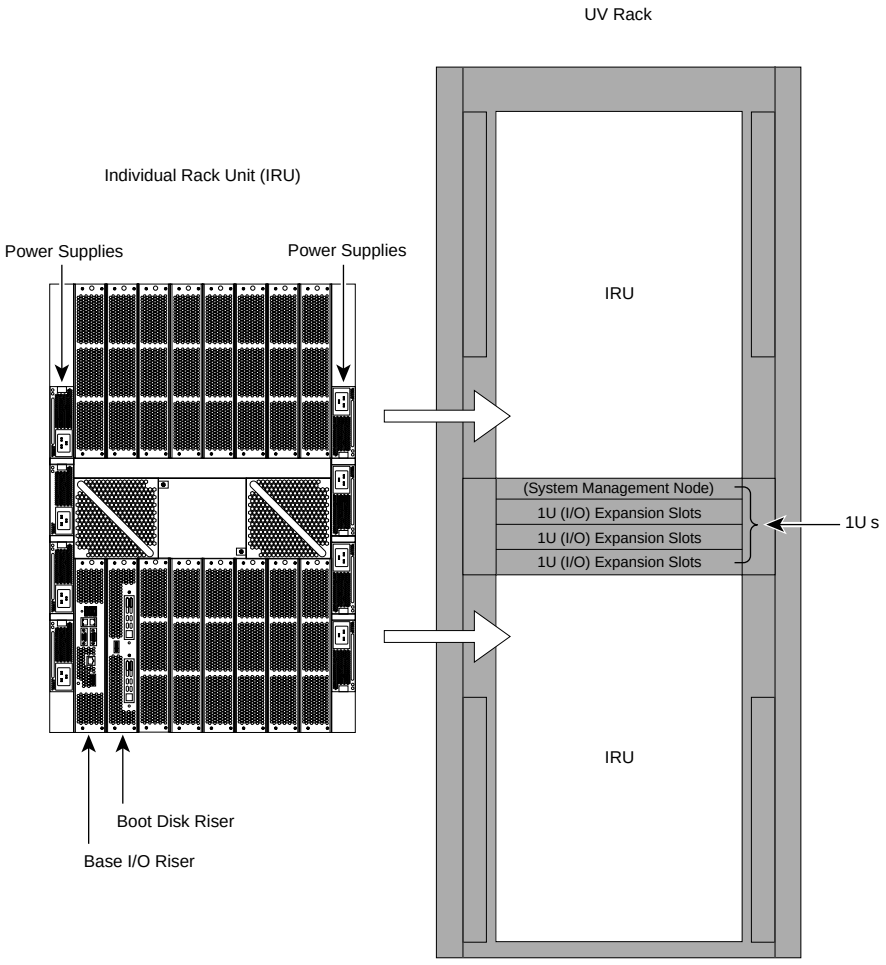

Figure 1-2 Basic System Building Blocks for Altix UV 1000 Systems

The Altix UV system supports direct attach I/O on the compute blade. The compute blade is designed to host one of four different I/O riser cards. Various PCI express based I/O components are supported. Figure 1-3 shows a full SGI Altix UV system rack.

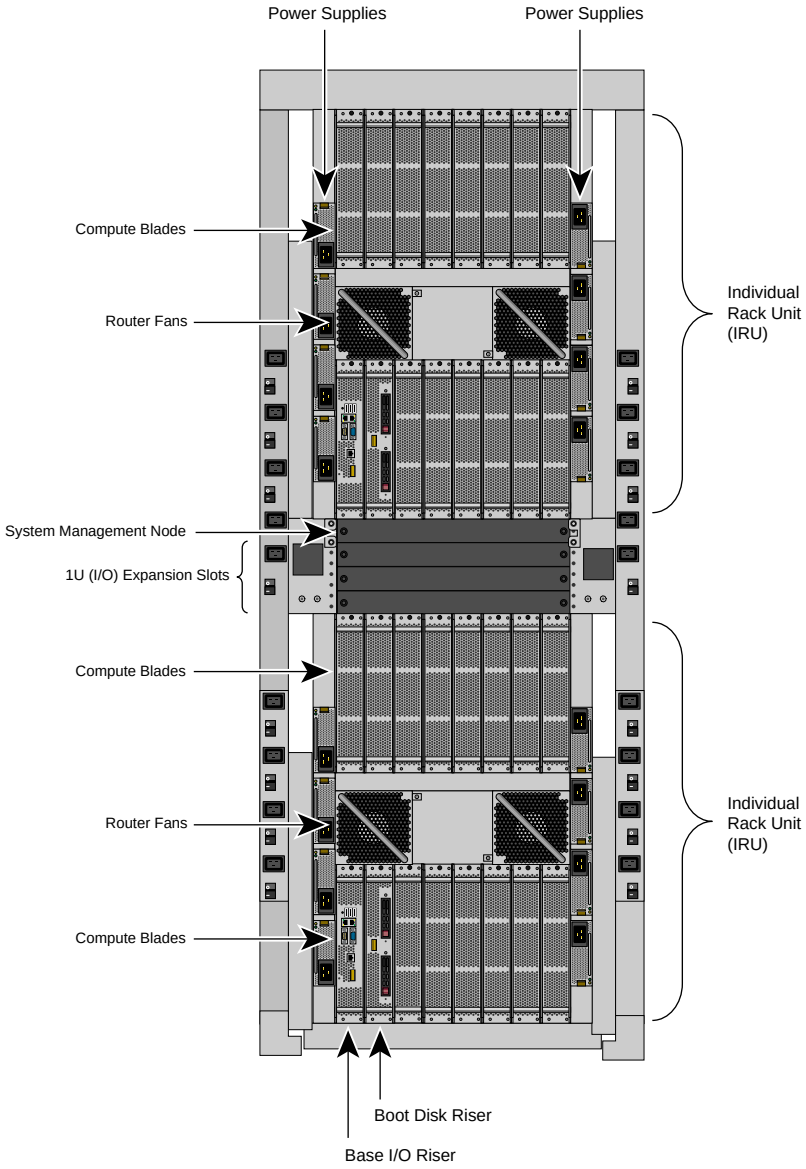

Figure 1-3 SGI Altix UV System Rack

For a detailed hardware description, see the *SGI Altix UV 1000 Systems User's Guide*. Figure 1-3 on page 5.

