HPC

BAS4 V5.1 Fix12

New Features Guide

REFERENCE
86 A2 44FD 00

HPC

BAS4 V5.1 Fix12

New Features Guide

Software

November 2009

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

REFERENCE
86 A2 44FD 00

Trademarks and Acknowledgements

The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Preface

Scope and Objectives

BAS4 V5.1 Fix12 includes new versions for some of the **BAS4 V5.1 Fix 11** software components. The features of these software component versions are described in this manual so that the System Administrator can easily pinpoint the impact of the Fix12 for their cluster.

Note BAS4 V5.1 Fix12 also includes some bug fixes which are not described in this manual.

Intended Readers

This manual is for System Administrators and Users of systems of **BAS4 V5.1 Fix12** clusters.

Prerequisites

Refer to the *Bull HPC BAS4 V5.1 Fix12 Software Release Bulletin* (SRB) (Ref. 86 A2 52EJ) for details of any restrictions which apply to your release.

Important

The Software Release Bulletin contains the latest information regarding BAS4V5.1 Fix12. This should be read first. Contact Bull Technical Support for more information.

Bibliography

- *BAS4 V5.1 FIX12 – SRB (Software Release Bulletin)* - 86 A2 52EJ
- The *BAS4 V5.1 Documentation* CD-ROM (86 A2 97ER 09), delivered with FIX11, includes the following manuals:
  - *Bull HPC BAS4 Installation and Configuration Guide* (86 A2 28ER 11)
  - *Bull HPC BAS4 Administrator's Guide* (86 A2 30ER 12)
  - *Bull HPC BAS4 User's Guide* (86 A2 29ER 09)
  - *Bull HPC BAS4 Application Tuning Guide* (86 A2 19ER 06)
  - *Bull HPC BAS4 Maintenance Guide* (86 A2 46ER 06)
- The *BAS4 V5.1 FIX12 Documentation* CD-ROM (86 A2 47FD 00) includes the following manuals:
  - *BAS4 V5.1 FIX11 to FIX12 Upgrade Procedure* - 86 A2 43FD 00
  - *BAS4 V5.1 FIX12 New Features Guide* - 86 A2 44FD 00
  - *Lustre Guide* - 86 A2 46FD
  - *InfiniBand Guide* - 86 A2 42FD

Note The Bull Support Web site may be consulted for product information, documentation, downloads, updates and service offers:
http://support.bull.com

Table of Contents

List of Figures

List of Tables

Chapter 1. Fix12 - New Features

BAS4 V5.1 Fix12 includes new versions of the following products:

- NSCTRL / NSFIRM commands
  Refer to Chapter 2
- MPIBull2
  Refer to Chapter 4
- BSM HPC Edition
  Refer to Chapter 3
- Scientific Studio
  Refer to Chapter 5
- Profiling Programs - HPC Toolkit
  Refer to Chapter 6
- Profiling Tools - profilecomm
  Refer to Chapter 7
- Compilers
  Refer to Chapter 8
- Lustre File System
  Refer to the *Lustre Guide* (86 A2 46FD), delivered on the *BAS4 V5.1 FIX12 Documentation* CD-ROM (86 A2 47FD)
- InfiniBand
  Refer to the *InfiniBand Guide* (86 A2 42FD), delivered on the *BAS4 V5.1 FIX12 Documentation* CD-ROM (86 A2 47FD)

Note Bug fixes and non-visible changes are not described in thist guide.

Chapter 2. nsctrl and nsfirm Commands

BAS4 V5.1 Fix 12 includes changes to the **nsctrl** command, and introduces the **nsfirm** command:

- *The nsctrl command* is now dedicated to the tasks related to hardware control alone.
- *The nsfirm command* carries out the tasks related to the node firmware for the **BIOS**, **FPGA**, **PAM** and **BMC**.

Both the **nsctrl** and **nsfirm** commands run from the Management Node.

2.1 The nsctrl command

The **nsctrl** command carries out various tasks related to hardware control. This command must be run from the Management Node. These tasks may be performed on any node (Compute Node, I/O Node, etc.), except for the Management Node.

Note

nsctrl provides the Administrator with a simple interface for running **Bull System Manager** commands. Depending on the action specified by the command, **nsctrl** will run one or more **BSM** commands, using the Cluster Database to obtain the information required (for example, PAP, PAM login, PAM Password, PAM IP addresses, etc.).

Usage:

/usr/sbin/nsctrl [options] <action> [<nodes>]

General Options:

--debug Debug mode (more than verbose).

--dbname name Specify database name.

--force, -f Do not ask for confirmation or state check.

--group, -g Specify a group of nodes. You can use the **dbmGroup show** command to display the defined groups.

--help, -h Display **nsctrl** help.

--time, -t Time to wait between each **interval**.

--interval, -i Specify the number of calls made before the waiting time (**--time** option).

--only_test, -o Display the BSM commands that would be launched, according to the options and actions specified. This is used for tests, no actions are performed.

--pap, -p Specify one or more pap(s): base1[i,j-k]base2[i,j-k] **.**

--verbose, -v Verbose mode.

Specifying nodes:

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

Actions for all types of nodes

dump
ping
poweron
poweroff
poweroff_force
reset
status
temperature

Actions specific to NovaScale 5XXX/6XXX

domainhardware
domaininfo
hardwareinfo (requires **--element <hardware component>**)
multithreading=yes/no
identitycard
PMB_temperature
FRU_temperatures (requires **--element <hardware component>**)
temperature
consumption
clear_fault_list
set_network_name (requires **--nwname <name>**)
haltonmcreset=yes/no
set_l3cachesize (requires **--size <l3cache size>**)
exclude_cpu=yes/no
get_clock_frequency
set_clock_frequency (requires **--Value <value-in-Khz>**))
infoNS56 Show actions specific to the NovaScale 5XXX/6XXX Series

Notes

- The **--element** parameter specifies one of the following hardware types QBB, CPU, MEM, IOB or IOR. The hardware type is retrieved using the **domainhardware** action.
- The **--Value** parameter specifies the clock frequency setting.

