bullx cluster suite

Maintenance Guide

extreme computing

REFERENCE
86 A2 24FA 04

extreme computing

bullx cluster suite

Maintenance Guide

Hardware and Software

August 2010

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A2 24FA 04

Printed in France

Trademarks and Acknowledgements

We acknowledge the rights of the proprietors of the trademarks mentioned in this manual.

All brand names and software and hardware product names are subject to trademark and/or patent protection.

Quoting of brand and product names is for information purposes only and does not represent trademark misuse.

Table of Contents

List of Tables

Preface

Intended Readers

The **BAS5 for Xeon** software suite has been renamed as **bullx cluster suite (bullx CS)**. Existing **BAS5 for Xeon** distributions can be upgraded to **bullx cluster suite XR 5v3.1U2**. **bullx cluster suite** is used for the management of all the nodes of a Bull Extreme Computing cluster.

This guide is intended for use by qualified personnel, in charge of maintaining and troubleshooting the Bull Extreme Computing clusters based on Intel® Xeon® processors.

Prerequisites

Readers need a basic understanding of the hardware and software components that make up a Bull Extreme Computing cluster, and are advised to read the documentation listed in the Bibliography below.

Bibliography

Refer to the manuals included on the documentation CD delivered with your system OR download the latest manuals for your **bullx cluster suite** release, and for your cluster hardware, from: http://support.bull.com/

The *bullx cluster suite Documentation* CD-ROM (86 A2 12FB) includes the following manuals:

- *bullx cluster suite Installation and Configuration Guide* (86 A2 19FA)
- *bullx cluster suite Administrator's Guide* (86 A2 20FA)
- *bullx cluster suite Application Developer's Guide* (86 A2 22FA)
- *bullx cluster suite Maintenance Guide* (86 A2 24FA)
- *bullx cluster suite High Availability Guide* (86 A2 25FA)
- *InfiniBand Guide* (86 A2 42FD)
- *Authentication Guide* (86 A2 41FD)
- *SLURM Guide* (86 A2 45FD)
- *Lustre Guide* (86 A2 46FD)

The following document is delivered separately:

- The *Software Release Bulletin* (SRB) (86 A2 80EJ)

Important **The Software Release Bulletin contains the latest information for your delivery. This should be read first. Contact your support representative for more information.**

For **Bull System Manager**, refer to the *Bull System Manager* documentation suite.

For clusters which use the **PBS Professional** Batch Manager, the following manuals are available on the *PBS Professional CD-ROM*:

- *Bull PBS Professional Guide* (86 A2 16FE)
- *PBS Professional Administrator's Guide*
- *PBS Professional User's Guide* (on the *PBS Professional CD-ROM*)

For clusters which use **LSF**, the following manuals are available on the LSF CD-ROM:

- *Bull LSF Installation and Configuration Guide* (86 A2 39FB)
- *Installing Platform LSF on UNIX and Linux*

For clusters which include the **Bull Cool Cabinet**:

- *Site Preparation Guide* (86 A1 40FA)
- *R@ck'nRoll & R@ck-to-Build Installation and Service Guide* (86 A1 17FA)
- *Cool Cabinet Installation Guide* (86 A1 20EV)
- *Cool Cabinet Console User's Guide* (86 A1 41FA)
- *Cool Cabinet Service Guide* (86 A7 42FA)

Highlighting

- Commands entered by the user are in a frame in 'Courier' font, as shown below:

```
mkdir /var/lib/newdir
```

- System messages displayed on the screen are in 'Courier New' font between 2 dotted lines, as shown below.

```
Enter the number for the path :
```

- Values to be entered in by the user are in 'Courier New', for example:

  COM1

- Commands, files, directories and other items whose names are predefined by the system are in '**Bold**', as shown below:

  The **/etc/sysconfig/**dump file.

- The use of *Italics* identifies publications, chapters, sections, figures, and tables that are referenced.
- < > identifies parameters to be supplied by the user, for example:

  <node_name>

WARNING
A Warning notice indicates an action that could cause damage to a program, device, system, or data.

CAUTION
A *Caution* notice indicates the presence of a hazard that has the potential of causing moderate or minor personal injury.

Chapter 1. Stopping/Restarting Procedures

This chapter describes procedures for stopping and restarting cluster components, which are mainly used for maintenance purposes.

The following procedures are described:

- 1.1 *Stopping/Restarting a Node*
- 1.2 *Stopping/Restarting an Ethernet Switch*
- 1.3 *Stopping/Restarting a Backbone Switch*
- 1.4 *Stopping/Restarting the Bull Cool Cabinet Door*
- 1.5 *Stopping/Restarting the Cluster*
- 1.6 *Checking Nodes after the Boot* Phase

1.1 Stopping/Restarting a Node

1.1.1 Stopping a Node

Follow these steps to stop a node:

1. Stop the application environment. Check that the node is not running any applications by using the SINFO command on the Management Node. All user applications and connections should be stopped or closed including shells and mount points.
2. Un-mount the file system.
3. Stop the node:
   From the Management Node enter:

```
nsctrl poweroff <node_name>
```

   This command executes an Operating System (OS) command. If the OS is not responding it is possible to use:

```
nsctrl poweroff_force <node_name>
```

   Wait for the command to complete.
4. Check the node status by using:

```
nsctrl status <node_name>
```

The node can now be examined, and any problems which may exist diagnosed and repaired.

1.1.2 Restarting a Node

To restart a node, enter the following command from the Management Node:

```
nsctrl poweron <node_name>
```

Note If during the boot operation the system detects an error (temperature or otherwise), the node will be prevented from rebooting.

Check the node status

Make sure that the node is functioning correctly, especially if you have restarted the node after a crash:

- Check the status of the services that have to be started during the boot. (The list of these services is in the **/etc/rc.d** file).
- Check the status of the processes that must be started by using the **cron** command.
- The mail server, **syslog-ng** and **ClusterDB** must be working.
- Check any error messages that the mails and log files may contain.

Restart SLURM and the filesystems

If the previous checks are successful, reconfigure the node for **SLURM** and restart the file systems.

1.2 Stopping/Restarting an Ethernet Switch

- Power-off the Ethernet switch to stop it.
- Power-on the Ethernet switch to start it.
- If an Ethernet switch must be replaced, the MAC address of the new switch must be set in the Cluster Database. This is done as follows:

1. Obtain the MAC address for the switch (generally written on the switch, or found by looking at **DHCP** logs).
2. Use the **phpPgAdmin** Web interface of the DATABASE to update the switch MAC address (http://IPadressofthemanagementnode/phpPgAdmin/ user=`clusterdb` and password=`clusterdb`).
3. In the **eth_switch** table look for the **admin_macaddr** row in the line corresponding to the name of your switch. Edit and update this MAC address. Save your changes.
4. Run a **dbmConfig** command from the management node:

```
dbmConfig configure --service sysdhcpd --force -nodeps
```

5. Power-off the Ethernet switch.
6. Power-on the Ethernet switch.

The switch issues a **DHCP** request and loads its configuration from the Management Node.

See The *Administrator's Guide* for information about how to change the management of the Cluster Database.

1.3 Stopping/Restarting a Backbone Switch

The backbone switches enable communication between the cluster and the external world. They are not listed in the **ClusterDB**. It is not possible to use **ACT** for their reconfiguration.

1.4 Stopping/Restarting the Bull Cool Cabinet Door

1.4.1 Using the GUI of the Bull Cool Cabinet Door

Use the **GUI** Console of the Bull Cool Cabinet Door to power on/off the Cool Cabinet Door.

mportant **Check the power off/on states of hardware equipment included in the rack before stopping/starting the Bull Cool Cabinet Door, in order to avoid overheating issues.**

See The *Cool Cabinet Door Console User's Guide* for details about the GUI console.

1.4.2 Using nsclusterstart and nsclusterstop Commands

The Bull Cool Cabinet Doors are stopped/started when the **nsclusterstart/nsclusterstop** commands are used to stop/start the cluster, as is the case for the cluster nodes.

See Section 1.*5 Stopping/Restarting the Cluster* for more information.

1.4.3 Using the coldoorStart Command

The Cool Cabinet Door **GUI** and the **nsclusterstart** command allow you to start the Cool Cabinet Doors. Alternatively, Cool Cabinet Doors can be started using the **coldoorStart** CLI.

coldoorStart Syntax

/usr/sbin/coldoorStart { --door { <door name> | <ip address> } | --startall | --status } [--dbname <database name>] [--logfile <logfile name>] [--help]

The **coldoorStart** command starts all or a specified Cool Cabinet Door(s), or displays the status of all the Cool Cabinet Doors.

--door <door name> or **<ip address>**
: Specify one Cool Cabinet Door (by its name or its IP address) to be started.

--startall
: Start all the Cool Cabinet Doors

--status
: Get the power status of all the Cool Cabinet Doors.

[--dbname <database name>]
: Specify the Database name other than the default
Default value: **clusterdb**

[--logfile <logfile name>]
: Specify a logfile other than the default.
Default value: **/tmp/coldoorStart.log**

[--help]
: Display this menu

1.5 Stopping/Restarting the Cluster

The **nsclusterstop/nsclusterstart** scripts are used to stop or start the whole cluster. These scripts launch the various steps, in sequence, making it possible to stop/start the cluster in complete safety. For example, the stop process includes the following steps:

1. Checking the various equipment
2. Stopping the file systems (Lustre for example)
3. Stopping the storage devices
4. Stopping the nodes, except the Management Node(s)
5. Stopping the Bull Cool Cabinet Doors, if any

Important **In order to ensure that the hardware has time to cool down, delays are included for the stopping sequence for the cluster. Before stopping/starting the cluster the power off/on status for the hardware in the cabinet should be checked to ensure there is no risk of overheating.**

1.5.1 Stopping the Cluster

To stop the whole cluster in complete safety it is necessary to launch the different steps in sequence. The **nsclusterstop** script includes all the required steps.

1. From the Management Node, run:

```
nsclusterstop
```

2. Stop the Management Node.

1.5.2 Starting the Cluster

To start the whole cluster in complete safety it is necessary to launch different stages in sequence. The **nsclusterstart** script includes all the required stages.

1. Start the Management Node.
2. From the Management Node, run:

```
nsclusterstart
```

1.5.3 Configuring and Using nsclusterstop and nsclusterstart

The **nsclusterstop** and **nsclusterstart** commands use configuration files to define:

- The delay parameters between the different stages required to stop/start the cluster
- The sequence in which the group of nodes should be stopped/started. (You can run **dmbGroup show** to display the groups configured.)

By default the configuration files are respectively **/etc/clustmngt/nsclusterstop.conf** and **/etc/clustmngt/nsclusterstart.conf**. The **--file** option allows you to specify another configuration file.

Usage:

/usr/sbin/nsclusterstop [-h] | [-f, --file <filename>]

/usr/sbin/nsclusterstart [-h] | [-f, --file <filename>]

Options:

--file <filename>, -f	Specify a configuration file (default: **/etc/clustmngt/nsclusterstop.conf** and **/etc/clustmngt/nsclusterstart.conf**).
-h	Display **nsclusterstart/nsclusterstop** help.
--only_test , -o	Display the commands that would be launched according to the specified options. This is a testing mode, no action is performed.
--verbose, -v	Verbose mode.