The SGI hardware manuals contain detailed descriptions of Altix system architecture. For a list of these manuals, see "Related Publications" on page xii.

Note: Online and postscript versions of SGI documentation is available at SGI Technical Publications Library at http://docs.sgi.com.

Altix UV 100 Overview

The SGI Altix UV 100 system is a small, blade-based, cache-coherent, non-uniform memory access (ccNUMA), computer system that is based on the Intel Xeon 7500 series processor. The SGI Altix UV 100 system scales, as follows:

A maximum of 768 processor cores

From 16 to 1536 threads in a single system image (SSI)

Note: Each processor core supports two threads.

The main component is a 3U-high IRU, shown in Figure 1-4, that supports two compute blades and is configurable to support multiple topology options.

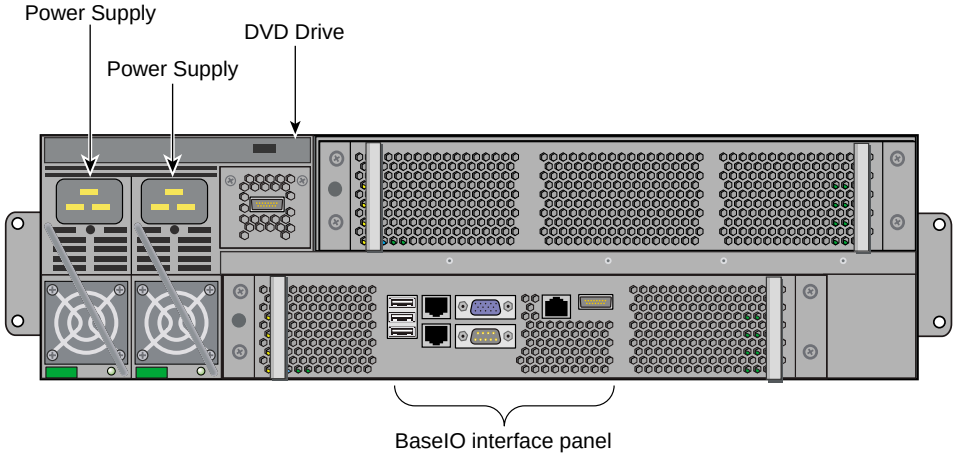

Figure 1-4 SGI Altix UV 100 IRU Front View

The two compute blades in the IRU are interconnected using NUMAlink 5 technology. NUMAlink 5 has a peak aggregate bi-directional bandwidth of 15 GB/s. Multiple IRUs are also interconnected with NUMAllink 5 technology.

A maximum of twelve IRUs can be placed into a standard 42U 19" custom tall rack. Each rack supports a maximum of 384 processor cores.

The Altix UV system supports direct attach I/O on the compute blade. The compute blade is designed to host one of four different I/O riser cards. Various PCI express based I/O components are supported. For a detailed hardware description, see the *SGI Altix UV 100 Systems User's Guide.*

System Management

The system management provides a single control point for system power up, initialization, booting and maintenance. System management on an SGI Altix UV 1000 consists of three levels. The first level of system management is the board management controllers (BMCs) on the node boards. The second level is the chassis management controllers (CMC) in the rear of the IRU. The third level is the system management node (SMN). The SMN is required on SGI Altix UV 1000 series systems. It is not required for the SGI Altix UV 100 series systems.

Important: The UV 1000 and UV 100 system control network is a private, closed network. It is not to be reconfigured in any way different from the standard UV installation, nor is it to be directly connected to any other network. The 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 also is not security hardened, nor is it tolerant of heavy network traffic, and is vulnerable to Denial of Service attacks.

The System Management Node acts as a gateway between the UV system control network and any other networks.

SGI Management Center (SMC) software running on the system management node (SMN) provides a robust graphical interface for system configuration, operation, and monitoring. This manual describes commands that can be used on systems without an SMN or not running the SMC. For more information, see *SGI Management Center System Administrator's Guide*.

Chassis Manager Controller

The chassis manager controller (CMC) in the rear of the IRU, as shown in Figure 1-5, and Figure 1-6, supports powering up and down of the compute blades and environmental monitoring of all units within the IRU. The CMC sends operational requests to the baseboard manager controller (BMC) on each compute node. The CMC provides data collected from the compute nodes within the IRU to the system management node upon request. The CMC blade on the right side of the IRU is the primary CMC. A secondary CMC is currently not supported.