Relation between nsctrl actions and Commands

The following table shows which **nsctrl** actions are performed by the **BSM** commands.

	BSM Command			
	bsmreset	bsmpower	bsminfo	bsmpamcmd
nsctrl action	reset	poweron poweroff poweroff_force status	temperature	dump
nsctrl action specific to NovaScale 5XXX/6XXX Series			domainhardware domaininfo hardwareinfo identitycard FRU_temperatures	clear_fault_list consumption exclude_cpu=yes/no get_clock_frequency haltonmcreset multithreading=yes/no PMB_temperature reset set_clock_frequency set_l3cachesize set_network_name

Table 2-1. Relation between nsctrl action and BSM Commands

Examples:

Note In the following examples the **–o** option (**--only_test**) is used to display which BSM commands would be launched for the specified action.

Examples of commands available for all types of nodes

- To power off node `tiger1`, enter:

```
# nsctrl -o poweroff_force tiger1
```

```
tiger1 : /usr/BSMHW/bin/bsmpower.sh -a off_force -m tiger -H tiger1 -u user2
```

- To ping node `tiger1`, enter:

```
# nsctrl -o ping tiger1
```

```
tiger1 : ping -c 1  tiger1
```

Examples of commands specific to NovaScale 5XXX/6XXX Series

- To get information about `QBB 0` of `MODULE 0` on node `nova9`, enter:

```
# nsctrl -o hardwareinfo -e MODULE_0/QBB_0 nova9
```

```
nova9 : /usr/BSMHW/bin/bsminfo.sh -i hardwareinfo -e PAM:/CELLSBLOCK_1XAN-S11-
00024/MODULE_0/QBB_0 -m fame -D 1XAN-S11-00024 -M papu0c2 -u administrator
```

- To get information about the temperatures of the hardware components of `QBB 0` of `MODULE 0` on node `nova9`, enter:

```
# nsctrl -o FRU_temperatures -e MODULE_0/QBB_0 nova9
```

```
nova9 : /usr/BSMHW/bin/bsminfo.sh -i FRUtemperatures -e PAM:/CELLSBLOCK_1XAN-S11-
00024/MODULE_0/QBB_0 -m fame -D 1XAN-S11-00024 -M papu0c2 -u administrator
```

- To exclude the CPUs on node `nova9`, enter:

```
# nsctrl -o exclude_cpu=yes nova9
```

```
Pass n°1 /usr/sbin/nsctrl set_exclusion_state=yes nova9 --numqbb 0 --force --only_test
Pass n°2 /usr/sbin/nsctrl set_exclusion_state=yes nova9 --numqbb 1 --force --only_test
Pass n°3 /usr/sbin/nsctrl set_exclusion_state=yes nova9 --numqbb 2 --force --only_test
Pass n°4 /usr/sbin/nsctrl set_exclusion_state=yes nova9 --numqbb 3 --force --only_test
Pass n°5 /usr/sbin/nsctrl poweroff_force nova9 --force --only_test
Pass n°1: nova9 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_exclusion_state -E yes -e
PAM:/CELLSBLOCK_1XAN-S11-00024/MODULE_0/QBB_0/CPU_2 -m pam -D 1XAN-S11-00024 -M papu0c2
-u administrator
Pass n°2: nova9 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_exclusion_state -E yes -e
PAM:/CELLSBLOCK_1XAN-S11-00024/MODULE_0/QBB_1/CPU_2 -m pam -D 1XAN-S11-00024 -M papu0c2
-u administrator
Pass n°3: nova9 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_exclusion_state -E yes -e
PAM:/CELLSBLOCK_1XAN-S11-00024/MODULE_0/QBB_2/CPU_2 -m pam -D 1XAN-S11-00024 -M papu0c2
-u administrator
Pass n°4: nova9 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_exclusion_state -E yes -e
PAM:/CELLSBLOCK_1XAN-S11-00024/MODULE_0/QBB_3/CPU_2 -m pam -D 1XAN-S11-00024 -M papu0c2
-u administrator
Pass n°5: nova9 : /usr/BSMHW/bin/bsmpower.sh -a off_force -m fame -D 1XAN-S11-00024 -M
papu0c2 -u administrator
```

- To generate a dump of node `ns6`, enter:

```
nsctrl -o dump ns6
```

```
Confirm your request: dump on ns6 (y/n)?
y
ns6 : /usr/BSMHW/bin/bsmpower.sh -a diag -m ipmilan -H hwm6 -u administrator
```

- To set the clock frequency to the value 60 KHz on node `nova6`, enter:

```
# nsctrl -o set_clock_frequency nova6 --Value 60
```

```
nova6 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_clock_frequency -C 60 -m
pam -D 2XAN-S11-00026 -M papu0c1 -u administrator
```

2.2 The nsfirm command

The **nsfirm** command carries out various tasks related to firmware. This command must be run from the Management Node. These tasks may be performed on any node (Compute Node, I/O Node, etc.), except for the Management Node.

Note **nsfirm** implements BSM commands in the same manner as nsctrl. Depending on the action specified in the command, **nsfirm** will run one or more BSM commands, using the ClusterDB to provide the BSM command with the required information (for example, PAP, PAM login, PAM Password, PAM IP addresses, etc.).

Usage:

/usr/sbin/nsfirm [options] <action> [<nodes>]

General Options:

--help, -h	Display nsfirm help.
--dbname name	Specify database name.
--pap, -p	Specify one or more pap(s): base1[i,j-k]base2[i,j-k].
--group, -g	Specify a group of nodes.
--force, -f	Do not ask for confirmation or state check.
--verbose, -v	Verbose mode.
--debug	Debug mode (more than verbose).
--only_test, -o	Testing mode, no action performed.