/etc/clustmngt/nsclusterstart.conf Configuration file

```
################################################################################
#
# First Part is used to control the power on safety delay for the Cool Cabinet
Door
#
################################################################################

# time to wait for the Cool Cabinet Doors to be started
coldoor_StartDelay = 30

################################################################################
#
# Second Part is used to control the power supply of DDN and servers
#
################################################################################

# time to wait for all diskarrays ok, before powering the powerswitches on
disk_arrays_StartDelay = 300

# time to wait for all powerswitches being ON after a poweron

couplets_StartDelay = 60

# time to wait after poweron for all servers being effectively operational
servers_StartDelay = 480

################################################################################
#
# Following part is used to control the order to start nodes groups
#
################################################################################

# GROUP <nb simultaneous poweron> <time to wait> <period to wait> <time to
wait after this GROUP>
IO 5 1 5 5
META 5 1 5 5
COMP 5 1 5 5
```

/etc/clustmngt/nsclusterstop.conf Configuration file

```
################################################################################
#
# First Part is used to control the power off safety delay for the Cool
Cabinet Doors
#
################################################################################

# time to wait for before the Cool Cabinet Doors are stopped
coldoor_StopDelay = 120

################################################################################
#
# Second Part is used to controls the power supply of DDN and servers
#
################################################################################

# time to wait after poweroff for all servers being effectively down
servers_StopDelay = 180

# time to wait for ddn processing shutdown
ddnShutdown_Time = 180

# time to wait after poweroff for all powerswitches being OFF
couplets_StopDelay = 30

################################################################################
#
# Following part is used to control the order to stop nodes groups
#
################################################################################

# GROUP <nb simultaneous poweron> <time to wait> <period to wait> <time to
wait after this GROUP>
COMP 5 1 5 5
META 5 1 5 5
IO 5 1 5 5
```

1.6 Checking Nodes after the Boot Phase

This section describes how to use **postbootchecker** to check nodes after boot. **postbootchecker** detects when a Compute Node is starting and runs check operations on this node after its boot phase. The objective is to verify that **CPU** and memory parameters are coherent with the values stored in the **ClusterDB**, and if necessary to update the **ClusterDB** with the real values.

1.6.1 Prerequisites

- **syslog-ng** must be installed and configured as follows:
  - Management Node : Management of the logs coming from the cluster nodes.
  - Compute Nodes : Detection of the compute nodes as they start.
- The **postbootchecker** service must be installed before the RMS service, to avoid jobs being disturbed.

1.6.2 Checking the Compute Nodes

The **postbootchecker** service (**/etc/init.d/postbootchecker**) detects when a Compute Node starts. Whilst the node is starting up, **postbootchecker** runs three scripts to retrieve information about processors and memory. These scripts are the following:

Script name	Description
procTest.pl	Retrieves the number of CPUs available for the node.
memTest.pl	Retrieves the size of memory available for the node.
modelTest.pl	Retrieves model information for the CPUs available on the node.

Then **postbootchecker** returns this information to the Management Node using **syslog-ng**.

1.6.3 Checking the Management Node

On the Management Node, the **postbootchecker** server gets information returned from the Compute Nodes and compares it with information stored in the **ClusterDB**:

- The number of CPUs available for a node is compared with the **nb_cpu_total** value in the ClusterDB.
- The size of memory available for a node is compared with the **memory_size** value in the ClusterDB.
- The CPUs model type for a node is compared with the **cpu_model** value in the ClusterDB.

If discrepancies are found, the ClusterDB is updated with the new values. In addition, the **Nagios** status of the **postbootchecker** service is updated as follows:

- If the discrepancies concern the number of CPUs or the memory size the service is set to **CRITICAL**.
- If the discrepancies concern the model of the CPUs the service is set to **WARNING**.
- If no discrepancies were found, the service is **OK**.

Chapter 2. Discovering Hardware

Important **This chapter only applies to small clusters or to cluster extensions where the Cluster DB Prelaod file is NOT in place.**

This chapter describes the tools to discover, and to add, cluster hardware to the Cluster Database. Some of this hardware, including new **Ethernet** switches and hardware management cards, will also be configured by these tools. These tools may be used when installing the Bull distribution for the first time, or when adding hardware to extend a cluster.

Note Contact Bull Technical Support for information regarding the hardware operations required to extend a cluster.

For a first installation, the following procedures replace the *Database Configuration*, *Update MAC Address in the Cluster Database* and *Configure Ethernet switches* sections in STEPs 2 and 3 of the installation process, described in Chapter 3 in the *Installation and Configuration Guide*.

Important **The physical location data for all equipment added to the cluster database is set by default to a virtual rack by these tools. The Administrator has to use the phpPgAdmin web interface to change and update this information.**

Specifically, this chapter describes:

1. The use of the **initClusterDB** command to initialize the cluster database without the **ClusterDB** preload file being used.
2. The use of the **swtDiscover** command to configure **Ethernet** switches, which are not listed in the cluster database.
3. The use of the **nodeDiscover** command to add new nodes to the cluster database with the configuration of their hardware management cards.
4. The use of the **equipmentRecord** command to update dynamically the Cluster Database with the MAC addresses of Bull Cool Cabinet Doors, Blade Servers (Chassis and inherent Nodes), Nodes and their Hardware Managers.

2.1 Cluster-init.xml Initialization File

Note This paragraph applies only when installing **bullx cluster suite** for the first time on a cluster with no **Cluster DB** preload file in place.

The file **cluster-init.xml** file lists all the parameters used by the **initClusterDB** command to initialize the cluster database. This initialization file, **/etc/clustmngt/discovery-config/cluster-init.xml,** is supplied by default. The structure for the **cluster-init.xml** file is defined by the **/etc/clustmngt/discovery-config/cluster-init.dtd** file.

Edit the **cluster-init.xml** file and add the cluster specific information including **name**, **network IP classes,** and **node profile** definitions. By default, the **cluster-init.xml** file contains the following details:

```
<!DOCTYPE cluster-init SYSTEM "cluster-init.dtd">
<cluster-init>
   <cluster name="clusrhel" mode="100" actif_vlan="false"
   actif_ha="false" actif_crm="false" actif_backbone="false"
   resource_manager="slurm" dbversion="20.5.0"/>
   <ip_nw>
      <network name="admin" type="admin" subnet="13.1.0.0"
      netmask="255.255.0.0" mngt_link="eth0" />
      <network name="ic" type="interconnect" subnet="13.10.0.0"
      netmask="255.255.0.0" />
   </ip_nw>
   <node_profile>
      <profile name="mngt" boot_loader="grub" admin_link="eth0" />
      <profile name="compute" boot_loader="grub" admin_link="eth0" />
   </node_profile>
</cluster-init>
```

2.2 initClusterDB Command

Note This paragraph only applies when installing **bullx cluster suite** for the first time on a cluster with no **Cluster DB** preload file in place and with an empty cluster database.

Prerequisites

The **cluster-init.xml** initialization file must have been customised for the cluster, as described in the section above.

Syntax

 # initClusterDB [–dbname <database name>] [–logfile <logfile name>] [–verbose] [–help]

This command has to be executed as **root** on the Management Node. If no database name is given the **ClusterDB** access information is retrieved from the **/etc/clustmngt/clusterdb/clusterdb.cfg** file.

This command:

1. Inserts data from the cluster database initialization file into the **cluster**, **ip_nw** and **node_profile** tables.
2. Inserts virtual rack equipment in the rack table, so that newly detected equipment can be placed in it.

3. Inserts the Management Node in the node table.
4. Inserts the interconnect interfaces of the local node into the **ic_board** table.
5. Inserts the hardware manager of the local node in the **hwmanager** table.
6. Configures the management link and its alias with the **IP** address of the cluster database.

Important **The local node type is 'Management Node'. The Administrator should use the phpPgAdmin web interface to change and modify this information, as necessary.**

2.3 swtDiscover Command

Note This section only applies to clusters which include **Ethernet** switches that are not defined in the cluster database.

Prerequisites

- Only **CISCO** switches are supported for this command.
- Switches must boot using **DHCP** and retrieve a configuration file using **TFTP** (factory settings).

Syntax

 # swtDiscover –sw_number <number of new switches>[–nw_type <type of network>] [–dbname <database name>] [–logfile <logfile name>] [–verbose] [–help]

This command has to be executed as **root** on the Management Node. If no database name is given the Cluster DB access information is retrieved from the **/etc/clustmngt/clusterdb/clusterdb.cfg** file.

The **sw_number** option is mandatory. This specifies the number of switches that are expected to be discovered, and added to the cluster database.

This command:

1. Inserts new switches into the cluster database.
2. Configures these switches. The parameters used to generate the configuration details are described in the **/usr/lib/clustmngt/ethswitch-tools/data/cluster-network.xml** file.

See The *Configuring Ethernet Switches* section in the *Configuring Switches and Cards* chapter in the *Installation and Configuration Guide*.

3. Updates the **node** and **hwmanager** tables in the cluster database with the connection details for these switches, as necessary.

2.4 nodeDiscover Command

Note This section only applies for a cluster with nodes that are not defined in the cluster database.

Prerequisites

- The nodes must have been configured to use the **PXE** boot environment.
- The nodes must be connected to **Ethernet** switches declared in the cluster database and configured using the **swtAdmin** or **swtDiscover** commands.

Syntax

 # nodeDiscover start|stop|status [–dbname <database name>] [–logfile <logfile name>] [–verbose] [–help]

This command has to be executed as **root** on the Management Node.

If no database name is given the Cluster Database access information is retrieved from the **/etc/clustmngt/clusterdb/clusterdb.cfg** file. The default log file name is **/var/log/node_discover.log**.

The procedure to discover the node(s) consists of the following steps:

1. Start the node discovery procedure.

```
# nodeDiscover start
```

2. Manually reboot the nodes you want to discover.
3. Check that all rebooted nodes have been discovered:

```
# nodeDiscover status
```

   This step may take several minutes; repeat this command until all the rebooted nodes appear in the status report. However, after 5 minutes, if some nodes are still missing, check the error messages in the log file.

4. Stop the discovery procedure:

```
# nodeDiscover stop
```

   This command:

   a. Inserts the hardware manager for the discovered nodes in the **hwmanager** table.
   b. Inserts the discovered nodes as Compute Nodes in the node table.
   c. Inserts the interconnect interfaces of the discovered nodes in the **ic_board** table.
   d. Configures the hardware manager IP address for the discovered nodes.
   e. When the procedure ends (**stop** action) all the configuration files on the Management Node are regenerated (you will be asked interactively for each service).

Note The tool updates the **MAC** address information only, and configures the hardware manager **IP** address for nodes that are already in the cluster database (following the connection to the management network).

mportant **The node type and profile is always 'Compute'. The Administrator should use the phpPgAdmin web interface to change and modify this information, as necessary.**

2.5 equipmentRecord Command

The **equipmentRecord** command allows the Cluster Database to be updated dynamically with the MAC addresses of various types of hardware equipment: Bull Cool Cabinet Door, Blade Server (Chassis and Nodes), Nodes and their Hardware Managers. This command collects the MAC addresses from the Ethernet switches. It requires that the Ethernet switches are configured and the target equipment is connected both electrically and by Ethernet.

Syntax

The **equipmentRecord** command supports the following types of equipment:

- **bladeS** (Blade Server)
- **coolCD** (Cool Cabinet Door)
- **node**.

The syntax for each type is as follows:

/usr/sbin/equipmentRecord bladeS {--cmc <cmcname> | --all }
[--dbname <database name>] [--logfile <logfile name>] [--verbose] [--help]

/usr/sbin/equipmentRecord coolCD [--dbname <database name>]
[--logfile <logfile name>] [--verbose] [--help]

/usr/sbin/equipmentRecord node --action { start|stop|status }
[--dbname <database name>] [--logfile <logfile name>] [--verbose] [--help]

--cmc <cmcname>	Specify a CMC (Chassis Management Controller) name. Used only with the **bladeS** type.
--all	Specify all Blade chassis in the cluster. Used only with the **bladeS** type.
--action { start|stop|status }	Specify the record action to be performed on the nodes. Used only with the **node** type.
[--dbname <database name>]	Specify the Database name other than the default. Default value: **clusterdb**
[--logfile <logfile name>]	Specify Log File other than the default. Default value: **/tmp/equipementRecord.log**
[--verbose]	Turn to verbose mode.
[--help], [-h]	Display the **equipmentRecord** command usage.