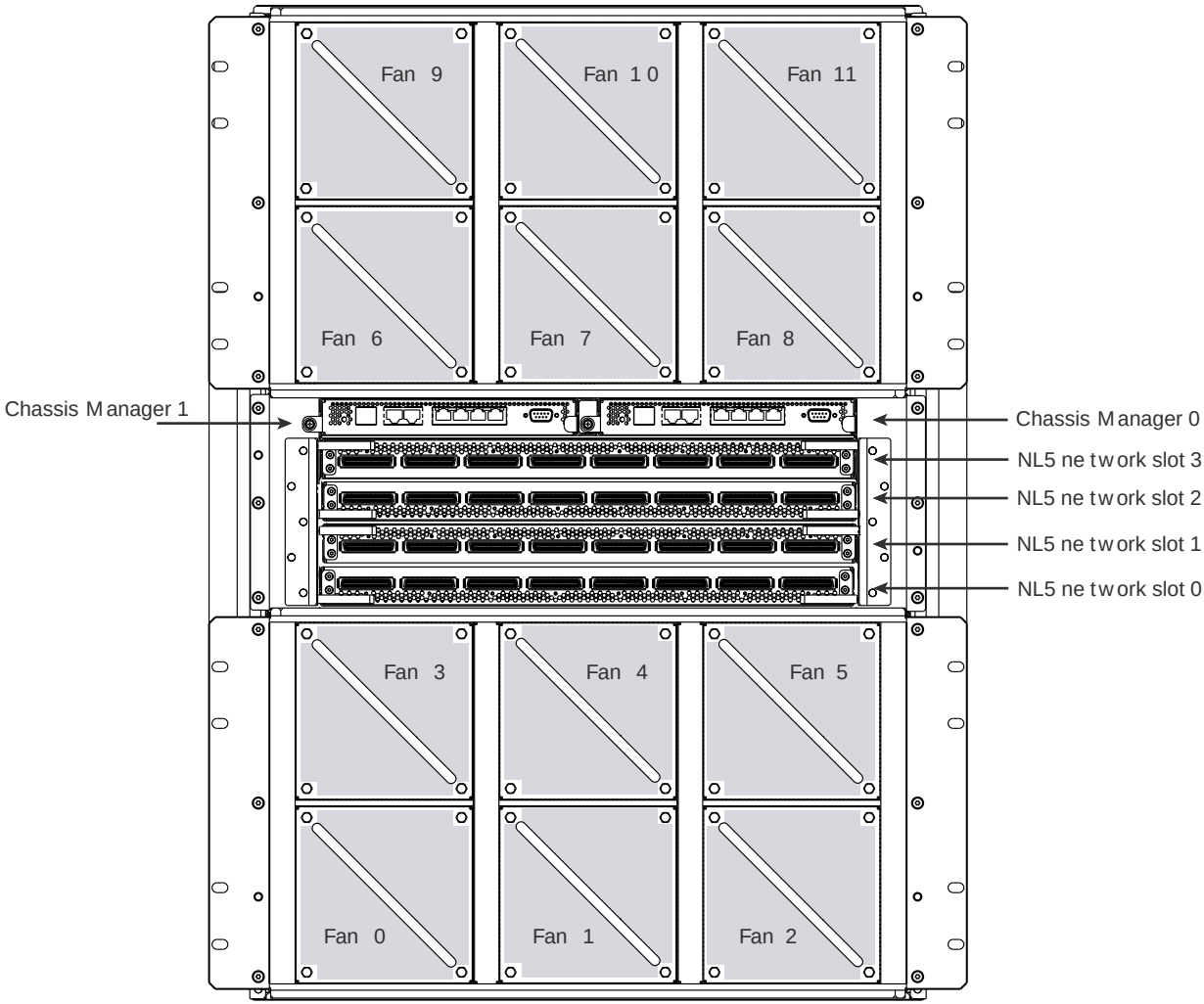

Figure 1-5 Chassis Manager Controller

System Control Network

Chassis manager controller (CMC) for SGI Altix UV 1000 systems has seven RJ45 Ethernet ports, as shown in Figure 1-6.

The Ethernet ports are used, as follows:

- **SMN** - the system management node port is used to connect to the SMN.
- **SBK** - Each 16 rack group is called a super block. A building block is four racks. A super block is four building blocks. The SBK connects one super block to another super block.
- **CMC0** and **CMC1** - these two ports are used to interconnect multiple IRUs within a building block together.
- **EXT0**, **EXT1**, **EXT2** - connects to external devices such as I/O chassis and smart PDUs.

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

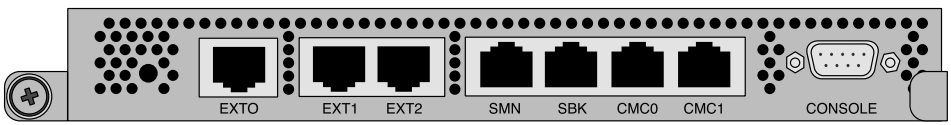

Figure 1-6 CMC Ethernet Ports on SGI Altix UV 1000 Systems

For information on finding the CMC IP address and hostname, see “Finding the CMC IP Address” on page 24.

The chassis manager controller (CMC) for SGI Altix UV 100 systems is a board assembly integrated into the IRU and has four RJ45 Ethernet ports, as shown in Figure 1-4.

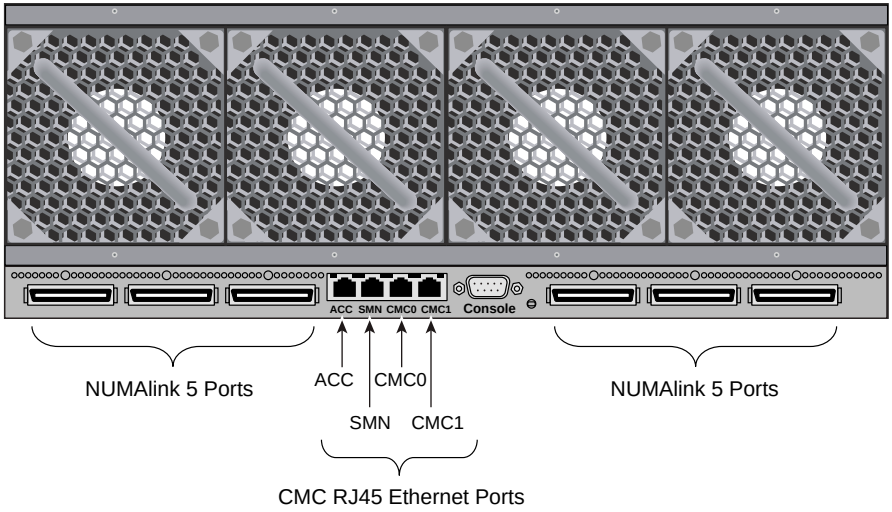

Figure 1-7 CMC Ethernet Ports on SGI Altix UV 100 Systems

The Ethernet ports are used, as follows:

- **ACC** - the accessory is used to connect miscellaneous devices to the CMC network, for example smart power distribution units (PDUs).
- **SMN** - the system management node port is used to connect to the SMN.
- **CMC0** and **CMC1** - these two ports are used to interconnect multiple IRUs together to form a string topology.