Specifying nodes:

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

Actions specific to NovaScale 5XXX/6XXX

get_bios_modes
set_bios_modes (requires --Index and --Value)
upgrade_pam (same as **send_pam + set_pam**) (requires –File, --Version and --pap)
send_pam (requires --File and --pap)
set_pam (requires --Version and --pap)
upgrade_bios (same as **send_bios + set_bios**) (requires --File and --Version)
send_bios (requires --File and --pap)
set_bios (requires --Version and --pap)
upgrade_fpga (same as **send_fpga + set_fpga**) (requires –File, --Version and --pap)
send_fpga (requires --File and --pap)
set_fpga (requires --Version)
load_bios (same as **set_recovery + reset**)
set_recovery
reset

Actions specific to HPC T10 Storage System

upgrade_bios (same as **push_package** + **exec_package**) (requires --File)
push_package (requires --File)
exec_package (requires --File)
exec_reboot
biosversion

Actions specific to NovaScale 30X5 Series

upgrade_firmware (same as **push_fwpackage** + **set_nextbootEFI** + **fw_activate**)
(requires --Directory)
push_fwpackage (requires --Directory)
set_nextbootEFI
fw_activate
bios_version
bmc_version

Notes

- The **--Version** parameter sets the version of the PAM or BIOS or FPGA to activate on the PAP.
- The **--File** parameter specifies the file to be used for the BIOS, PAM, FPGA or BMC uprade.
- The **--Value** parameter specifies a value for a entry in a Index for the BIOS (--Index)
- The **--Index** parameter specifies a BIOS index [0,7].
- The **--Directory** parameter specifies the full path where to find the firmware files to be transferred for **EFI** (30x5 series only).
- The **--FW** parameter specifies which firmware in [BMC, SAL] to update by EFI. Default is both. (30x5 series only).

Examples_of commands specific to the NovaScale 5XXX/6XXX Series

- To set the BIOS mode index 0 to the value 0 on node nova6, enter:

```
# nsfirm –o set_bios_modes nova6 --Index 0 --Value 0
```

```
nova6 : /usr/BSMHW/bin/bsmpamcmd.sh -a set_bios_modes -X 0 -C 0
-m pam -D 2XAN-S11-00026 -M papu0c1 -u administrator
```

- To upgrade the BIOS with the **/root/newbios** file on PAP papu0c0, enter:

```
# nsfirm -o upgrade_bios --File /root/newbios --Version 1 -p papu0c0
```

```
Pass n°1 /usr/sbin/nsfirm send_bios --File /root/newbios --pap papu0c0 --only_test
Pass n°2 /usr/sbin/nsfirm set_bios --Version 1 --pap papu0c0 --only_test
Pass n°1: Confirm your request: send_bios on papu0c0 (y/n)?
y
Pass n°1: papu0c0 : /opt/BSMHW/bin/bsmpamcmd.sh -a transfer_file -T BIOS -S 192.168.32.22
-F /root/newbios -L linux -W linux -m pam -M papu0c0 -u administrator
Pass n°2: Confirm your request: set_bios on papu0c0 (y/n)?
y
Pass n°2: papu0c0 : /opt/BSMHW/bin/bsmpamcmd.sh -a set_ref_bios -v 1 -m pam -M papu0c0 -u
administrator
```

Examples of commands specific to the NovaScale 3005 Series

- To display the **BIOS** version of node ns6, enter:

```
# nsfirm –o bios_version ns6
```

```
ns6 : /usr/BSMHW/bin/bsmFWcmd.sh -a get_bios_version -m NS3045 - M ns6
```

- To display the **BMC** version of node ns6, enter:

```
# nsfirm -o bmc_version ns6
```

```
ns6 : /usr/BSMHW/bin/bsminfo.sh -i BMCinfo -m ipmilan -H hwm6 -u administrator
```

- To upgrade the BMC firmware of node ns10, with the files contained in the **/tmp** directory, enter:

Note The **/tmp** directory must contain two files (**.rom** file and **.efi** file); for example: `fw_03-21_03-22.rom` and `fwupdate_Rev1_11.efi`

```
# nsfirm –o upgrade_firmware --Directory /tmp ns10 --FW BMC
```

```
Pass n°1 /usr/sbin/nsfirm push_fwpackage --Directory /tmp --FW BMC ns10 --only_test
Pass n°2 /usr/sbin/nsfirm set_nextbootEFI ns10 --only_test
Pass n°3 /usr/sbin/nsfirm fw_activate ns10 --only_test
Pass n°1: Confirm your request: push_fwpackage on ns10 (y/n)?
y
Pass n°1: ns10 : /usr/BSMHW/bin/bsmFWcmd.sh -a push_package -D /tmp -U B -m NS3045c -M ns10
Pass n°2: Confirm your request: set_nextbootEFI on ns10 (y/n)?
y
Pass n°2: ns10 : /usr/BSMHW/bin/bsmFWcmd.sh -a set_nextbootEFI -m NS3045c -M ns10
Pass n°3: Confirm your request: fw_activate on ns10 (y/n)?
y
Pass n°3: ns10 : /usr/BSMHW/bin/bsmFWcmd.sh -a exec_reboot -m NS3045c -M ns10
```

Relation between nsfirm actions and BSM commands

The following table shows which BSM command is performed for each **nsfirm** action.