Chapter 3. Administrating the Cluster

This chapter describes the following topics:

- 3.1 *Managing Consoles through Serial Connections (conman, ipmitool)*
- 3.2 *Managing Hardware*
- 3.3 *Using Argos to maintain the cluster*
- 3.4 *Collecting Information for Resolving Problems*

3.1 Managing Consoles through Serial Connections (conman, ipmitool)

The serial lines of the servers are the communication channel to the firmware and enable access to the low-level features of the system. This is why they play an important role in the system **init** surveillance, or in taking control if there is a crash or a debugging operation is undertaken.

The serial lines are brought together with Ethernet/Serial port concentrators, so that they are available from the Management Node.

- **ConMan** can be used as a console management tool.
  See 3.1.1 *Using ConMan*.
- **ipmitool** allows you to use a Serial Over Lan (**SOL**) link.
  See 3.1.2 *Using ipmi Tools*.

Note Storage Units may also provide console interfaces through serial ports, allowing configuration and diagnostics operations.

3.1.1 Using ConMan

The **ConMan** command allows the administrator to manage all the consoles, including server consoles and storage subsystem consoles, on all the nodes. It maintains a connection with all the lines that it administers. It provides access to the consoles and uses a logical name. It supports the key sequences that provide access to debuggers or to dump captures (Crash/Dump).

ConMan is installed on the Management Node.

The advantages of ConMan with a simple telnet connection are as follows:

- Symbolic names are mapped per physical serial line.
- There is a log file for each machine.
- It is possible to join a console session or to take it over.
- There are three modes for accessing the console: monitor (read-only), interactive (read-write), broadcast (write only).

Syntax:

conman <OPTIONS> <CONSOLES>

-b	Broadcast to multiple consoles (write-only).
-d HOST	Specify server destination. [127.0.0.1:7890]
-e CHAR	Specify escape character. [&]
-f	Force connection (console-stealing).
-F FILE	Read console names from file.
-h	Display this help file.
-j	Join connection (console-sharing).
-l FILE	Log connection output to file.
-L	Display license information.
-m	Monitor connection (read-only).
-q	Query server about specified console(s).
-Q	Be quiet and suppress information messages.
-r	Match console names via **regex** instead of **globbing**.
-v	Verbose mode.
-V	Display version information.

Once a connection is established, enter "**&.**" to close the session, or "**&?**" to display a list of currently available escape sequences.

See the **conman** man page for more information.

Examples:

- To connect to the serial port of node `bull47`, run the command:

```
conman bull47
```

Configuration File:

The **/etc/conman.conf** file is the conman configuration file. It lists the consoles managed by conman and configuration parameters.

The **/etc/conman.conf** file is automatically generated from the ClusterDB information. To change some parameters, the administrator should only modify the **/etc/conman-tpl.conf** template file, which is used by the system to generate **/etc/conman.conf**. It is also possible to use the **dbmConfig** command. See the *Cluster Data Base Management* chapter for more details.

See the **conman.conf** man page for more information.

Note The **timestamp** parameter, which specifies the watchdog frequency, is set to 1 minute by default. This value is suitable for debugging and tracking purposes but generates a lot of messages in the **/var/log/conman** file. To disable this function, comment the line `SERVER timestamp=1m` in the **/etc/conman-tpl.cfg** file.

3.1.2 Using ipmi Tools

The **ipmitool** command provides a simple command-line interface to the **BMC** (Baseboard Management Controller).
To use a **SOL** (Serial Over Lan) interface, run the following command:

```
ipmitool –I lanplus –C 0 –U <BMC_user_name> –P <BMC_password>
–H <BMC_IP_Address> sol activate
```

`BMC_user_name`, `BMC_password` and `BMC_IP_Address` are values defined during the configuration of the BMC and are taken from those in the **ClusterDB**. The standard values for user name/password are `administrator/administrator`.

ipmitool Command Useful Options

Note If **–H** is not specified, the command will address the BMC of the local machine.

- To start a remote SOL session (to access the console):

```
ipmitool -I lanplus –C 0 -H <ip addr> sol activate
```

- To reset the BMC and return to BMC shell prompt:

```
ipmitool -I lanplus -C 0 -H <ip addr> bmc reset cold
```

- To edit the FRU of the machine:

```
ipmitool -H <ip addr> fru print
```

- To edit the network configuration:

```
ipmitool -I lan -H <ip_addr> lan print 1
```

- To trigger a dump (signal INIT):

```
ipmitool -H <ip addr> power diag
```

- To power down the machine:

```
ipmitool -H <ip addr> power off
```

- To perform a hard reset:

```
ipmitool -H <ip addr> power reset
```

- To display the events recorded in the System Event Log (SEL):

```
ipmitool -H <ip addr> sel list
```

- To display the MAC address of the BMC:

```
ipmitool -I lan -H <ip addr> raw 0x06 0x52 0x0f 0xa0 0x06 0x08 0xef
```

- To know more about the **ipmitool** command, enter:

```
ipmitool –h
```

3.2 Managing Hardware

3.2.1 Managing Nodes and CMC using nsctrl

The **nsctrl** command carries out various actions related to hardware. This command must be launched from the Management Node.

The actions can be performed on any type of node (Compute Node, I/O Node, etc.) except the Management Node. All **nsctrl** actions support the CMC (Chassis Management Controller), except the dump, identify and temperature actions.

mportant **The actions performed on a particular chassis also operate on the nodes included in this chassis. Pay attention when running the poweron, reset, poweroff and poweroff_force actions.**

Usage

/usr/sbin/nsctrl [options] action nodes

Actions

dump
ping
poweron
poweroff
poweroff_force
reset
status
temperature
locate
identify --start|--stop
infoNS56 Show actions specific to NovaScale 5XXX/6XXX Series

General Options

--help, -h	Display this help message
--dbname name	Specify database name
--group, -g	Specify a group of nodes
--force, -f	Do not ask for confirmation or state checking
--verbose, -v	Verbose mode
--debug	Debug mode (more than verbose)
--only_test, -o	Testing mode, no action performed
--time, -t	Time to wait between each interval
--interval, -I	Specify the number of calls made before the waiting time (**--time** option).

Specifying nodes

The nodes are specified as follows: **basename[i,j-k]** or **--group NAME**.
If no nodes are explicitly specified, **nsctrl** uses the nodes defined by the **--group** option.

Examples

- To power off node `ns1`, enter:

```
nsctrl poweroff_force ns1
```

- To ping node `ns1`, enter:

```
nsctrl ping ns1
```

- To start the LED identification on the node **`clusrhel9`**, enter:

```
# nsctrl identify --start clusrhel9
```

```
clusrhel9 : Chassis identify interval: 250 seconds
```

- To stop the LED identification on the node **`clusrhel9`**, enter:

```
# nsctrl identify --stop clusrhel9
```

```
clusrhel9 : Chassis identify interval: off
```

- To locate the nodes `clusrhel8` and `clusrhel9` in the computer center, enter:

```
nsctrl locate clusrhel[8-9]
```

```
Hostname     Rack Name  Rack Level   Rack Model   Rack X-Y Coordinates
clusrhel8:   R_COMP_1   F            COMPUTE      Z-101
clusrhel9:   R_COMP_1   G            COMPUTE      Z-101
```

3.2.2 Managing PDUs using nsctrl or clmpdu

Two commands, **nsctrl** and **clmpdu**, are available for controlling and obtaining status information from the **PDU**s.

Usage

/usr/sbin/nsctrl --pdu [options] action equipment_name

/usr/sbin/clmpdu [options] action equipment_name

Actions

ping
poweron
poweroff
status

Options specific to PDU equipment

--pdu	Specify a Power Distribution Unit
--version	SNMP version (default is 1, supported are 2c and 3)

--community	SNMPv1 community (defaults are 'public' (action 'status') and 'private' (actions 'poweron' and 'poweroff'))
--user	SNMPv3 user
--level	SNMPv3 SecurityLevel (noAuthNoPriv|authNoPriv|authPriv)
--authPass	SNMPv3 authentification passphrase (15 to 32 ASCII characters)
--privPass	SNMPv3 privacy passphrase (15 to 32 ASCII characters, different than --authPass)
--authPro	SNMPv3 authentification protocol (default is 'MD5')
--privPro	SNMPv3 privacy protocol (default is 'DES')

nsctrl versus clmpdu

The **nsctrl** command will ask for confirmation when using the **poweron** and **poweroff** actions, whilst the **clmpdu** command is straightforward and does not ask for a confirmation.

See the syntax differences between these commands in the examples below:

```
nsctrl poweron --pdu talim1
```

```
clmpdu poweron talim1
```

mportant **- Verify that equipment (Service Racks, Storage, etc.) plugged into the PDUs can be safely powered on or off before using the nsctrl or clmpdu commands.**

- When a PDU is powered off all PDU electrical outlets are turned off immediately - as shown in the examples below

Examples:

- To ping `talim1` equipment, enter:

```
# nsctrl ping --pdu talim1
```

```
PING talim1 (13.1.0.41) 56(84) bytes of data.
--- talim1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.992/2.992/2.992/0.000 ms
```

- To get `talim1` equipment status with snmpV3 used, enter:

```
# nsctrl status --pdu talim1 --version 3 --user root --level authPriv
--authPass 0123456789012345 --privPass 01234567890123456
```

```
Power Distribution Unit: talim1, MODEL: "AP7922", Serial Nb:
"ZA0904000484", Firm Rev: "v3.5.7"
Outlet1 power: On(1)
[...]
Outlet16 power: On(1)
```

- To power on the `talim1` equipment with snmpV3 used, enter:

```
# nsctrl poweron --pdu talim1 --version 3 --user root --level authPriv
--authPro MD5 --authPass 0123456789012345 --privPro DES --privPass
01234567890123456
```

```
Confirm your request: poweron on talim1 (y/n)?
y
Outlet1 power: immediateOn(1)
[...]
Outlet16 power: immediateOn(1)
```

- To power on the `talim1` equipment with snmpV1 used, enter:

```
# nsctrl poweron --pdu talim1
```

```
Confirm your request: poweron on talim1 (y/n)?
y
Outlet1 power: immediateOn(1)
[...]
Outlet16 power: immediateOn(1)
```

- To power off the `talim1` equipment with snmpV3 used, enter:

```
# nsctrl poweroff --pdu talim1 --version 3 --user root --level
authPriv --authPass 0123456789012345 --privPass 01234567890123456
```

```
Confirm your request: poweroff on talim1 (y/n)?
y
Outlet1 power: immediateOff(2)
[...]
Outlet16 power: immediateOff(2)
```

- To power off the `talim1` equipment with snmpV3 used using the **clmpdu** command, enter:

```
# clmpdu poweroff talim1 --version 3 --user root --level authPriv --
authPass 0123456789012345 --privPass 01234567890123456
```

```
Outlet1 power: immediateOff(2)
[...]
Outlet16 power: immediateOff(2)
```

3.2.3 Using Remote Hardware Management CLI (BSM Commands)

The Remote Hardware Management **CLI** (Command Line Interface) is a set of commands that perform hardware tasks on Bull Extreme Computing clusters, these are also known as **BSM** Commands. These commands provide the Administrator with an easy way to automate scripts to power on/off and to get hardware information about the nodes.