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

Determining 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. Currently, only the rack and 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 address for the various VLANs on the CMC and are used by any software interacting with the system controller network to target operations.

For large Altix 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 to 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 (SBK) consists 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 SBK 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 SBK

In summary, a single SBK can support up to four building blocks, or in other words, 16 racks.

Altix UV System Controller Software

The controller is designed to manage and monitor the individual blades in SGI Altix UV systems. Depending on your system configuration, you can monitor and operate the system from the system management node (SMN) or on smaller systems, such as, the Altix UV 100 from the CMC itself. UV 1000 systems up to 16 racks (four building blocks, also called one super block) can also be controlled and monitored from a CMC in the system.

The following list summarizes the control and monitoring functions that the CMC performs. Many of the controller functions are common across both IRU and routers; however, some functions are specific to the type of enclosure.

- Controls and monitors IRU and router fan speeds
- Reads system identification (ID) PROMs
- Monitors voltage levels and reports failures
- Monitors and controls warning LEDs on the enclosure
- Provides the ability to create multiple system partitions (single system image) running their own operating system.
- Provides ability to flash system BIOS

Using the Altix UV CMC Software Commands

This chapter describes how to use the CMC controllers to power on, manage, and monitor an SGI Altix UV 1000 or UV 100 system in the following sections:

Connecting to the UV System Controller Network

The console type and how these console types are connected to the Altix UV 1000 systems is determined by what console option is chosen. Establish either a serial connection or network / Ethernet connection to the CMC.

Establish a serial connection

If you have an Altix UV 1000 system and wish to use a serially-connected "dumb terminal", you can connect the terminal via a serial cable to the (DB-9) RS-232-style console port connector on the CMC board of the IRU.

The terminal should be set to the following functional modes:

- pin 2 - receive
- pin 3 - transmit
- pin 5 - ground
- Baud: 115200
- Data bits: 8
- Parity: no
- Stop bits: 1
- No flow control

Note that a serial console is generally connected to the first (bottom) IRU in any single rack configuration. For more information, see the “Console Hardware Requirements” section in the *SGI Altix UV 1000 System User’s Guide*.

Establish a Network/Ethernet connection (see SBK port, EXT port, and SMN port in Figure 1-6)

CMCs have their rack and u position set at the factory. The CMC will assign itself IP addresses, as follows:

SBK 172.17.<rack>.<slot>

EXT 10.<rack>.<slot>.1

On the system management node (SMN) port, the CMC is configured to request an IP address via dynamic host configuration protocol (DHCP).

Either connection, serial or network, will present a login prompt. For more information, see the “Levels of System Control” section in the *SGI Altix UV 1000 System User’s Guide*.

Power on and Booting an Altix UV System from Complete Power Off

To boot an SGI Altix UV 1000 or UV 100 system from complete power off, perform the following steps:

1. Make sure the power breakers are on.
2. Establish a serial connection to the **CONSOLE** on the CMC (see Figure 1-6 on page 10). See "Connecting to the UV System Controller Network" on page 15 or skip to the next step.
3. Establish a network connection to the CMC. "Connecting to the UV System Controller Network" on page 15. Use the `ssh` command to connect to the CMC, similar to the following example:

   Note: This is only valid if your PC is connected to the CMC (via the network connection) has its `/etc/hosts` file setup to include the CMCs.

   ```
   ssh root@hostname-cmc
   SGI Chassis Manager Controller, Firmware Rev. 0.0.22

   CMC:r1i1c>
   ```

   Typically, the default password set out of the factory is root. The CMC prompt appears. **`CMC:r1i1c`** refers to rack 1, IRU 1, CMC (see Figure 1-5 on page 9 and Figure 1-6 on page 10)

   If the host name is **not** set up in the PC / workstation's hosts file, you can simply use the IP address of the CMC, as follows:

   ```
   ssh root@<IP-ADDRESS>
   ```

4. Power up your Altix UV system using the power on command, as follows:

   ```
   CMC:r1i1c> power on
   ```

Note: You can open a second window on the CMC, **`ssh root@`***hostname***`-cmc`** and use the **`uvcon`** command to open a console and watch the system power on.

5. Open a second console to the CMC using the `uvcon` command to see the system power on, as follows:

   ```
   ssh root@hostname-cmc
   SGI Chassis Manager Controller, Firmware Rev. 0.0.22

   CMC:r1i1c> uvcon
   uvcon: attempting connection to localhost...
   uvcon: connection to SMN/CMC (localhost) established.
   uvcon: requesting baseio console access at r001i01b00...
   uvcon: tty mode enabled, use 'CTRL-]' 'q' to exit
   ```

```
uvcon: console access established
uvcon: CMC <--> BASEIO connection active
*********************************************
*******  START OF CACHED CONSOLE OUTPUT  *******
*********************************************

******** [20100512.143541] BMC r001i01b10: Cold Reset via NL
broadcast reset
******** [20100512.143541] BMC r001i01b07: Cold Reset via NL
broadcast reset
******** [20100512.143540] BMC r001i01b08: Cold Reset via NL
broadcast reset
******** [20100512.143540] BMC r001i01b12: Cold Reset via NL
broadcast reset
******** [20100512.143541] BMC r001i01b14: Cold Reset via NL
broadcast reset
******** [20100512.143541] BMC r001i01b04: Cold Reset via NL
                  ....
```

Note: Use CTRL-] q to exit the console.

6. Depending upon the size of your system, in can take 5 to 10 minutes for the Altix UV system to power on. When the **shell>** prompt appears, enter fs0, as follows:

   ```
   shell> fs0
   ```

7. At the **fs0** prompt, enter boot, as follows:

   ```
   fs0> boot
   ```

   ELILO Linux Boot loader is called and various SGI configuration scripts are run and the SUSE Linux Enterprise Server 11 SP1 installation program appears.