	BSM Command				
	bsmpamcmd	bsmFWcmd	bsmbioscmd	bsmreset	bsminfo
Actions specific to NovaScale 5XXX/6XXX Series	get_bios_modes load_bios send_bios set_bios_modes send_pam set_bios set_pam set_recovery upgrade_bios upgrade_pam upgrade_fpga send_fpga set_fpga			reset	
Actions specific to HPC T10 Storage Systems		biosversion	upgrade_bios push_package exec_package exec_reboot		
Actions specific to NovaScale 3005 Series		biosversion fw_activate push_fwpackage set_nextbootEFI upgrade_firmware			bmc_version

Table 2-2. Relation between nsfirm actions and BSM commands

Chapter 3. Monitoring with Bull System Manager - HPC Edition

Bull System Manager - HPC Edition provides the monitoring functions for Bull Extreme Computing systems. It uses **Nagios** and **Ganglia** open source software. **Nagios** is used to monitor the operating status for the different components of the cluster. **Ganglia** collects performance statistics for each cluster node and displays them graphically for the whole cluster. The status of a large number of elements can be monitored.

This chapter covers the following topics:

- 3.1 *Launching Bull System Manager - HPC* Edition
- 3.2 *Access Rights*
- 3.3 *Hosts, Services and Contacts for Nagios*
- 3.4 *Using Bull System Manager - HPC* Edition
- 3.5 *Map Button*
- 3.6 *Status Button*
- 3.7 *Log Window*
- 3.8 *Alerts Button*
- 3.9 *Storage Overview*
- 3.10 *Shell*
- 3.11 *Monitoring the Performance - Ganglia Statistics*
- 3.12 *Group Performance View*
- 3.13 *Global Performance View*
- 3.14 *Configuring and Modifying Nagios Services*
- 3.15 *General Nagios Services*
- 3.16 *Management Node Nagios Services*
- 3.17 *Ethernet Switch Services*
- 3.18 *Portserver Services*

3.1 Launching Bull System Manager - HPC Edition

Note The cluster database (**ClusterDB**) must be running before monitoring is started.

1. If necessary restart the **gmond** and **gmetad** services:

```
service gmond restart
service gmetad restart
```

2. Start the monitoring service:

```
service nagios start
```

3. Start **Mozilla** and enter the following URL:

   http://<ManagementNode>/BSM/

Note **Mozilla** is the mandatory navigator for **Bull System Manager – HPC Edition**

3.2 Access Rights

3.2.1 Administrator Access Rights

By default, the Administrator uses the following login and password:

login:	**nagios**
password:	**nagios**

Once the graphical interface for monitoring has opened, see *Figure 3-1*, the Administrator is able to enter host and service commands, whereas an ordinary user will only be able to consult the interface.

3.2.2 Standard User Access Rights

By default, an ordinary user uses the following login and password:

login:	**guest**
password:	**guest**

3.2.3 Adding Users and Changing Passwords

The **htpasswd** command is used to create new user names and passwords.

Create additional users for the graphical interface as follows:

1. Enter the following command:

```
htpasswd /opt/BSMServer-Base/core/etc/htpasswd.users <login>
```

   This command will prompt you for a password for each new user, and will then ask you to confirm the password.

2. You must also define the user profile in the **/opt/BSMServer-Base/core/share/console/NSMasterConfigInfo.inc** file (either as an Administrator or as an Operator).

Change the password for an existing user as follows:

1. Enter the following command:

```
htpasswd /opt/BSMServer-Base/core/etc/htpasswd.users <login>
```

2. Enter and confirm the new password when prompted.

Note Some of these steps have to be done as the **root** user.

See The **Bull System Manager** documentation for more information on adding users and on account management.

3.3 Hosts, Services and Contacts for Nagios

Nagios defines two entities: **hosts** and **services.**

A **host** is any physical server, workstation, device etc. that resides on a network.

The **host group** definition is used to group one or more hosts together for display purposes in the graphical interface.

The **service** definition is used to identify a *service* that runs on a host. The term *service* is used very loosely. It can mean an actual service that runs on the host (**POP**, **SMTP**, **HTTP**, etc.) or some other type of metric associated with the host (response to a ping, number of users logged-in, free disk space, etc.).

Note **Bull System Manager – HPC Edition** will display the services specific to each host when the host is selected within the interface.

The **contact** definition is used to identify someone who should be contacted in the event of a problem on your network.