See Refer to **Bull System Manager** documentation for more information about the Remote Hardware Management CLI.

3.2.4 Using nsfirm command

The **nsfirm** command is used for various maintenance operations, such as obtaining the **BIOS** or **BMC** version, upgrading the firmware, flashing the BIOS, etc.

Usage

/usr/sbin/nsfirm [options] action nodes

Options

--help, -h	Display this help message
--dbname name	Specify database name
--group, -g	Specify a group of nodes
--force, -f	Do not ask for confirmation or state checking
--verbose, -v	Verbose mode
--debug	Debug mode (more than verbose)
--only_test, -o	Testing mode, no action performed
--File <upgrade_file>	Specify the BIOS or BMC upgrade file
--SDR <<platform>-sdr.dat >	Specify the Sensor Data Records parameters file to update in the BMC. "**platform**" equals either **r421**, **r422** (for bullx R422 and R422 E1 severs) or **R423**. Example: **r421-sdr.dat**.

Specifying nodes

nodes	Specify the nodes in the form: **basename[i,j-k]** or **--group NAME**

Specific to R4xx Series machines

flash_bios	The **--File** option must be specified (R421-422)
upgrade_bmc	The **--File** and the **--SDR** options must be specified as indicated in the following table:

nsfirm command	Mandatory option(s)	Supported servers	BMC Type
nsfirm flash_bios	--File	bullx R422E2, R423E2, R425E2	
nsfirm upgrade_bmc	--File --SDR	bullx R421E1, R422E1, R423E1	SIMSO
nsfirm upgrade_bmc	--File	bullx R423E2	SIMSO
nsfirm upgrade_bmc	--File	bullx R422E2, R423E2, R424E2, R425E2	Winbond

3.3 Using Argos to maintain the cluster

Argos is a cluster maintenance product integrated into the Extreme Computing environment which:

- Records the composition of, and changes to, the cluster and its components, from a hardware and software perspective.
- Monitors the behaviour of software components and hardware equipment, including nodes, Compute Nodes, I/O Nodes, Switches, and Storage sub-systems.
- Records the time availability of the hardware equipment.
- Keeps a record of any incidents or problems which may occur.

See The Argos documentation for more information.

3.4 Collecting Information for Resolving Problems

The **hpcsnap** tool collects Extreme Computing cluster information. This information can be sent to Bull Support and used for problem analysis.

To facilitate the analysis, the **hpclog** tool can extract lines corresponding to a given period.

The package is delivered as an alpha version (0.2.5) in the **XHPC** media.

Installation

```
yum localinstall /release/XBAS5V<version_nb>/XHPC/BONUS/hpcsnap-0.2.5-0.noarch.rpm
```

Usage

Using **hpcsnap** with default options:

```
# cd /usr/local/hpcsnap.0.2.5

# ./hpcsnap
```

If you have opened a request to Bull Support System (D1 or PARM ticket) use the ticket number as TAG parameter with the **–t** option, as follows:

```
# ./hpcsnap -t <TAG>
```

To list all available options enter:

```
# ./hpcsnap -h
```

Reporting Problems

If you find problems using **hpcsnap**, please send an email to team-linux@support.frec.bull.fr

Chapter 4. Managing System Logs

This chapter describes **syslog-ng** and how it can be configured.

4.1 Introduction to syslog-ng

For security and tracking purposes, and also to facilitate the administration of the cluster, all the system logs are centralized on the Management Node. There are two ways to send system log information to the Management Node:

- The logs are collected on each node, using standard mechanisms for archival and log file permutation. Various utilities ensure compression, transfer and archival of these log files on the Management Node in asynchronous mode. A centralized operation is performed on the Management Node, in order to extract and search events according to the criterion required, for example date, type, gravity, and so on.

  This asynchronous process facilitates curative actions for the incidents that have occurred on the cluster.

- Some events are immediately reported to the Management Node. Filters are used, which specify the type and gravity level of the events that have to be transferred immediately.

  This synchronous process instantaneously gives the Administrator a global view of system events.

syslog-ng (Syslog New Generation) is the powerful system log manager used on Bull Extreme Computing clusters to manage cluster system logs and includes the following features:

- The ability to filter messages based on content using regular expressions.
- Encoding and authentication of the network traffic.
- Forwarding logs using TCP and UDP protocols.
- Log compression.

See http://www.balabit.com/support/documentation/?product=syslog-ng&type=guide&language for more documentation on **syslog-ng**.

4.2 Configuring syslog-ng

syslog-ng is installed on the cluster using the default configuration. The scripts used to transfer log files are also installed. Administrators can modify the default configuration according to their needs.
The **/etc/syslog-ng/syslog-ng.conf** file contains the configuration parameters for syslog-ng. This file is divided into five sections:

options section	General options
source section	Source events
destination section	Log destinations
filter section	Filter definitions
log section	Actions to be performed on messages

4.2.1 options Section

All general parameters may be configured in the options section. An example is below:

```
# Start of options area
options {
 sync (0);      # Number of events before writing in the logs
 time_reopen (10);   # Wait 10s before reconnecting if the connection
     failed. Used when logs are centralized through network
 #time_reap (number);# Closes a log file that is not accessed after
     "number" seconds
 log_fifo_size (1000); # number of event lines stored, before writing them.
     Enables events to be taken quickly into account
     and to free the process that has generated them.
 long_hostnames (off);  # Usage of long names
 use_dns (no)   # Usage of DNS to find addresses
 use_fqdn (no); # Usage of machine short name
 owner("root"); # logs owner
 group("root"); # logs group
 perm("644");   # logs rights mask
 keep_hostname (yes);#
 create_dir (yes);   # Create directories for log storage
 use_time_recv(no);  # Local time will be used instead of the time written
in the logs
 #gc_idle_threshold(100); # The garbage collector is started after 100
       events if syslog-ng is inactive.
 #gc_busy_threshold(100); # The garbage collector is started after 3000
events if syslog-ng is active.
};
```

4.2.2 source Section

The source section defines the log source from the following: network, local files, peripheral, pipe, stream.

Syntax

source <identifier>
{source-driver(params); source-driver(params); etc.};

For example, the following lines are suitable for a Linux system. They enable the **/dev/log** stream to be read and also to receive syslog-ng internal messages, and to handle kernel start messages:

```
source src {
unix-stream("/dev/log");
internal();
file("/proc/kmsg");
};
```

Possible sources are as follows:

unix-stream(<filename>) Stream pipes (used in Linux).

file(<filename>) File data (Linux kernel messages for example).

pipe(<filename>) Named pipes (for interfacing with Nagios for example).

tcp(<ip>,<port>) and **udp(<ip>,<port>)** To listen on an address and a port.

internal() syslog-ng internal messages.

4.2.3 destination Section

This section defines the destination of the logs.

Syntax

destination <identifier>
{ destination-driver(params); destination-driver(params); etc.};

The possible destinations are the following ones:

file(<filename>) To send to a file.

tcp(<ip>,<port>) and **udp(<ip>,<port>)**
To send the logs on the network to another machine.

unix-stream(<filename>) To send to stream pipes (used in Linux).

userttyr(<user>) To send to the <user > consoles, but only if this user is connected. You can use the "*" character to specify that the messages have to be sent to all users.

program(<commandtorun>) To send towards a program.

Examples

You can specify several destination directives in a destination section, as in the following example:

```
destination debug {file("/var/log/debug.log"); };
destination messages {file("/var/log/messages.log"); };
destination console {usertty("root"); };
destination xconsole {pipe("/dev/xconsole"); };
destination mail2admin {program("/usr/bin/MailToAdmin"); };
destination full{
file("/dev/tty12");
file("/var/log/full.log" log_fifo_size(2000));
};
```

Note You can add specific options such as `log_fifo_size(2000)` as shown in the example above.

In the following example, all the logs will be sent to the Management Node, whose address is `192.168.0.100`:

```
destination central_log {tcp ("192.168.0.100" port(514); }
```

Using Macros

It may be useful to use macros to set intelligible names for your destination files. Predefined macros exist, such as FACILITY, PRIORITY or LEVEL, DATE, FULLDATE, ISODATE, YEAR, MONTH, DAY, HOUR, MIN, SEC, FULLHOST, HOST. Some examples are below:

```
destination full {
file("/dev/tty12");
file("/var/log/full_$DAY-$MONTH-$YEAR.log"
owner("root")
group("adm")
perm(0640));
};
```

```
destination hosts {
file("/var/log/HOSTS/$HOSTS/$FACILITY/$YEAR/$MONTH/$DAY/$FACILITY$YEAR
$MONTH$DAY"
owner("root")
group("adm")
perm(0600)
dir_perm(0700)
create_dirs(yes));
};
```

Note Do not forget to remove or archive older files regularly.

4.2.4 filter Section

This section describes the filtering mechanism for events.

Syntax

filter <identifier> {expression; };

The filters are defined by the following keywords:

facility(facility[,facility])	To filter by type.
level(pri[,pri1, .. pri2 [,pri3]])	To filter by priority or level.
program(regexp)	To filter by the name of the program that has generated the message.
host(regexp)	To filter by the regular expression of the name of the host that has sent the message.
match(regexp)	To filter by a regular expression.
filter(filtername)	To use another filter.

All keywords may be used several times. The expressions can contain the AND, OR and NOT operators.

Examples

```
filter f_iptables { match("IN=.*OUT=.*MAC=.*"); };
filter f_snort { match("snort: "); };
filter f_full { not filter(f_snort) AND NOT filter(f_iptables); };
filter f_messages { level(info..warn) AND NOT facility(auth, authpriv,
mail, news); };
```

4.2.5 log Section

In this section you define how the messages will be processed using source, destination and filters commands defined in the previous sections.

Syntax

log { source(s1); source(s2); ...
filter(f1); filter(f2); ...
destination(d1); destination(d2);
flags(flag1[, flag2...]; };

Examples

```
log { source(src);
filter(f_news); filter(f_notice);
destination(newsnotice);
};
log { source(src);
destination(full);
};
```

Chapter 5. Monitoring the System and Devices

This chapter describes the monitoring for the following devices:

- 5.1 *Monitoring the System*
- 5.2 *Getting Information about Storage Devices (lsiocfg)*
- 5.3 *Checking Device Power State* (pingcheck)
- 5.4 *Setting Up Outlet Air Temperature*

5.1 Monitoring the System

This section describes tools that can be used for system monitoring.

5.1.1 Time

The first determinant to find is the run-time for a specific operation; this will be used as a yardstick in the optimization process. Different benchmark operations, similar to those defined in the call to tender, can be used.

The **time** command is used to measure the duration of execution for a particular operation. The execution time is reported in terms of user CPU time, system CPU time, and real time.

The **etime** function is used to give the time of execution for a particular part of the application program.

5.1.2 IOstat

The **iostat** Linux command is used for monitoring system input/output device loading by observing the time the devices are active in relation to their average transfer rates. The **iostat** command generates reports that can be used to change a system's configuration to better balance the input/output load between physical disks.

Performance problems may be the result of too many files being repeatedly opened, read and written to, and then closed. This type of problem is indicated by increasing seek times and may be identified using **iostat.**

The first report generated by the **iostat** command provides statistics for the time elapsed since the system was first booted. Each subsequent report covers the period of time since the previous report. The interval parameter stipulates the time period in seconds for each report.

The count parameter may be used with the interval parameter. These determine the number of reports generated and the time period for each report. If the interval parameter is used without the count parameter, the **iostat** command generates reports continuously.

All I/O statistics are collected each time the **iostat** command runs. The report consists of a CPU header row followed by a row of CPU statistics. On multiprocessor systems, CPU statistics are calculated system-wide as averages among all processors. A device header row is displayed followed by a line of statistics for each device that is configured.