Power off an Altix UV System

To power down the Altix UV stem, use the `power off` command, as follows:

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

You can use the `power status` command, to check the power status of your system

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

```
on: 0, off: 32, unknown: 0, disabled: 0
```

Power NMI to Drop into KDB

To send a nonmaskable interrupt (NMI) signal from the `power` command to the CMC to drop into the kernel debugger (KDB), use the power nmi command, as follows:

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>
```

Viewing Your System Configuration

To view your system configuration, use the `config -v` command, as follows:

```
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
```

r001i01b00 refers to rack 0, IRU 1, and blade 0. For a view of the physical layout of an IRU, see Figure 1-1 on page 3, Figure 1-2 on page 4, and Figure 1-3 on page 5.

Using the hwcfg Command

The hwcfg command allows you to set hardware configuration overrides. Many of the hardware overrides available on early Altix UV systems are no longer necessary due to firmware enhancements and hardware changes. Most of these overrides are still available but have been “hidden” and require the -h flag to make them visible.

To see a list of current override settings, use the "hwcfg" command. This will show all overrides set on any blades in the system. If any overrides are set on some blades and not on others, the output will show a count of blades where the override is set.

To see a list of blades where each override is set, use the `hwcfg -v` command.

To see individual list of blades and their overrides, use `hwcfg -vv` command.

To set one or more overrides, use `hwcfg <name>=<value>` command. For example,

```
hwcfg DEBUG_SW=0x4
```

To clear overrides, use hwcfg -c command.

- To clear all overrides, use the `hwcfg -c -a` command.
- To clear one or more variables, use `hwcfg -c <name> [...<name>]`.

Multiple `<name>=<value>` pairs can be set in one command

You can show a list of `hwcfg` variables available, as follows:

```
uv44-cmc 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, perform the following:

```
uv44-cmc 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
uv44-cmc CMC:r1i1c>
```

To show a list of hidden hardware overrides, perform the following:

```
uv44-cmc:~ # hwcfg --hidden --all
BACKPLANE_TYPE=default
BLADE_DISABLE=no
DEBUG_SW=0x100
ICH_DISABLE=no
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes
NL5_ENABLE=yes
PARTITION=0
ROUTER_TYPE=ordinary
SMT_DISABLE=no
SOCKET_DISABLE=no
{HIDDEN} BIOS_GO_REG=(null):0x0
{HIDDEN} BIOS_HOLD=no
{HIDDEN} BMC_RESET_DEBUG=0x0
{HIDDEN} BOOTMODE=DC
{HIDDEN} HUB_CORE_SPEED=375
{HIDDEN} HUB_DISABLE=no
{HIDDEN} HUB_QPI_BMC_CONFIG=FULL
{HIDDEN} HUB_QPI_SPEED=5.86
{HIDDEN} HUB_XDP=no
{HIDDEN} IOH_QPI_BMC_CONFIG=NONE
{HIDDEN} IOH_QPI_SPEED=6.4
{HIDDEN} IO_XDP=no
{HIDDEN} LTC_FILE=
{HIDDEN} NL5_CABLE_RATE=6.25
{HIDDEN} NL5_CABLE_TX_EMPHASIS=0x0
{HIDDEN} NL5_HUB2_WAR=yes
{HIDDEN} NL5_NEAR_LB=no
{HIDDEN} NL5_NO_NI3_WAR=no
{HIDDEN} NL5_RATE=6.25
{HIDDEN} NL5_SCRAMBLE=yes
{HIDDEN} NL5_TUNE_ENABLE=yes
{HIDDEN} NL5_TX_EMPHASIS=0x0
{HIDDEN} QPI_CRC_MODE=0x0
{HIDDEN} SKT_QPI_BMC_CONFIG=NONE
{HIDDEN} SKT_QPI_SPEED=6.4
{HIDDEN} SPREAD_SPEC_CLK=no
```

Finding the CMC IP Address

CMCs have their rack and u position set at the factory. The CMC will assign itself IP addresses, as follows:

SBK 172.17.<rack>.<slot>

EXT 10.<rack>.<slot>.1

On the system management node (SMN) port, the CMC is configured to request an IP address via dynamic host configuration protocol (DHCP).

To find the IP address of the CMC, connect a network cable to the SMN jack and CMC will request and get a DHCP address. See “Connecting to the UV System Controller Network” on page 15.

The IP address and hostname of your system CMC resides in the `/etc/sysconfig/ifcfg-eth0` file, as follows:

```
CMC:r1i1c> cat /etc/sysconfig/ifcfg-eth0
BOOTPROTO=static
IPADDR=137.38.82.88
NETMASK=255.255.255.0
GATEWAY=137.38.82.254
HOSTNAME=uv15-cmc
```

System Partitioning

A single SGI ProPack for Linux server can be divided into multiple distinct systems, each with its own console, root filesystem, and IP network address. Each of these software-defined group of processors are distinct systems referred to as a partition. Each partition can be rebooted, loaded with software, powered down, and upgraded independently. The partitions communicate with each other over an SGI NUMAlink connection. Collectively, all of these partitions compose a single, shared-memory cluster.

The following example shows how to use CMC software to partition a two rack system containing four IRUs into four distinct systems, use the uvcon command to open a console and boot each partition and repartiton it back to a single system.

Important: Each partition must have one base I/O blade and one disk blade for booting. `001i01b00` refers to rack 1, IRU 0, and blade00. r001i01b01 refers to rack 1, IRU 0, and blade01.