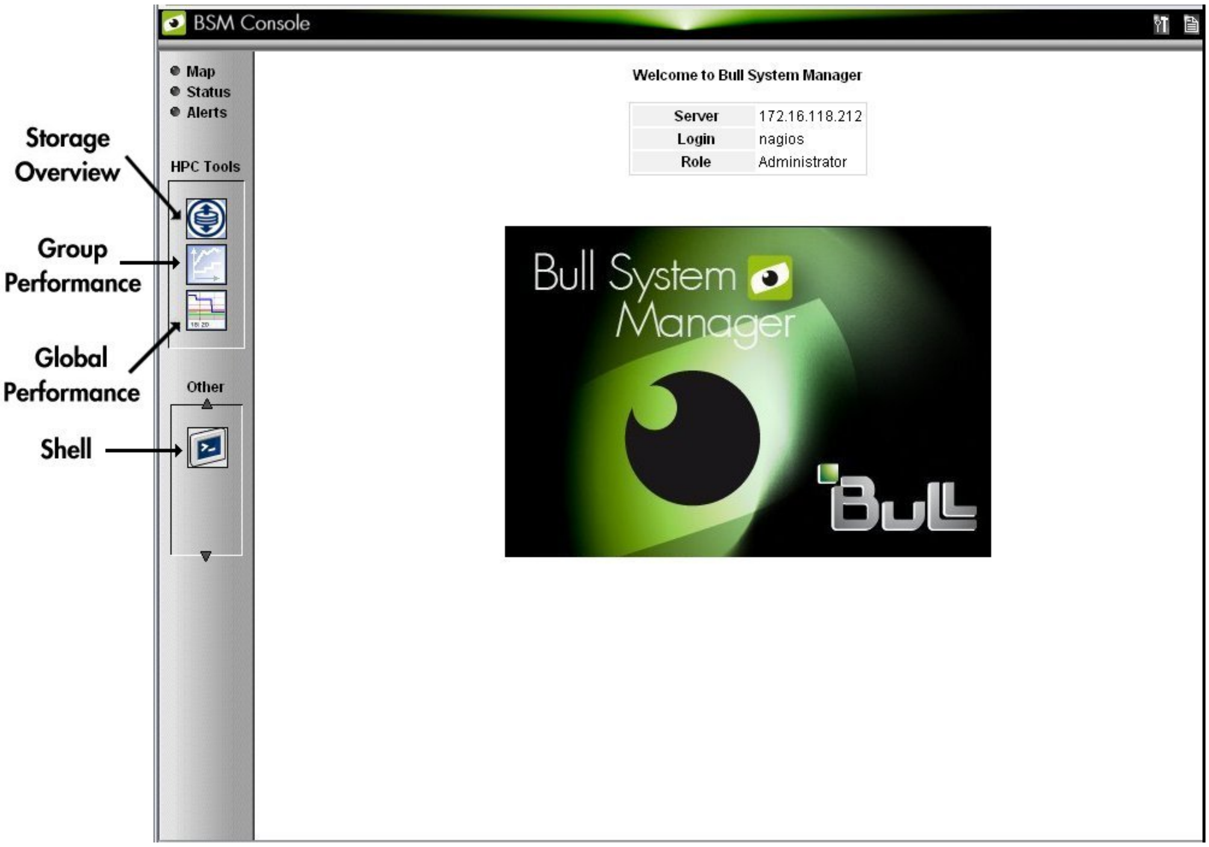
The **contact group** definition is used to group one or more contacts together for the purpose of sending out alert/recovery notifications. When a **host** or **service** has a problem or recovers, Nagios will find the appropriate contact groups to send notifications to, and notify all contacts in these contact groups. This allows greater flexibility in determining who gets notified for particular events.

For more information on the definitions, and the arguments and directives which may be used for the definitions see:
http://nagios.sourceforge.net/docs/3_0/

Alternatively, select the **Documentation** link from the **Bull System Manager** opening screen or select the **Documentation** button in the title bar.

3.4 Using Bull System Manager - HPC Edition

The graphical interface of **Bull System Manager - HPC Edition** is shown inside a Web browser.



Welcome to Bull System Manager

Server	172.16.118.212
Login	nagios
Role	Administrator

Figure 3-1. Bull System Manager - HPC Edition opening view

3.4.1 Bull System Manager - HPC Edition – View Levels

Initially, the console will open and the administrator can choose to view different types of monitoring information, with a range of granularity levels, by clicking on the icons in the left hand vertical tool bar, and then clicking on the links in the different windows displayed. The information displayed is contextual depending on the host or service selected. Using the links it is possible to descend to a deeper level, to see more detailed information for a particular host, host group, or service. For example, the **Cabinet Rack** map view in *Figure 3-2* leads to the **Rack View** in *Figure 3-3*, which in turns leads to the more detailed **Services** view in *Figure 3-4*, for the host selected in the **Rack View**.

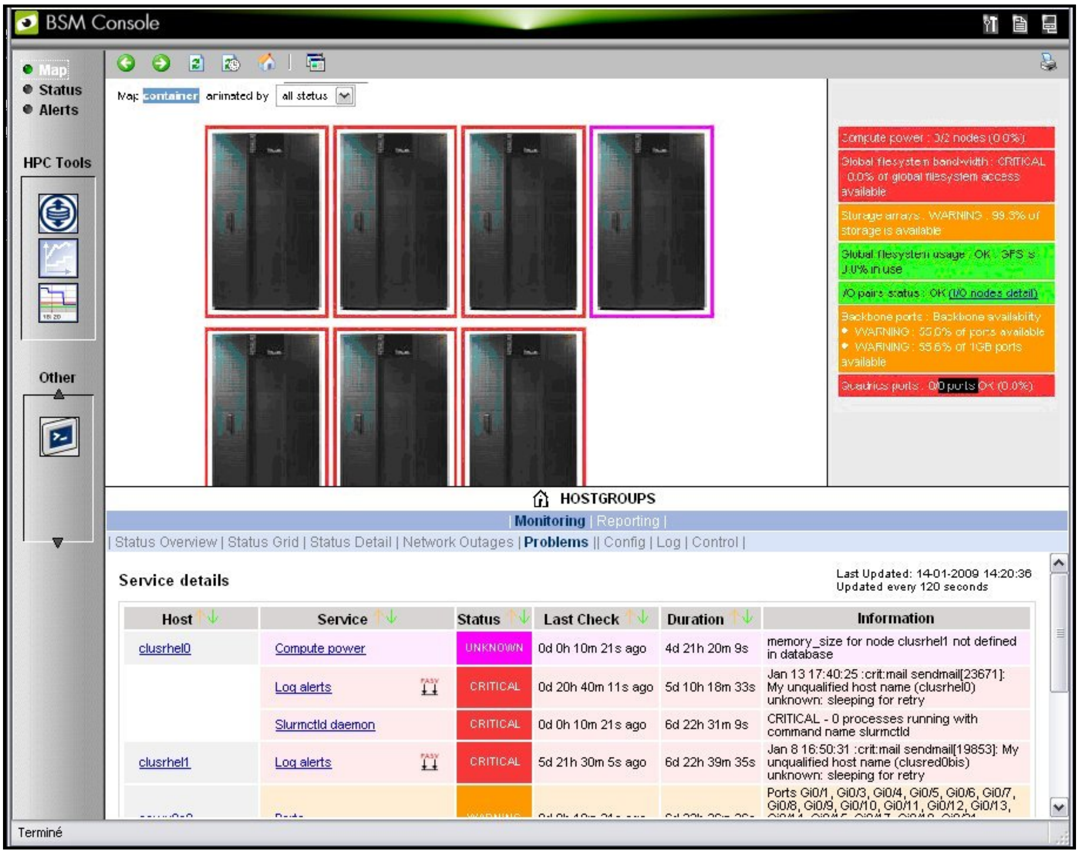
3.5 Map Button

The **Map** button is displayed at the top right hand side of the opening. When this is selected the drop down menu provides two view options, **all status** or **ping,** inside the main window.