The **iostat** command generates two types of reports, the CPU Utilization report and the Device Utilization report.

- On multiprocessor systems the CPU Utilization Report provides the CPU values which are global averages for all processors.
- The Device Utilization Report provides statistics either by physical device or by partition.

Examples

The following command displays four reports of extended statistics at two second intervals.

```
iostat -x 2 4
```

The following command displays six reports of extended statistics at two second intervals for devices **hda** and **hdb**.

```
iostat -x hda hdb 2 6
```

For more information on the formats of the reports and the commands which are available refer to the man page for **iostat**, alternatively look at http://linuxcommand.org/

5.1.3 dstat

dstat overcomes some of the limitations of iostat. The **dstat** command can be used to monitor systems during performance tuning tests, benchmarks, or troubleshooting. This command allows you to view all of your system resources instantly. You can compare disk usage in combination with interrupts from IDE controllers, or compare the network bandwidth numbers directly with the disk throughput (in the same interval).

dstat allows you to aggregate block device throughput for a certain disk set or network set, so that you can see the throughput for all the block devices that make up a single file system or storage system.

By default, dstat's output is viewed in real-time, the data being displayed in coloured columns. However, it can also be saved in a file in a **CSV** format that can be imported into **Gnumeric** or **Excel** so that the data can be viewed graphically. The counters can be configured so that they appear in the order that makes the most sense for your cluster.

dstat Plugins

Dstat includes external plugins for dedicated counters. It is open source and written in **python** allowing new specific counters to be developed for your cluster. The plugins include the following:

dstat_app	The most expensive process on the system
dstat_freespace	See the disk usage per partition
dstat_nfs3	The NFS v3 client operations
dstat_nfsd3	The NFS v3 server operations
dstat_postfix	Counters of the different queues (needs postfix)
dstat_thermal	CPU temperature

dstat performance impact

Before running any tests check what impact **dstat** in terms of resource usage. Use the **-t** option together with the **–debug** option to examine performance time variations, according to whether or not a plugin is loaded. If the impact is higher than expected, then reduce the number of stats or remove the expensive stats.

See http://dag.wieers.com/home-made/dstat/ for more information

5.2 Getting Information about Storage Devices (lsiocfg)

lsiocfg is a tool used for reporting information about storage devices. It is mainly dedicated to external storage systems (DDN and FDA disk arrays) and their dedicated Host Board Adapters (Emulex FC adapters), but it can also be used with internal system storage (system disks) and their Host Board Adapters tools.

Reported information is related to several inventories:

- Host Board Adapters (-c flag)
- Disks (-d flag)
- Disk partitions (-p flag)
- Disk usages.

5.2.1 lsiocfg Command Syntax

According to needed information, **lsiocfg** can be used with options related to each inventory.

- **lsiocfg [-P] [-v] -c [HBAs IDs]**
  Gives information about all SCSI controllers. If HBAs IDs are specified, only applies to this list of HBAs.
- **lsiocfg [-P] [-v] -d [-u] [devices names]**
  Gives information about SCSI devices. [-u] has to be used to display non disk devices. If devices are specified, only applies to this list of devices.
- **lsiocfg -p**
  Displays partitions.
- **lsiocfg [-P] [-v] -a**
  Dsplays all (= -cdp).
- **lsiocfg [-r user] -n remote node [-P] [-v] [-c|-d|-a]**
  Gives information from remote node about controllers/disks.
- **lsiocfg -M [devices names]**
  Gives information about SCSI devices usage.
- **lsiocfg <-l|-L> <wwpn>**
  Reports WWPN owner. The –l flag uses **/etc/wwn** file, and the –L flag uses cluster manager database.
- **lsiocfg <-w|-W>**
  Displays all WWPN owners. The –w flag uses **/etc/wwn** file, and the –W flag uses cluster manager database.

General flags

-P No headers (before -[a|c|d] commands).

-v Verbose (before -[a|c|d] commands). WWPN verbose information is extracted from /etc/wwn file.

-h Help message. Exclusive with other options.

-V Display the version. Exclusive with other options.

Online help and a man page provide information about lsiocfg usage.

5.2.2 HBA Inventory

Using the lsiocfg HBA inventory option, you can get basic information about Host Board Adapters:

- model,
- link up or down.

When getting HBA inventory in verbose mode, more details are available:

- firmware levels,
- serial number,
- WWNN and WWPN (for fibre channel HBAs).

Example

```
# lsiocfg -cv
```

```
------ HOST/CHANNEL INVENTORY ---------------------------------------
Host   Driver     Unique_id Cmd/Lun HostQ  State               Model
-------------------------------------------------------------------

host0  mptbase    0         7       -                         -
host1  mptbase    1         7       -                         -
host2  lpfc       0         30      -     LINK_UP             LP11000
        DRV=8.0.30_p1
        FW=2.10A7 (B2D2.10A7)
        Bus-Number=26
        SN=VM53824841
        Host-WWNN=20:00:00:00:c9:4b:e7:02
        Host-WWPN=10:00:00:00:c9:4b:e7:02
        FN=20:00:00:00:c9:4b:e7:02
        speed=2 Gbit
host3  usb-storage 0         1       -                          -
```

5.2.3 Disks Inventory

Using the lsiocfg Disk inventory option, you can get basic information about the available disks:

- system location
- vendor
- state
- disk size

When getting the disk inventory in verbose mode, more details are shown:

- model
- serial number
- firmware revision
- WWPN (fiber channel devices).

```
# lsiocfg -dv
```

```
-------------------------------------------------------------------------------

   ----- DISK INVENTORY ------------------------------------------------------
   Dev   Location  Maj:Min  Vendor        state    Size (MB) QueueDepth  Lname
     (location= Host:Channel:Id:LUN)
   -------------------------------------------------------------------------
   sdb   0:0:10:0  8:16     SEAGATE       running  286102    31
           MODEL=SEAGATE ST3300007LC
           FWREV=0003
           SERIAL=3KR0KTPH00007547TR0P
           TRANSPORT=SPI
   sdc   0:0:11:0  8:32     SEAGATE       running  286102    31
           MODEL=SEAGATE ST3300007LC
           FWREV=0003
           SERIAL=3KR0KTHM000075475NWC
           TRANSPORT=SPI
   sda   0:0:9:0   8:0      SEAGATE       running  286102    31
           MODEL=SEAGATE ST3300007LC
           FWREV=0003
           SERIAL=3KR0JT0T00007548GUXA
           TRANSPORT=SPI
   sdd   2:0:0:0   8:48     DDN           running  10000     30  /dev/ldn.ddn0.13
           MODEL=DDN S2A 8500
           FWREV=5.20
           SERIAL=02A820510D00
           TRANSPORT=FC
           WWPN=24:00:00:01:ff:03:02:a8
           NAME=unknown
   sde   2:0:0:1   8:64     DDN           running  125000    30  /dev/ldn.ddn0.14
           MODEL=DDN S2A 8500
           FWREV=5.20
           SERIAL=02A820540E00
           TRANSPORT=FC
           WWPN=24:00:00:01:ff:03:02:a8
           NAME=unknown
   sdf   2:0:0:2   8:80     DDN           running  10000    30  /dev/ldn.ddn0.15
           MODEL=DDN S2A 8500
           FWREV=5.20
           SERIAL=03E020570F00
           TRANSPORT=FC
           WWPN=24:00:00:01:ff:03:02:a8
           NAME=unknown
   sdg   2:0:0:3   8:96     DDN           running  125000    30  /dev/ldn.ddn0.16
           MODEL=DDN S2A 8500
           FWREV=5.20
           SERIAL=03E0205A1000
           TRANSPORT=FC
           WWPN=24:00:00:01:ff:03:02:a8
           NAME=unknown

-------------------------------------------------------------------------------
```

5.2.4 Disk Usage and Partition Inventories

These inventories give information about system and logical use of the devices. Such information is mostly used for system administration needs.

5.3 Checking Device Power State (pingcheck)

The **pingcheck** command checks the power state (on or off) of the specified devices.

Usage

pingcheck [options] --Type <device type> command devices

Options

--dbname name	Specify database name.
--debug, -d	Debug mode (more than verbose).
--help, -h	Display **pingcheck** help.
--interval, -i	Specify the number of bsm calls before waiting the period defined by the **--time** option.
--jobs, -j	Number of simultaneous bsm actions (for example, with -j 5 you can run 5 simultaneous **bsmpower** processes). Default: 30.
--only_test, -o	Display the NS Commands that would be launched according to the specified options and action. This is a testing mode, no action is performed.
--time, -t	Time to wait after the number of bsm calls defined by the **--interval** option.
--verbose, -v	Verbose mode.

Parameters

--Type <device type>	Type of devices to be «pinged »: **disk_array** or **server**.
command	**on** or **off**.
devices	Specify the name of the devices, using the **basename[i,j-k]** or **lc-like** syntax.

Examples

- The following command verifies that all the power supplies for **disk_array** `10` to `15` are in an **on** state and lists those that are not.

```
pingcheck --Type disk_array on da[10-15]
```

- The following command verifies that servers `nova5` to `7` are in `off` state and lists those that are not.

```
pingcheck --Type server off nova[5-7]
```

5.4 Setting Up Outlet Air Temperature

Use the GUI Console of the Bull Cool Cabinet Door to modify the default outlet air temperature value of the Computer Centre.

See The *Cool Cabinet Door Console User's Guide* for details.

Chapter 6. Debugging Tools

This chapter describes the following debugging tasks:

- 6.1 *Modifying the Core Dump Size*
- 6.2 *Identifying InfiniBand Network Problems*
- 6.3 *Using dump tools with RHEL5 (crash, proc, kdump)*
- 6.4 *Configuring systems to take dumps from the Management Network*
- 6.5 *Identifying problems in the different parts of a kernel*

6.1 Modifying the Core Dump Size

By default, the maximum size for core dump files for Bull Extreme Computing clusters is set to 0, which means that no resources are available and core dumps cannot be done. In order that core dumps can be done the values for the **ulimit** command have to be changed.

For more information refer to the options for the **ulimit** command in the **bash** man page.

6.2 Identifying InfiniBand Network Problems (ibtracert)

ibtracert uses Subnet Manager Protocols (**SMP**) to trace the path from a source GID/LID to a destination GID/LID. Each hop along the path is displayed until the destination is reached or a hop does not respond. By using the **-mg** and/or **-ml** options, multicast path tracing can be performed between the source and destination nodes.

Syntax

ibtracert [options] <src-addr> <dest-addr>

Flags

-n Simple format; no additional information is displayed.

-m <mlid> Show the multicast trace of the specified **mlid**.

Examples

- To show trace between lid 2 and 23, enter:

```
ibtracert 2 23
```

- To show multicast trace between lid 3 and 5 for mcast lid 0xc000, enter:

```
ibtracert -m 0xc000 3 5
```

Output

The output for a command between two points is displayed in both hexadecimal format and in human-readable format – as shown in the example below for the trace between the two lids 0x22 and 0x2c. This is very useful in helping to identify any port/switch problems in the **InfiniBand** Fabric.

```
ibtracert 0x22 0x2c
```

```
>From ca {0008f10403979958} portnum 1 lid 0x22-0x22 "lynx13 HCA-1"
[1] -> switch port {0008f104004118e2}[8] lid 0x4-0x4 "ISR9024D Voltaire"
[13] -> switch port {0008f104004118e8}[16] lid 0x3-0x3 "ISR9024D-M Voltaire"
[21] -> switch port {0008f104004118e4}[13] lid 0x1-0x1 "ISR9024D Voltaire"
[4] -> ca port {0008f10403979985}[1] lid 0x2c-0x2c "lynx19 HCA-1"
To ca {0008f10403979984} portnum 1 lid 0x2c-0x2c "lynx19 HCA-1"


In short:
=> OUT  lynx13 (lid 0x22 / port 1
=> INTO node switch (lid 0x4) / port 8
=> OUT  node switch (lid 0x4) / port 13
=> INTO top switch  (lid 0x3) / port 16
=> OUT  top switch  (lid 0x3) / port 21
=> INTO node switch (lid 0x1) / port 13
=> OUT  node switch (lid 0x1) / port 4
=> INTO lynx 19 (lid 0x2c) / port 1
```

6.3 Using dump tools with RHEL5 (crash, proc, kdump)

Various tools allow problems to be analysed whilst the system is in operation:

- **crash** portrays system data symbolically using the possibilities provided by the **GDB** debugger. The commands which it offers are system oriented, for example, the list of tasks, tracing function calls for a task which is waiting, etc.