Base I/O and the boot disk are displayed by the `config -v` command, similar to the following:

```
r001i01b00 IP93-BASEIO

r001i01b01 IP93-DISK
```

1. Use the `hwcfg` command to create four system partitions, as follows:

```
CMC:r1i1c>hwcfg partition=1 "r1i1b*"
CMC:r1i1c>hwcfg partition=2 "r1i2b*"
CMC:r1i1c>hwcfg partition=3 "r2i1b*"
CMC:r1i1c>hwcfg partition=4 "r2i2b*"
```

2. Use the `config -v` command to show the four partitions, as follows:

```
CMC:r1i1c> config -v

CMCs:            4
        r001i01c UV1000 SMN
        r001i02c UV1000
        r002i01c UV1000
        r002i02c UV1000

BMCs:           64
        r001i01b00 IP93-BASEIO P001
        r001i01b01 IP93-DISK P001
        r001i01b02 IP93-INTPCIE P001
        r001i01b03 IP93 P001
        r001i01b04 IP93 P001
        r001i01b05 IP93 P001
        r001i01b06 IP93 P001
        r001i01b07 IP93 P001
        r001i01b08 IP93 P001
        r001i01b09 IP93-INTPCIE P001
        r001i01b10 IP93-INTPCIE P001
        r001i01b11 IP93-INTPCIE P001
        r001i01b12 IP93-INTPCIE P001
        r001i01b13 IP93 P001
        r001i01b14 IP93 P001
        r001i01b15 IP93 P001
```

```
r001i02b00 IP93-BASEIO P002
r001i02b01 IP93-DISK P002
r001i02b02 IP93-INTPCIE P002
r001i02b03 IP93 P002
r001i02b04 IP93 P002
r001i02b05 IP93 P002
r001i02b06 IP93 P002
r001i02b07 IP93 P002
r001i02b08 IP93 P002
r001i02b09 IP93 P002
r001i02b10 IP93 P002
r001i02b11 IP93 P002
r001i02b12 IP93 P002
r001i02b13 IP93 P002
r001i02b14 IP93 P002
r001i02b15 IP93 P002
r002i01b00 IP93-BASEIO P003
r002i01b01 IP93-DISK P003
r002i01b02 IP93 P003
r002i01b03 IP93 P003
r002i01b04 IP93 P003
r002i01b05 IP93 P003
r002i01b06 IP93 P003
r002i01b07 IP93 P003
r002i01b08 IP93 P003
r002i01b09 IP93 P003
r002i01b10 IP93 P003
r002i01b11 IP93 P003
r002i01b12 IP93 P003
r002i01b13 IP93 P003
r002i01b14 IP93 P003
r002i01b15 IP93 P003
r002i02b00 IP93-BASEIO P004
r002i02b01 IP93-DISK P004
r002i02b02 IP93 P004
r002i02b03 IP93 P004
r002i02b04 IP93 P004
r002i02b05 IP93 P004
r002i02b06 IP93 P004
r002i02b07 IP93 P004
r002i02b08 IP93 P004
r002i02b09 IP93 P004
r002i02b10 IP93 P004
r002i02b11 IP93 P004
r002i02b12 IP93 P004
```

```
        r002i02b13 IP93 P004
        r002i02b14 IP93 P004
        r002i02b15 IP93 P004

Partitions:      4
        partition001 BMCs:   16
        partition002 BMCs:   16
        partition003 BMCs:   16
        partition004 BMCs:   16
```

3. Use can also use the `hwcfg` command to display the four partitions, as follows:

```
CMC:r1i1c> hwcfg
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)
```

4. To reset the system and boot the four partitions, use the following commands:

```
CMC:r1i1c> power on
CMC:r1i1c> power reset "p*"
```

Note: In the **power reset "p*"** command, above, quotes are required to prevent shell expansion.

---

5. Use the uvcon command to open consoles to each partition and boot the partitions. Open a console to partition one, as follows:

```
CMC:r1i1c> uvcon p1
uvcon: attempting connection to localhost...
uvcon: connection to SMN/CMC (localhost) established.
uvcon: requesting baseio console access at partition 1
(r001i01b00)...
uvcon: tty mode enabled, use 'CTRL-]' 'q' to exit
uvcon: console access established (OWNER)
uvcon: CMC <--> BASEIO connection active
**********************************************
*******  START OF CACHED CONSOLE OUTPUT  *******
**********************************************

******** [20100513.215944] BMC r001i01b15: Cold Reset via NL
broadcast reset
******** [20100513.215944] BMC r001i01b07: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b13: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b05: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b06: Cold Reset via NL
broadcast reset
******** [20100513.215946] BMC r001i01b10: Cold Reset via NL
broadcast reset
******** [20100513.215946] BMC r001i01b09: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b11: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b12: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b04: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b08: Cold Reset via NL
broadcast reset
```

```
******** [20100513.215946] BMC r001i01b02: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b00: Cold Reset via NL
broadcast reset
******** [20100513.215945] BMC r001i01b14: Cold Reset via NL
broadcast reset
******** [20100513.215947] BMC r001i01b09: Cold Reset via ICH
******** [20100513.215946] BMC r001i01b12: Cold Reset via ICH
******** [20100513.215947] BMC r001i01b10: Cold Reset via ICH
******** [20100513.215947] BMC r001i01b11: Cold Reset via ICH
******** [20100513.215947] BMC r001i01b02: Cold Reset via ICH
******** [20100513.215947] BMC r001i01b00: Cold Reset via ICH
******** [20100513.215953] BMC r001i01b03: Cold Reset via NL
broadcast reset
******** [20100513.220011] BMC r001i01b01: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b08: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b07: Cold Reset via NL
broadcast reset
******** [20100513.220011] BMC r001i01b15: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b06: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b05: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b14: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b13: Cold Reset via NL
broadcast reset
******** [20100513.220011] BMC r001i01b04: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b03: Cold Reset via NL
broadcast reset
******** [20100513.220013] BMC r001i01b09: Cold Reset via NL
broadcast reset
******** [20100513.220013] BMC r001i01b10: Cold Reset via NL
broadcast reset
******** [20100513.220013] BMC r001i01b11: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b12: Cold Reset via NL
broadcast reset
******** [20100513.220012] BMC r001i01b02: Cold Reset via NL
broadcast reset
```

```
******** [20100513.220012] BMC r001i01b00: Cold Reset via NL
broadcast reset
******** [20100513.220014] BMC r001i01b09: Cold Reset via ICH
******** [20100513.220014] BMC r001i01b10: Cold Reset via ICH
******** [20100513.220014] BMC r001i01b11: Cold Reset via ICH
******** [20100513.220013] BMC r001i01b12: Cold Reset via ICH
******** [20100513.220013] BMC r001i01b02: Cold Reset via ICH
******** [20100513.220016] BMC r001i01b00: Cold Reset via ICH
******** [20100513.220035] BMC r001i01b14: Cold Reset via NL
broadcast reset
******** [20100513.220035] BMC r001i01b06: Cold Reset via NL
broadcast reset
******** [20100513.220034] BMC r001i01b15: Cold Reset via NL
broadcast reset
******** [20100513.220035] BMC r001i01b05: Cold Reset via NL
broadcast reset
******** [20100513.220034] BMC r001i01b01: Cold Reset via NL
broadcast reset
******** [20100513.220035] BMC r001i01b07: Cold Reset via NL
broadcast reset
.......................
Hit [Space] for Boot Menu.
ELILO boot:
...                .....
```