3.5.1 All Status Map View

The **all status** map view presents a chart of the cluster representing the various server rack cabinets in the room. The frame color for each cabinet is determined by the component within it with the highest alarm status, for example if an **Ethernet interface** is in the **critical** status than the status for the whole rack will be **critical**.

By default, in addition to the view of the rack cabinets in the room, the **Monitoring - Problems** window will appear at the bottom of the screen with a status for all the **hosts** and **services** and the **Availability Indicators** view window will appear at the top right hand side of the screen – see *Figure 3-2*.


BSM Console
Map
Status
Alerts
HPC Tools
Other
HOSTGROUPS
| Monitoring | Reporting |
| Status Overview | Status Grid | Status Detail | Network Outages | Problems || Config | Log | Control |
Terminé

Map container animated by all status

Compute power : 0/2 nodes (0.0%)

Global filesystem bandwidth : CRITICAL : 0.0% of global filesystem access available

Storage arrays : WARNING : 99.3% of storage is available

Global filesystem usage : OK : GFS is 0.0% in use

I/O pairs status : OK (I/O nodes detail)

Backbone ports : Backbone availability
- WARNING : 55.0% of ports available
- WARNING : 55.6% of 1GB ports available

Quadrics ports : 0/0 ports OK (0.0%)

Service details

Last Updated: 14-01-2009 14:20:36
Updated every 120 seconds

Host	Service	Status	Last Check	Duration	Information
clusrhel0	Compute power	UNKNOWN	0d 0h 10m 21s ago	4d 21h 20m 9s	memory_size for node clusrhel1 not defined in database
	Log alerts	CRITICAL	0d 20h 40m 11s ago	5d 10h 18m 33s	Jan 13 17:40:25 :crit:mail sendmail[23671]: My unqualified host name (clusrhel0) unknown: sleeping for retry
	Slurmctld daemon	CRITICAL	0d 0h 10m 21s ago	6d 22h 31m 9s	CRITICAL - 0 processes running with command name slurmctld
clusrhel1	Log alerts	CRITICAL	5d 21h 30m 5s ago	6d 22h 39m 35s	Jan 8 16:50:31 :crit:mail sendmail[19853]: My unqualified host name (clusred0bis) unknown: sleeping for retry
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Ports Gi0/1, Gi0/3, Gi0/4, Gi0/5, Gi0/6, Gi0/7, Gi0/8, Gi0/9, Gi0/10, Gi0/11, Gi0/12, Gi0/13,

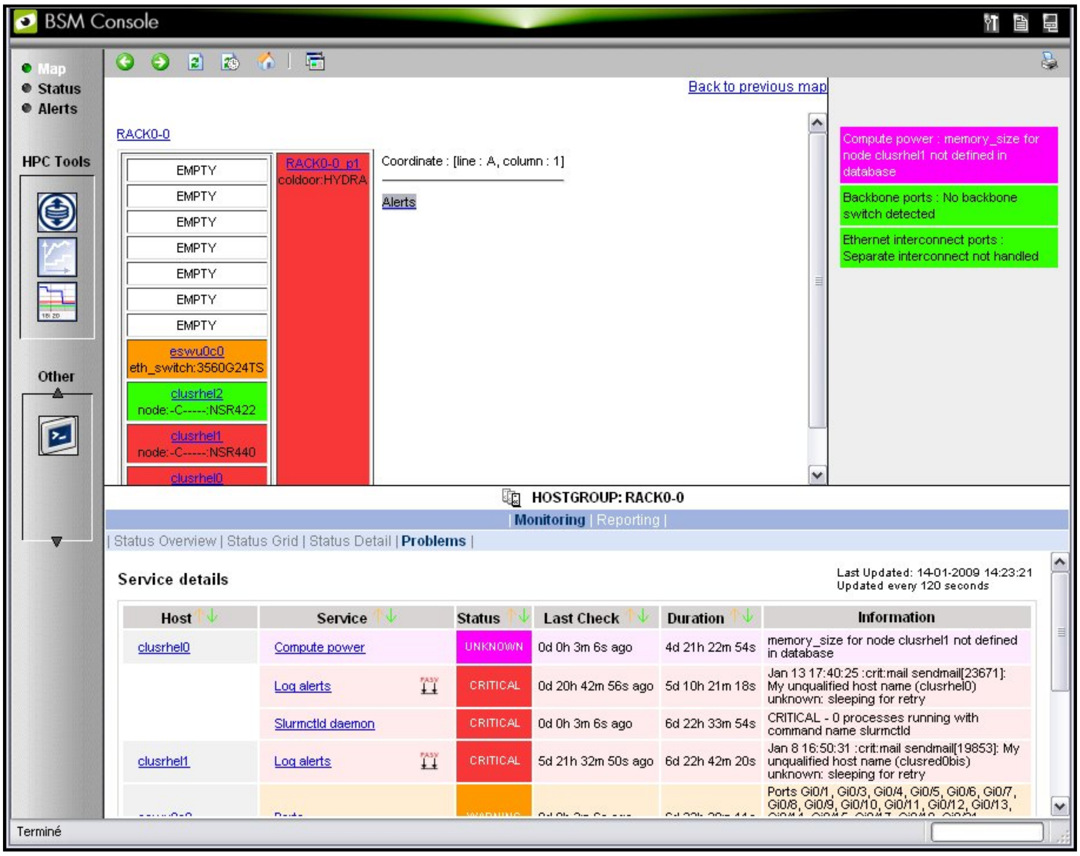
Figure 3-2. **Map** button **all status** opening view

When the cursor passes over a rack, information about it (label, type, and the elements contained in the rack) is displayed. When the user clicks on a cabinet, a detailed view of the cabinet is displayed – see **Rack view** in *Figure 3-3*. This displays additional information, including its physical position and the services which are in a non-OK state.