  See the **crash** man page for more information.

- The system file **/proc** may be used to view, and if necessary modify, system information. In particular it can be used to examine system information for different tasks, the state of the memory allocation, etc.

  See the **proc** man page for more information.

- In the event of a system crash, memory will be written to the configured disk location using **kdump**. Upon subsequent reboot, the data will be copied from the old memory and formatted into a **vmcore** file and stored in the **/var/crash/** subdirectory. The end result can then be analysed using the **crash** utility. An example command is shown below.

```
crash /usr/lib/debug/lib/modules/<kernel_version>/vmlinux vmcore
```

6.4 Configuring systems to take dumps from the Management Network

In addition to forcing a dump for a kernel crash, it is possible to force a dump using the **ipmitool** command from the Management Node. This is done as follows:

Add **nmi_watchdog=0** to the kernel boot options in the **/boot/grub/menu.lst** file in order to deactivate the NMI watchdog used by **RHEL**, so that the other NMIs can be put into effect.

An example of the menu.1st file is shown below:

```
kernel /vmlinuz-2.6.18-53.d5.ELsmp ro root=LABEL=/ nmi_watchdog=0
console=tty0 console=ttyS1,115200n8 console=ttyS0,1152,00n8 rhgb quiet
```

Once the system has been restarted the kernel has to be reconfigured so that a panic is launched when an unknown NMI is received. This can be set to happen automatically by configuring the **kernel.unknown_nmi_panic = 1** option in the **/etc/sysct1.conf** file

Alternatively, this can be done manually by using the command.

```
echo 1 > /proc/sys/kernel/unknown_nmi_panic
```

An **NMI** dump may be launched using **IPMI** via the command:

```
ipmitool -H <bmc_address> -U <user_name> -P <pwd> chassis power diag
```

or by using the **nsctrl** command.

See http://kbase.redhat.com/faq/FAQ_105_9036.shtm for more information.

Notes

- If watchdog is still active after the **kernel.unknown_nmi_panic = 1 option** is set the machine will no longer boot.
- There is also a dump button on the back of the **NovaScale R460** series machines, that will launch an NMI dump for these machines.

Further information can be found in the **kdump** man pages.

Important **It is essential to use non-stripped binary code within the kernel. Non-stripped binary code is included in the debuginfo RPM available from:**
http://people.redhat.com/duffy/debuginfo/index-js.html
This package installs the kernel binary in the folder
`/usr/lib/debug/lib/modules/<kernel_version>/`

6.5 Identifying problems in the different parts of a kernel

Various configuration parameters enable traces or additional checks to be used on different kernel operations, for example, locks, memory allocation and so on.

It is usually possible to focus the debug mode on the problematic part of the kernel which has been identified after recompilation. It is also possible to insert code, e.g. **printk**, to help examine the problematic part.

The different compilation tasks for a machine – stopping, starting, resetting, creating a dump, bootstrapping a compiled system and debugging may be carried out from a remote work station, connected to a development machine configured as a DHCP server.

Chapter 7. Troubleshooting the Cluster

Troubleshooting deals with the unexpected and is an important contribution towards maintaining a cluster in a stable and reliable condition. This chapter is aimed at helping you to develop a general, comprehensive methodology for identifying and solving problems on- and off-site.

The following topics are described:

- *7.1 Troubleshooting Node Deployment*
- *7.2 Troubleshooting Storage*
- *7.3 Troubleshooting FLEXlm License Manager*
- *7.4 Troubleshooting the equipmentRecord Command*
- *7.5 Troubleshooting the Bull Cool Cabinet Door*

7.1 Troubleshooting Node Deployment

ksis is the deployment tool used to deploy node images on Bull Extreme Computing systems. This section describes how deployment problems are logged by **ksis** for different parts of the deployment procedure.

7.1.1 ksis deployment accounting

Following each deployment **ksis** take stock of the nodes, and identifies those that have had the image successfully deployed onto them, and those that have not.
This information is listed in the files below, and remains available until the next image deployment:

- List of nodes successfully deployed to - **/tmp/ksisServer/ksis_nodes_list**
- List of nodes not deployed to - **/tmp/ksisServer/ksis_exclude_nodes_list**

When the image has failed to be deployed to a particular node, **Ksis** adds a line in the **ksis_exclude_nodes_list** file to indicate:

a. The name of the node (between square brackets)

b. The consequences of the problem for the node.
   Three states are possible:
   - **not touched** The node was excluded by the deployment with no impact (for the node).
   - **restored** The configuration of the node was modified, but its initial configuration was able to be restored.
   - **corrupt** The node was corrupted by the operation.

c. The circumstance which led to the deployment problem.

Example:

```
[node2] not touched: node is configured-in
```

Most of the time, the information in the excluded node list allows the source of the problem to be identified, without the need for further analysis.

7.1.2 Possible Deployment Problems

There are 2 areas where deployment problems may occur.

7.1.2.1 Pre-check problems

Before the image is deployed, node states are verified in the **ClusterDB** Database, and through the use of **bsm** commands. If there are any problems, the nodes in question will be excluded for the deployment.
The error will be displayed once the deployment has finished, and will also be logged in the **/tmp/ksisServer/ksis_exclude_nodes_list** file.

7.1.2.2 Image transfer problems

Problems may occur during the phase when the image is being transferred onto the target nodes. These problems are logged and centralised by **Ksis** on the Management Node.

The errors will be displayed once the deployment has finished, and will also be logged in the **/tmp/ksisServer/ksis_exclude_nodes_list** file.

ksis image server logs

ksis server logs are saved on the Management Node in
/var/lib/systemimager/overrides/ka-d-server.log

and
Ksis server traces are saved on the Management Node in
/var/lib/systemimager/overrides/server_log

Note Traces are only possible for the ksis server, and for client nodes, if the **ksis deploy** command is executed using the **–g** option.

---

ksis image client logs

ksis client logs on the Management Node in
/var/lib/systemimager/overrides/imaging_complete_<nodeIP>
or
/var/lib/systemimager/overrides/patching_complete_<nodeIP>
or
/var/lib/systemimager/overrides/unpatching_complete_<nodeIP>

and ksis client traces on the Management Node in
/var/lib/systemimager/overrides/imaging_complete_error_<nodeIP>

These traces will only be logged if the deployment error occurs on the client side.

Patch deployment client traces on the Management Node in
/var/lib/systemimager/overrides/patching_complete_error_<nodeIP>
or
/var/lib/systemimager/overrides/unpatching_complete_error_<nodeIP>

The client log files will be used during the post-check phase. **Ksis** client and image server errors are compared in order to identify the source of any problems which may occur.

The trace files are kept for support operations.

7.2 Troubleshooting Storage

This section provides some tips to help the administrator troubleshoot a storage configuration.

7.2.1 Verbose Mode (-v Option)

Some of the storage commands have a **–v** (verbose) option, which provides more output information during the processing of the command.

See *Administrator's Guide* for an inventory of storage commands supporting the **–v** option.

7.2.2 Log/Trace System

Principle

If the verbose mode is not enough, a system of traces can also be configured to obtain more information on some commands. To activate these traces you can set the trace level in the appropriate **/etc/storageadmin/*.conf** file

There are two lines in these files to set the trace. These lines look as follows, where <command_name> is the name of the command to debug:

```
#<command_name>_TRACE_STDOUT_LEVEL =
#<command_name>_TRACE_LOG_FILE_LEVEL =
```

The first line is used to activate traces on stdout, the second one is used to generate traces in a **/tmp/storregister.PID.traces** log file. By default the two lines are in comment.

Note It is recommended to use this trace tool only for temporary debugging because there is no automatic cleaning of the **/tmp/<command_name>.PID.traces** log files.

Four levels of traces are available:

- 4 => TRACE_LEVEL_DEBUG
- 3 => TRACE_LEVEL_INFO
- 2 => TRACE_LEVEL_WARNING
- 1 => TRACE_LEVEL_ERROR

Level 4 is the most verbose level, level 1 traces only error messages.

Note It is not possible to add new commands. All the commands accepting this system of traces are listed in the corresponding ***.conf** file.

See *Administrator's Guide* to identify the right configuration file.

Example

The following example explains how to obtain log file and/or stdout traces on **storregister** command.

1. Find the right **/etc/storageadmin/*.conf** file to modify. In the case of the **storregister** command, it is **storframework.conf** because of the presence of these two lines:

```
# storregister_TRACE_STDOUT_LEVEL =
# storregister_TRACE_LOG_FILE_LEVEL =
```

2. Edit the **storframework.conf** file:
   - Uncomment one of the two previous lines.
   - Choose a level of trace between 1 (lowest) and 4 (highest) level.

   For example, to add traces of debug level (4 = highest level) on stdout only , the **storframework.conf** file must contain the following lines:

```
# STDOUT trace level configuration :
…
storregister_TRACE_STDOUT_LEVEL = 4
…
# log file trace level configuration :
# storregister_TRACE_LOG_FILE_LEVEL =
```

3. Save the **storframework.conf** file.
4. Relaunch **storregister**. New traces will appear on the stdout.

7.2.3 Available Troubleshooting Options for Storage Commands

The following table sums up the available troubleshooting options for the storage commands.

Command	User Command	-v option	Log/Traces	Name of the corresponding .conf File
fcswregister	Yes			
iorefmgmt	Yes			
ioshowall	Yes			
lsiocfg	Yes	Yes		
lsiodev	Yes			
nec_admin	Yes		Yes	nec_admin.conf
nec_stat	Yes			
stordepha	Yes			
storcheck	Yes		Yes	storframework.conf
stordepmap	Yes	Yes		
stordiskname	Yes			
storiocellctl	Yes		Yes	storframework.conf
storioha	Yes			
storiopathctl	Yes		Yes	storframework.conf
stormap	Yes	Yes		
stormodelctl	Yes		Yes	storframework.conf
storregister	Yes		Yes	storframework.conf
storstat	Yes		Yes	storframework.conf
stortrapd	No		Yes	storframework.conf
stortraps	No		Yes	storframework.conf

Table 7-1. Troubleshooting options available for storage commands

7.2.4 nec_admin Command for Bull FDA Storage Systems

The **nec_admin** command is used to manage Bull FDA Storage Systems This command interacts with the FDA CLI. A retry mechanism has been implemented to manage the fact that the CLI may reject commands when overloaded. If, despite default setting, the **nec_admin** command occasionally fails, you may change the timeout and retry values defined in the **/etc/storageadmin/nec_admin.conf** file.

```
# Number of retries in case of iSMserver Busy (Not Mandatory)
retry = 3

# If "retry" is set: time in second between two retries (Not
Mandatory)
rtime = 5

# Timeout value : when timeout is reached, the command is considered
as failed
# If number of retries does not exceed the "retry" value, the
# command is launched again, otherwise it is failed.
cmdtimeout = 300
```

See *Administrator's Guide* for more details about the **nec_admin** command.