Note: Use the uvcon command to open consoles on the other three partitions and boot them. The system will then have four single system images.

6. Use the **hwcfg -c** partition command to clear the four partitions, as follows:

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

7. To reset the system and boot it as a single system image (one partition), use the following command:

```
CMC:r1i1c> power reset "p*"
```

For detailed instructions on how to use the UV controller commands to partition a system, see “System Partitioning” in the SGI Altix UV *Linux Configuration and Operations Guide.*

Upgrading System BIOS

To upgrade the compute blade BIOS, perform the following steps:

1. From the CMC prompt, to show the current PROM level perform the following command:

```
CMC:r1i1c> showbios
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/
3. Copy the latest BIOS to a directory on the CMC in `/work/bmc/common/` An example directory is, as follows:

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

4. Use the `flashbios` command to flash the compute blade BIOS, as follows:

```
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.
...
```

There are three firmware flashing commands available from the system management node (SMN) for flashing an entire SGI Altix UV system, as follows:

- `flashcmc`
- `flashbmc`
- `flashiobmc`

For more information on how to use these commands, see the “Updating Firmware” section in chapter one of the *SGI Altix UV System Management Node Administrator’s Guide*.

Hyper-Threading on Altix UV 100 or Altix UV 1000 Systems

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

Hyper-Threading (HT) Technology, developed by Intel Corporation, 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.

For more information about using HT, see “Using Cpusets with Hyper-Threads” in the *Linux Resource Administration Guide*.

Altix UV CMC Software Commands

You can use SGI Altix UV controller commands to monitor and manage SGI Altix UV systems. You can use them from the SGI Management Node command line interface (CLI) or the chassis manager controller (CMC) CLI.

Typically, commands available from the SMN or CMC command line are in the form of:

command [options] [targets]

For available commands see below or type "help" at the SMN / CMC prompt. Commands are located in `/sysco/bin` on both the SMN and CMC. For options available with each command, type "`command -help`".

CMC Command Targets

Multiple targets may be specified, if no targets are specified, all BMCs or CMCs (as appropriate for a command) are assumed. Some targets may need to be quoted to avoid wildcard expansion by the SMN / CMC shell.

Table 3-1 CMC Command Targets

Command Target	BMCs (rack,upos,slot,BMC type)	CMCs (rack,upos,CMC type)
*,all	Any,Any,Any,Any	Any,Any,Any
r*	rack,Any,Any	rack,Any,Any
r*i*	rack,upos,Any,IRUCOMP	rack,upos,IRU
r*q*	rack,upos,Any,	rack,u,
r*i*b*	rack,upos,slot,IRUCOMP	N / A
r*i*r*	rack,upos,slot,	N / A

Table 3-1 CMC Command Targets

Command Target	BMCs (rack,upos,slot,BMC type)	CMCs (rack,upos,CMC type)
r*q*r*	rack,upos,slot,	N / A
*c	N / A	Any,Any,Any
r*i*c	N / A	rack,upos,IRU
r*q*c	N / A	rack,upos,
allb,allc	Any,Any,Any,IRUCOMP	N / A
allr	Any,Any,Any	N / A
allri	Any,Any,Any	N / A
allrq N / A	Any,Any,Any,IRUCOMP	N / A
p*	all IRUCOMP with matching partition	N / A

An asterisk references all values in that position such that r* references all racks, i* references all IRUs, etc. A decimal number can replace any asterisk to narrow the target selection. Ranges are not support but a space separated list of targets is supported.

BMC Types

IRUCOMP - IRU computer blades (blade slots 0 through 15 on Altix UV 1000, or 0 through 1 on Altix UV 100).

CMC Types

IRU - Altix UV 1000 or Altix UV 100

Note: Most of the commands (`bios`, `bmc`, `cmc`, `config`, `hwcfg`, `log`, `power`, `uvcon` (console)) are available at both the CMC prompt (as user **`root`**), as well as, the system management node (SMN) prompt (as user **`sysco`**).

The available commands on the CMC are, as follows:

```
CMC:r1i1c> help
available commands are:

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

type '<cmd> --help' for help on individual command.
```

bios

Use the `bios` command to get BIOS information for your system after it is powered on, as follows:

```
CMC:r1i1c> bios
==== r001i01b00 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
2010 at 16:02:59
==== r001i01b01 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
2010 at 16:02:59
==== r001i01b02 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
2010 at 16:02:59
==== r001i01b03 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
2010 at 16:02:59
==== r001i01b04 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
2010 at 16:02:59
==== r001i01b05 ====
SGI BIOS Version 1 Revision 2 built in 20100506_1553 by ajm on May  6
                ....
```

You can get a usage statement, as follows:

```
CMC:r1i1c> bios --help

usage: bios  [-v] [--help] [TARGET]...
-v, --version             display last BIOS version/banner
--help                    display this help and exit
```

bmc

Allows you to send a command to one or more board management controllers (BMCs). It is a shell command similar in behavior to the ssh command.

You can get a usage statement, as follows:

```
CMC:r1i1c> bmc --help

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

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 --help
--help                    display this help and exit
```

cmc

Allows you to send a command to one or more chassis manager controllers (CMCs). It is a shell command similar in behavior to the ssh command.