3.5.2 Rack View

The **Rack view** details the contents of the rack: the nodes, their position inside the rack, their state, with links to its **Alert** history, etc. The list of the problems for the rack is displayed at the bottom of the view – see *Figure 3-3*.

Clicking on a component displays a detailed view for it.



EMPTY
EMPTY
EMPTY
EMPTY
EMPTY
EMPTY
EMPTY
eswu0c0 eth_switch:3560G24TS
clusrhel2 node:-C-----:NSR422
clusrhel1 node:-C-----:NSR440
clusrhel0

RACK0-0_p1
coldoor:HYDRA

Coordinate : [line : A, column : 1]

Compute power : memory_size for node clusrhel1 not defined in database

Backbone ports : No backbone switch detected

Ethernet interconnect ports : Separate interconnect not handled

HOSTGROUP: RACK0-0

| Monitoring | Reporting |

| Status Overview | Status Grid | Status Detail | **Problems** |

Service details

Last Updated: 14-01-2009 14:23:21
Updated every 120 seconds

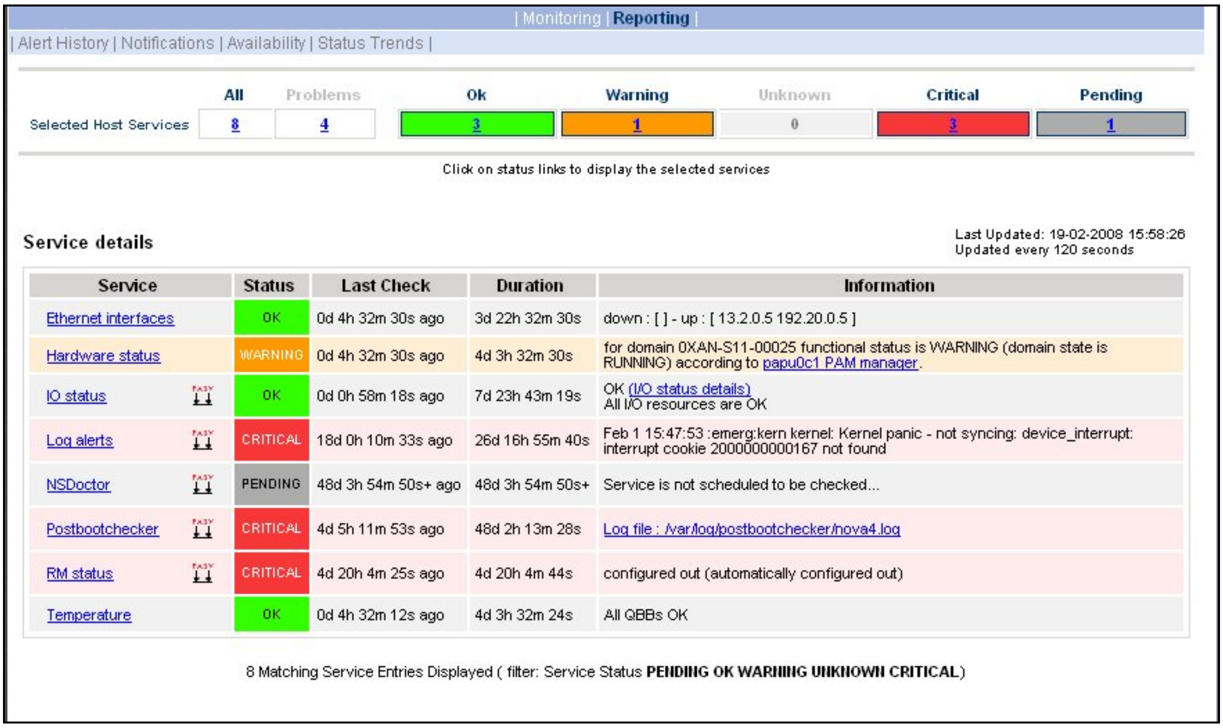
Host	Service	Status	Last Check	Duration	Information
clusrhel0	Compute power	UNKNOWN	0d 0h 3m 6s ago	4d 21h 22m 54s	memory_size for node clusrhel1 not defined in database
	Log alerts	CRITICAL	0d 20h 42m 56s ago	5d 10h 21m 18s	Jan 13 17:40:25 :crit:mail sendmail[23671]: My unqualified host name (clusrhel0) unknown: sleeping for retry
	Slurmctld daemon	CRITICAL	0d 0h 3m 6s ago	6d 22h 33m 54s	CRITICAL - 0 processes running with command name slurmctld
clusrhel1	Log alerts	CRITICAL	5d 21h 32m 50s ago	6d 22h 42m 20s	Jan 8 16:50:31 :crit:mail sendmail[19853]: My unqualified host name (clusred0bis) unknown: sleeping for retry
					Ports Gi0/1, Gi0/3, Gi0/4, Gi0/5, Gi0/6, Gi0/7, Gi0/8, Gi0/9, Gi0/10, Gi0/11, Gi0/12, Gi0/13,

Figure 3-3. **Rack view** with the **Problems** window at the bottom

More detailed information regarding the hardware components and services associated with a host appear, when the host in the rack view is clicked. This leads to another pop up window which includes further information for the host and its services – see *Figure 3-4*.