7.3 Troubleshooting FLEXlm License Manager

7.3.1 Entering License File Data

You can edit the hostname on the server line (first argument), the port address (third argument), the path to the vendor-daemon on the VENDOR line (if present), or any right half of a string (b) of the form a=b where (a) is all lower case. Any other changes will invalidate the license.
Be cautious when transferring data received by Mailers. Many Mailers add characters at the end-of-line that may confuse the reader about the real license data.

7.3.2 Using the lmdiag utility

The **lmdiag** command analyzes a license file with respect to the SERVER, the FEATUREs, license counts and dates. It may help you to understand problems that may occur. **lmdiag** attempts to checkout all FEATUREs and explains failures. You may run extended diagnostics attempting to connect to the license manager on each port on the host.

7.3.3 Using INTEL_LMD_DEBUG Environment Variable

Setting this environment variable will cause the application to produce product diagnostic information at every checkout.

Daemon Startup Problems.

Cannot find license file. Most products have a default location in their directory hierarchy (or use **/opt/intel/licenses/server.lic**). The environment variable INTEL_LICENSE_FILE names this directory. Startup may fail if these variables are set wrong, or the default location for the license is missing.

No such Feature exists

The most common reason for this is that the wrong license file, or an outdated copy of the file, is being used.

Retrying Socket Bind

This means the TCP port number is already in use. Almost always, this means an **lmgrd.intel** is already running, and you have tried to start it twice. Sometimes it means that another program is using this TCP port number. The number is listed on the SERVER line in the license file as the last item. You can change the number and restart **lmgrd.intel**, but only do this if you do not already have an **lmgrd.intel** running for this license file.

INTEL: cannot initialise

```
(INTEL) FLEXlm version 7.2
(lmgrd) Please correct problem and restart daemons
```

You may be starting the **lmgrd.intel** from the wrong directory, or with relative paths. Use the following lines in the start up and add a full root path to 'INTEL' to the end of the VENDOR line in the license file:

```
cd <installation-directory>
`pwd`/lmgrd.intel -c `pwd`/server.lic -l `pwd`/lmgrd.intel.log
```

License manager: cannot initialize: Cannot find license file

You have started lmgrd.intel on a non-existent file. The recommended way to specify the file for lmgrd.intel to use -c <license>:

```
cd <installation-directory>
`pwd`/lmgrd.intel -c `pwd`/server.lic -l `pwd`/lmgrd.intel.log
```

Invalid license key (inconsistent encryption code for 'FEATURE')

This happens for 3 different reasons:

1. The license file has been typed in incorrectly.

   (Cutting and pasting from email is a safe way to avoid this). Or the data have been altered by the end user. See "Entering License File Data" above.

2. The license is generated incorrectly. Your vendor will have to generate a new license if this is the case.

3. The license vendor has changed encryption seeds (rare).

MULTIPLE vendor-daemon-name servers running

There are 2 lmgrd and vendor-daemons running for this license file. Only one process per vendor-daemon/per node is allowed to run. Sometimes this can happen because the lmgrd was killed with a -9 signal (which should not be done!). The lmgrd was then not able to bring the vendor-daemon process down, so it's still running, although not able to serve licenses.

If lmgrd is killed with a -9, the vendor-daemons also then must be killed with a -9 signal. In general, lmdown should be used.

Vendor daemon cannot talk to lmgrd

This means a pre-version-3.0 lmgrd version is being used with a 3.0+ vendor daemon. Simply use the latest version of lmgrd (MUST be a version equal to or greater than the vendor daemon version). This can also happen if TCP networking does not function on the node where you are trying to run lmgrd (rare).

No licenses to serve

The license file has only 'uncounted' licenses, and these do not require a server. Uncounted licenses have a '0' or 'uncounted' in the 'number-of-licenses' field on the FEATURE line.

Other Starting lmgrd.intel from a remote directory may lead to unknown results. If lmgrd.intel is started from a remote directory the license file line:
`VENDOR INTEL`

Should be modified to include the root directory where the 'INTEL' vendor daemon resides:
`VENDOR INTEL <root-directory-path>`

The lmgrd.intel daemon MUST be started with the -c argument:

```
cd <installation-directory>
`pwd`/lmgrd.intel -c `pwd`/server.lic -l `pwd`/lmgrd.intel.log
```

Application Execution Problems

```
Cannot connect to license server
```

Usually this means the server is not running. It can also mean the server is using a different copy of the license file, which has a different port number than the license file you are currently using indicates. You can use the lmdiag utility to more fully analyze this error.

License Server does not support this Feature

This means the server is using a different copy of the license file than the application. They should be synchronized. This error will also report "UNSUPPORTED" in the debug log file.

Invalid Host

You may be attempting to run the application on a host not listed in the "HOSTID" field of your license. Use lmhostid to find the hostid number for the current host.

```
Cannot find license file.  No such file or directory
Expected license file location: <path>
```

The application was not able to find a license file. It gives you the location(s) where it was looking for a license file.

Check that the named file exists. To use a file at a different location, use the environment variable INTEL_LICENSE_FILE.

No such Feature exists

The license manager cannot find a 'FEATURE' line in the license file.

Feature has expired

Your license has expired. The system time may be set incorrectly. Run the 'date' command to make sure the date is not later than the Expiration Date listed in the license file.

```
<FEATURE name>: Invalid (inconsistent) license key
```

The license-key and data for the feature do not match. This usually happens when a license file has been altered. See "Entering License File Data" above.

System Bootup Problems

For reasons unknown some bootup files (/etc/rc, /sbin/rc2.d, etc) refuse to run lmgrd with the simple commands indicated above. Here are two workarounds:

1. Use 'nohup su username -c 'umask 022;lmgrd -c ...' (It is not recommended to run lmgrd as root; the "su username" is used to run lmgrd as a non-privileged user.)
2. Add 'sleep 2' after the lmgrd command.

7.4 Troubleshooting the equipmentRecord Command

The **equipmentRecord** command supports different equipment types: Blade Server, Cool Cabinet Door and Nodes. Run the command below with the **–h** option:

```
/usr/sbin/equipmentRecord -h
```

The output should list the supported types, as below:

```
. . .
SUPPORTED TYPES:
blades   : Blade Server
coolCD   : Cool Cabinet Door)
node     : Needs additional mandatory action: start|stop|status.
```

If one of these types is not displayed, it may be because a **RPM** is missing. Check that all **RPM**s are installed, using the following commands from the Management Node:

```
rpm –qa | grep coldoor-record (for the Cooled Cabinet Door)
```

The following **RPM** should be listed:

```
coldoor-record-x.x-Bull.x
```

```
rpm –qa | grep inca-record (for the Blade Server)
```

The following **RPM** should be listed:

```
inca-record-x.x-Bull.x
```

```
rpm –qa | grep node-record (for the Nodes)
```

The following **RPM** should be listed:

```
node-record-x.x-Bull.x
```

Note "x" designates the version number, which depends on the release.

7.5 Troubleshooting the Bull Cool Cabinet Door

7.5.1 No Cool Cabinet Door found

1. Check the Cool Cabinet Door is electrically plugged-in.
2. Run the commands:

```
su - postgres
psql -U clusterdb clusterdb
```

```
<Enter Password>

clusterdb=> SELECT e.admin_ipaddr, rp.id, rp.admin_eth_switch_id,
rp.admin_eth_switch_slot, rp.admin_eth_switch_port, rp.admin_ipaddr
FROM rack_port rp, eth_switch e WHERE e.id = rp.id and e.status !=
'not_managed';
```

This should return the Cool Cabinet Doors configured if any, in the following format:

```
admin_ipaddr|id|admin_eth_switch_id|admin_eth_switch_slot
        |admin_eth_switch_port|admin_ipaddr
```

Example:

```
---------------+----+-------------+---------------+----+-------------
172.17.0.210   |  0 |           0 |             0 | 23 | 172.17.0.103
```

It means that the Cool Cabinet Door whose IP address is `172.17.0.103` is connected to switch `172.17.0.210` on port `23`, slot `0`.

3. Check the wiring configuration:
   The Cool Cabinet Doors must be connected to the appropriate switch, as defined in the Cluster Database, and as returned by the previous **psql** command, above.

Chapter 8. Upgrading Emulex HBA Firmware

This chapter describes the following tasks:

- 8.1 *Upgrading Emulex Firmware on a Node*
- 8.2 *Upgrading Emulex Firmware on Multiple Nodes*

8.1 Upgrading Emulex Firmware on a Node

8.1.1 Emulex Core Application kit

Emulex Core Application Kit (elxlinuxcorekit) is a set of low level utilities from **Emulex**.

It can be found on the **Emulex** CDROM shipped with **bullx** servers, or it can also be obtained from the **Emulex** web site.

Following the **elxlinuxcorekit** package installation, you should stop and remove any **Emulex** services to avoid any performance problems when using **HBA**s:

```
service ElxRMSrv stop
service fcauthd stop
chkconfig --del ElxRMSrv
chkconfig --del fcauthd
```

8.1.2 Using lptools

The **lptools** package provides **lpflash**, a high level script used to upgrade the firmware of a set of **Emulex HBA**s. Emulex Core Application kit should be installed before using **lpflash**, otherwise **lpflash** will generate an error message

```
missing command /usr/sbin/hbanyware/hbacmd
check that 'elxlinuxcorekit' package from Emulex is installed
```

The **Emulex** driver (**lpfc** module) has to be loaded when using **lptools** (check with **lsmod**). Firmware updates are available from the **Emulex** Web site.

On a node, you can obtain details of the current **FW** level from all the Emulex HBAs by using the **lsiocfg** tool.

See The *Monitoring Devices* chapter in the *Maintenance Guide* for more information about the **isiocfg** tool.

WARNING

Be sure that FC devices are not being used when the Emulex HBA firmware is upgraded.

8.1.3 lpflash

lpflash flashes **Emulex HBA**s with the specified firmware file. **lpflash** may be used to upgrade in one shot all the **HBA**s on a server.

Syntax

lpflash <-m LP_Model -f path_to_firmware [-v]> | <-h> | <-V>

Flags

-m model	Emulex HBA model to flash (case insensitive)
-f file	Firmware file
-v	Verbose mode
-h	Displays help
-V	Displays version

Example:

```
lpflash –m lp11000 –f  /tmp/bd210a7.all
```

This command will upgrade all LP11000 HBAs to version `2.10A7` firmware.

8.2 Upgrading Emulex Firmware on Multiple Nodes

Emulex firmware can be upgraded in one shot on a set of nodes, by running the **pdcp/pdsh** commands,:

- Use **pdcp** to copy the new firmware file on all the nodes
- Use **pdsh** to run **lpflash** on these nodes.

Example

The following commands copy the **Emulex** firmware file on to the `node1`, `node2` and `node3` nodes, and then upgrade all **Emulex LP11000 HBA**s on these nodes to firmware version 2.10A7:

```
pdcp –w “node1,node2,node3” bd210a7.all /tmp/
pdsh –w “node1,node2,node3” lpflash –m lp11000 –f /tmp/bd210a7.all
```

Chapter 9. Updating the MegaRAID Card Firmware

The **MegaRAID SAS** driver for the **8408E** card is included in the **bullx cluster suite** delivery. The **MegaRAID** card will be detected and the driver for it installed automatically during the installation of the bullx cluster suite.

The **MegaCLI** tool used to update the firmware for the **MegaRAID** card and is available on the **Bull** support CD. The latest firmware file should be downloaded from the **LSI** web site.