```
CMC:r1i1c> cmc --help

usage: cmc exec <command> [-t <seconds>] [TARGET]...
exec                      executes command on CMC(s)
<command>                 command to execute
```

```
-t, --timeout=<seconds>   timeout value

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 --help
--help                    display this help and exit
```

config

The CMC `cofig` command shows your system configuration, as follows:

```
CMC:r1i1c> config -v

CMCs:            1
        r001i01c UV1000

BMCs:            2
        r001i01b00 IP93-BASEIO
        r001i01b01 IP93-DISK

Partitions:      1
        partition000 BMCs:    2

Time since last update: 1:23:53
```

You can get a usage statement, as follows:

```
CMC:r1i1c> config --help

usage: config  [-v] [--help]

   -v, --verbose             verbose output
--help                    display this help and exit
```

console (uvcon)

The console (uncon) command allows allows you to open a console window on the CMC or BMC or even a compute blade. Use `CTRL -] q` to exit. A usage statement is, as follows:

```
usage: console [-bnd23] [--steal] [--spy] [--kill] [--notty] [--nocache] [--clrcache]
[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
   --clrcache                    clear cached output
   TARGET                        console target

   NOTE: When tty mode is enabled, use 'CTRL-]' 'q' to exit.

   usage: console -d[band23c] [-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 after dumping (-cc to clear without dumping)
   TARGET                        console target(s)

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

flashbios

Use the `flashbios` command to flash the latest BIOS located in the `/work/bmc/common` directory on your CMC. For an example of how this command is used, see “Upgrading System BIOS” on page 31.

You can get a usage statement, as follows:

```
CMC:r1i1c> flashbios --help
Illegal option --
NAME
        flashbios -- Flash UV BIOS from CMC

SYNOPSIS
        flashbios [-e] [-n] [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 flash file must be located in the CMC directory /work/bmc/common
        on the CMC.  The default name of the flash file is 'bios.latest.fd'.

        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.
```

hwcfg

The `hwcfg` command can be used to access hardware override variables.

```
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
```

Here is an example showing a system with four partition.

```
uv32-cmc 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:

```
uv32-cmc CMC:r1i1c> hwcfg -c partition
PARTITION=0 <PENDING RESET>
```

You can get a usage statement, as follows:

```
CMC:r1i1c> hwcfg --help

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

var[=val]                variable [and value to set]
-d, --default            reset variable(s) to default value
-v, --verbose            show variable(s) regardless of override state
--help                   display this help and exit, use with
variable(s) to get specific help
```

log

Provides a log of various operations performed on the CMC.

power

The CMC `power` command allows you to power on, power off, reset, cycle, get status and invoke the kernel debugger (KDB). When using the `power` command, you no longer have to power up the individual rack unit (IRU). When you issue the `power` command, it checks to see if the IRU is powered on, if not, it will power it up first and then the compute blades.

You can get a usage statement, as follows:

```
CMC:r1i1c> power --help

usage: power [-vcow] on|up [bmc] [TARGET]...
on|up                       turn power on
bmc                         turn aux power on
-v, --verbose               verbose output
-c, --clear                 clear EFI variables (system and partition targets only)
-o, --override              override partition check
-w, --watch                 watch boot progress

usage: power [-vo] off|down [bmc] [TARGET]...
off|down                    turn power off
bmc                         turn aux power off
-v, --verbose               verbose output
-o, --override              override partition check

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

usage: power [-vhow] cycle [bmc] [TARGET]...
cycle                       cycle power off on
bmc                         cycle aux power
-v, --verbose               verbose output
-h, --hold                  hold reset high
-o, --override              override partition check
-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 on|off|cycle|reset all|c|<slot#>
on|off|cycle|reset        control aux power or BMC reset
all|c|<slot#>             blade slot

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

sensor

Only valid for CMC data in this form. Use the sensor command to get system temperatures, fan speed, and voltage information and so on, as follows:

```
CMC:r1i1c> sensor
```

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

```
cmc sensor
bmc sensor
```

showbios

Use the `showbios` command to show the latest BIOS version used on your system, as follows:

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

uvcon

The uvcon command allows you to open a console window on the CMC or BMC or even a compute blade. Use CTRL -] q to exit.

You can get a usage statement, as follows:

```
CMC:r1i1c> uvcon --help
usage: uvcon [-bnd23] [--smn=<hostname>] [--cmc=<hostname>] [--steal] [--spy] [--kill]
[--dump] [--notty] [--nocache] [--help] [TARGET] Note:
--smn=<hostname>               SMN hostname
--cmc=<hostname>               CMC hostname
-b, --baseio                   specifies baseio bmc console
-n, --normal                   specifies nbmc console (normal channel)
-d, --debug                    specifies nbmc console (debug channel)
-2, --chan2                    specifies nbmc console (channel 2)
-3, --chan3                    specifies nbmc console (channel 3)
--steal                        steal the console
--spy                          spy the console
--kill                         kill all other uvcon sessions
--dump                         dump the cached console, exit
--notty                        disables tty interventions
--nocache                      don't return cached output
--help                         display this help and exit
TARGET                         console target

Note: When tty mode is enabled, use 'CTRL-]' 'q' to exit.
```

Index

U

V