Follow the procedure described below to update the firmware:

1. Check the version of the firmware already installed by running the command:

```
/opt/MegaCli -AdpAllInfo -a0
```

This will provide full version and manufacturing date details for the firmware, as shown in the example below:

```
Adapter #0
==============================================================
                    Versions
                ================
Product Name    : MegaRAID SAS 8408E
Serial No       : P088043006
FW Package Build: 5.0.1-0053
                    Mfg. Data
                ================
Mfg. Date       : 01/16/07
Rework Date     : 00/00/00
Revision No     : (

                Image Versions In Flash:
                ================
Boot Block Version : R.2.3.2
BIOS Version       : MT25
MPT Version        : MPTFW-01.15.20.00-IT
FW Version         : 1.02.00-0119
WebBIOS Version    : 1.01-24
Ctrl-R Version     : 1.02-007

                Pending Images In Flash
                ================
None
```

Note The following **MegaRAID** card details are also provided when the **AdpAllInfo** command runs: PCI slot info, Hardware Configuration, Settings and Capabilities for the card, Status, Limitations, Devices present, Virtual Drive and Physical Drive Operations supported by the card, Error Counters, and Default Card Settings.

2. Decompress and extract the firmware by running the command below:

```
unzip ~/lsi/5.1.1-0054_SAS_FW_Image_1.03.60-0255.zip
```

```
Archive:  /root/lsi/5.1.1-0054_SAS_FW_Image_1.03.60-0255.zip
  inflating: sasfw.rom
  inflating: 5.1.1-0054_SAS_FW_Image_1.03.60.0255.txt
 extracting: DOS_MegaCLI_1.01.24.zip
```

3. Update the firmware using the MegaCLI tool using the command below:

```
/opt/MegaCli -adpfwflash -f sasfw.rom -a0
```

```
Adapter 0: MegaRAID SAS 8408E
Vendor ID: 0x1000, Device ID: 0x0411

FW version on the controller: 1.02.00-0119
FW version of the image file: 1.03.60-0255
Flashing image to adapter...
Adapter 0: Flash Completed.
```

4. Reboot the server so that the new firmware is activated for the card.

Appendix A. Tips

A.1. Replacing Embedded Management Board (OPMA) in Bull Cool Cabinet Door

Refer to the *R@ck'n Roll & R@ck-to-Build Installation & Service Guide* and the *Cool Cabinet Door Service Guide* for details on replacing the OPMA board.

mportant The ClusterDB should be updated with new Bull Cool Cabinet Door MAC address. Refer to *Installation and Configuration Guide* for details on the procedure.

Glossary and Acronyms

A

ABI
Application Binary Interface

ACL
Access Control List

ACT
Administration Configuration Tool

ANL
Argonne National Laboratory (MPICH2)

API
Application Programmer Interface

ARP
Address Resolution Protocol

ASIC
Application Specific Integrated Circuit

B

BAS
Bull Advanced Server

BIOS
Basic Input Output System

Blade
Thin server that is inserted in a blade chassis

BLACS
Basic Linear Algebra Communication Subprograms

BLAS
Basic Linear Algebra Subprograms

BMC
Baseboard Management Controller

BSBR
Bull System Backup Restore

BSM
Bull System Manager

C

CGI
Common Gateway Interface

CLI
Command Line Interface

ClusterDB
Cluster Database

CLM
Cluster Management

CMC
Chassis Management Controller

ConMan
A management tool, based on telnet, enabling access to all the consoles of the cluster.

Cron
A UNIX command for scheduling jobs to be executed sometime in the future. A cron is normally used to schedule a job that is executed periodically - for example, to send out a notice every morning. It is also a daemon process, meaning that it runs continuously, waiting for specific events to occur.

CUBLAS
CUDA™ BLAS

CUDA™
Compute Unified Device Architecture

CUFFT
CUDA™ Fast Fourier Transform

CVS
Concurrent Versions System

Cygwin
A Linux-like environment for Windows. Bull cluster management tools use Cygwin to provide SSH support on a Windows system, enabling command mode access.

D

DDN
Data Direct Networks

DDR
Double Data Rate

DHCP
Dynamic Host Configuration Protocol

DLID
Destination Local Indentifier

DNS
Domain Name Server:
A server that retains the addresses and routing information for TCP/IP LAN users.

DSO
Dynamic Shared Object

E

EBP
End Bad Packet Delimiter

ECT
Embedded Configuration Tool

EIP
Encapsulated IP

EPM
Errors per Million

EULA
End User License Agreement (Microsoft)

F

FDA
Fibre Disk Array

FFT
Fast Fourier Transform

FFTW
Fastest Fourier Transform in the West

FRU
Field Replaceable Unit

FTP
File Transfer Protocol

G

Ganglia
A distributed monitoring tool used to view information associated with a node, such as CPU load, memory consumption, and network load.

GCC
GNU C Compiler

GDB
Gnu Debugger

GFS
Global File System

GMP
GNU Multiprecision Library

GID
Group ID

GNU
GNU's Not Unix

GPL
General Public License

GPT
GUID Partition Table

Gratuitous ARP
A gratuitous ARP request is an Address Resolution Protocol request packet where the source and destination IP are both set to the IP of the machine issuing the packet and the destination MAC is the broadcast address xx:xx:xx:xx:xx:xx. Ordinarily, no reply packet will occur. Gratuitous ARP reply is a reply to which no request has been made.

GSL
GNU Scientific Library

GT/s
Giga transfers per second

GUI
Graphical User Interface

GUID
Globally Unique Identifier

H

HBA
Host Bus Adapter

HCA
Host Channel Adapter

HDD
Hard Disk Drive

HoQ
Head of Queue

HPC
High Performance Computing

Hyper-Threading
A technology that enables multi-threaded software applications to process threads in parallel, within each processor, resulting in increased utilization of processor resources.

I

IB
InfiniBand

IBTA
InfiniBand Trade Association

ICC
Intel C Compiler

IDE
Integrated Device Electronics

IFORT
Intel® Fortran Compiler

IMB
Intel MPI Benchmarks

INCA
Integrated Cluster Architecture:
Bull Blade platform

IOC
Input/Output Board Compact with 6 PCI Slots

IPMI
Intelligent Platform Management Interface

IPO
Interprocedural Optimization

IPoIB
Internet Protocol over InfiniBand

IPR
IP Router

iSM
Storage Manager (FDA storage systems)

ISV
Independent Software Vendor

K

KDC
Key Distribution Centre

KSIS
Utility for Image Building and Deployment

KVM
Keyboard Video Mouse (allows the keyboard, video monitor and mouse to be connected to the node)

L

LAN
Local Area Network

LAPACK
Linear Algebra PACKage

LDAP
Lightweight Directory Access Protocol

LDIF
LDAP Data Interchange Format:
A plain text data interchange format to represent LDAP directory contents and update requests. LDIF conveys directory content as a set of records, one record for each object (or entry). It represents update requests, such as Add, Modify, Delete, and Rename, as a set of records, one record for each update request.

LKCD
Linux Kernel Crash Dump:
A tool used to capture and analyze crash dumps.

LOV
Logical Object Volume

LSF
Load Sharing Facility

LUN
Logical Unit Number

LVM
Logical Volume Manager

LVS
Linux Virtual Server

M

MAC
Media Access Control (a unique identifier address attached to most forms of networking equipment).

MAD
Management Datagram

Managed Switch
A switch with no management interface and/or configuration options.

MDS
MetaData Server

MDT
MetaData Target

MFT
Mellanox Firmware Tools

MIB
Management Information Base

MKL
Maths Kernel Library

MPD
MPI Process Daemons

MPFR
C library for multiple-precision, floating-point computations

MPI
Message Passing Interface

MTBF
Mean Time Between Failures

MTU
Maximum Transmission Unit

N

Nagios
A tool used to monitor the services and resources of Bull HPC clusters.

NETCDF
Network Common Data Form

NFS
Network File System

NIC
Network Interface Card

NIS
Network Information Service

NS
NovaScale

NTP
Network Time Protocol

NUMA
Non Uniform Memory Access

NVRAM
Non Volatile Random Access Memory

O

OFA
Open Fabrics Alliance

OFED
Open Fabrics Enterprise Distribution

OPMA
Open Platform Management Architecture

OpenSM
Open Subnet Manager

OpenIB
Open InfiniBand

OpenSSH
Open Source implementation of the SSH protocol

OSC
Object Storage Client

OSS
Object Storage Server

OST
Object Storage Target

P

PAM
Platform Administration and Maintenance Software

PAPI
Performance Application Programming Interface

PBLAS
Parallel Basic Linear Algebra Subprograms

PBS
Portable Batch System

PCI
Peripheral Component Interconnect (Intel)

PDSH
Parallel Distributed Shell

PDU
Power Distribution Unit

PETSc
Portable, Extensible Toolkit for Scientific Computation

PGAPACK
Parallel Genetic Algorithm Package

PM
Performance Manager

Platform Management

PMI
Process Management Interface

PMU
Performance Monitoring Unit

pNETCDF
Parallel NetCDF (Network Common Data Form)

PVFS
Parallel Virtual File System

Q

QDR
Quad Data Rate

QoS
Quality of Service:
A set of rules which guarantee a defined level of quality in terms of transmission rates, error rates, and other characteristics for a network.

R

RAID
Redundant Array of Independent Disks

RDMA
Remote Direct Memory Access

ROM
Read Only Memory

RPC
Remote Procedure Call

RPM
RPM Package Manager

RSA
Rivest, Shamir and Adleman, the developers of the RSA public key cryptosystem

S

SA
Subnet Agent

SAFTE
SCSI Accessible Fault Tolerant Enclosures

SAN
Storage Area Network

SCALAPACK
SCALable Linear Algebra PACKage

SCSI
Small Computer System Interface

SCIPORT
Portable implementation of CRAY SCILIB

SDP
Socket Direct Protocol

SDPOIB
Sockets Direct Protocol over Infiniband

SDR
Sensor Data Record

Single Data Rate

SFP
Small Form-factor Pluggable transceiver - extractable optical or electrical transmitter/receiver module.

SEL
System Event Log

SIOH
Server Input/Output Hub

SIS
System Installation Suite

SL
Service Level

SL2VL
Service Level to Virtual Lane

SLURM
Simple Linux Utility for Resource Management – an open source, highly scalable cluster management and job scheduling system.

SM
Subnet Manager

SMP
Symmetric Multi Processing:
The processing of programs by multiple processors that share a common operating system and memory.

SNMP
Simple Network Management Protocol

SOL
Serial Over LAN

SPOF
Single Point of Failure

SSH
Secure Shell

Syslog-ng
System Log New Generation

T

TCL
Tool Command Language

TCP
Transmission Control Protocol

TFTP
Trivial File Transfer Protocol

TGT
Ticket-Granting Ticket

U

UDP
User Datagram Protocol

UID
User ID

ULP
Upper Layer Protocol

USB
Universal Serial Bus

UTC
Coordinated Universal Time

V

VCRC
Variant Cyclic Redundancy Check

VDM
Voltaire Device Manager

VFM
Voltaire Fabric Manager

VGA
Video Graphic Adapter

VL
Virtual Lane

VLAN
Virtual Local Area Network

VNC
Virtual Network Computing:
Used to enable access to Windows systems and Windows applications from the Bull NovaScale cluster management system.

W

WWPN
World–Wide Port Name

X

XFS
eXtended File System

XHPC
Xeon High Performance Computing

XIB
Xeon InfiniBand

XRC
Extended Reliable Connection:
Included in Mellanox ConnectX HCAs for memory scalability

Index

H

I

K

L

M

N

O

P

R

S

T

U

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A2 24FA 03