3.5.3 Host Services detailed View

Clicking the **Status** or a **Service** links in this window displays more specific information for the component or service.



| Monitoring | Reporting |

| Alert History | Notifications | Availability | Status Trends |

	All	Problems	Ok	Warning	Unknown	Critical	Pending
Selected Host Services	8	4	3	1	0	3	1

Click on status links to display the selected services

Service details

Last Updated: 19-02-2008 15:58:26
Updated every 120 seconds

Service	Status	Last Check	Duration	Information
Ethernet interfaces	OK	0d 4h 32m 30s ago	3d 22h 32m 30s	down : [ ] - up : [13.2.0.5 192.20.0.5]
Hardware status	WARNING	0d 4h 32m 30s ago	4d 3h 32m 30s	for domain 0XAN-S11-00025 functional status is WARNING (domain state is RUNNING) according to papu0c1 PAM manager.
IO status	OK	0d 0h 58m 18s ago	7d 23h 43m 19s	OK (I/O status details) All I/O resources are OK
Log alerts	CRITICAL	18d 0h 10m 33s ago	26d 16h 55m 40s	Feb 1 15:47:53 :emerg:kern kernel: Kernel panic - not syncing: device_interrupt: interrupt cookie 2000000000167 not found
NSDoctor	PENDING	48d 3h 54m 50s+ ago	48d 3h 54m 50s+	Service is not scheduled to be checked...
Postbootchecker	CRITICAL	4d 5h 11m 53s ago	48d 2h 13m 28s	Log file : /var/log/postbootchecker/nova4.log
RM status	CRITICAL	4d 20h 4m 25s ago	4d 20h 4m 44s	configured out (automatically configured out)
Temperature	OK	0d 4h 32m 12s ago	4d 3h 32m 24s	All QBBs OK

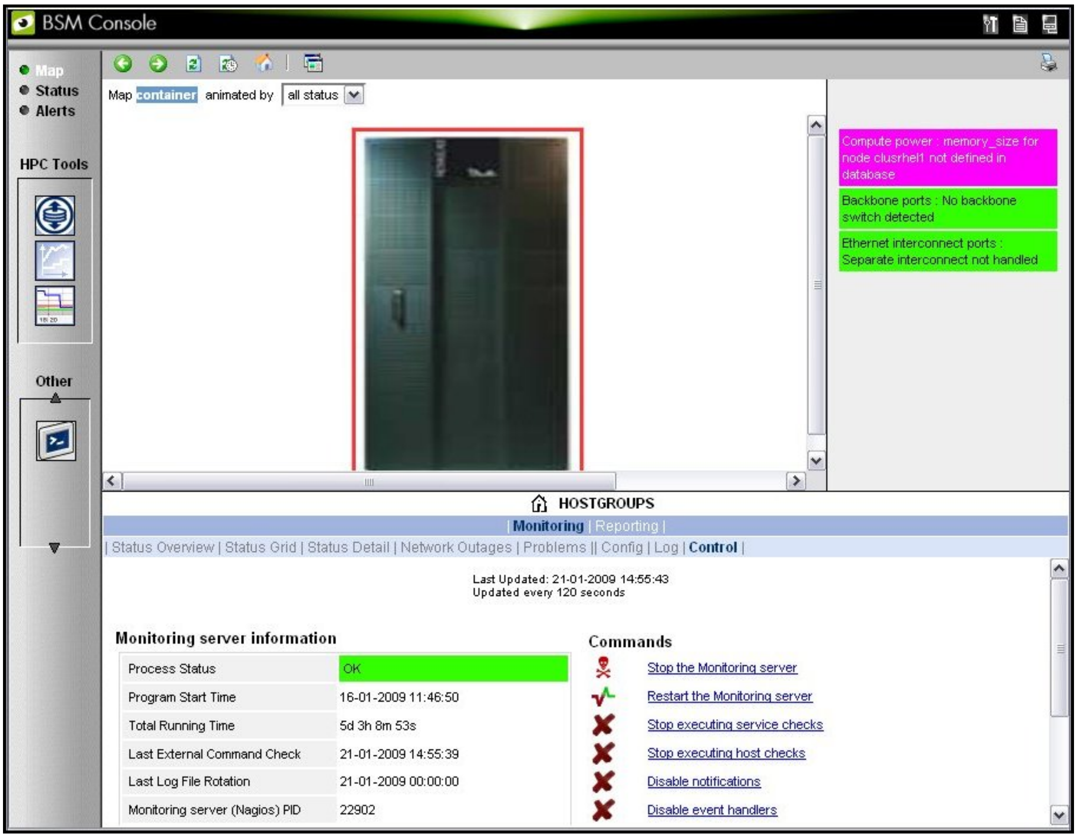
8 Matching Service Entries Displayed (filter: Service Status **PENDING OK WARNING UNKNOWN CRITICAL**)

Figure 3-4. Host Service details

By clicking on the links in the windows even more detailed information is provided for the services.

3.5.4 Control view

The **Control** button in the middle of screen provides details for the Management Node and the commands which apply to it - see *Figure 3-5*.


BSM Console
Map
Status
Alerts
HPC Tools
Other
Map container animated by all status
HOSTGROUPS
| Monitoring | Reporting |
| Status Overview | Status Grid | Status Detail | Network Outages | Problems || Config | Log | Control |

Compute power : memory_size for node clusrhel1 not defined in database

Backbone ports : No backbone switch detected

Ethernet interconnect ports : Separate interconnect not handled

Last Updated: 21-01-2009 14:55:43
Updated every 120 seconds

Monitoring server information

Process Status	OK
Program Start Time	16-01-2009 11:46:50
Total Running Time	5d 3h 8m 53s
Last External Command Check	21-01-2009 14:55:39
Last Log File Rotation	21-01-2009 00:00:00
Monitoring server (Nagios) PID	22902

Commands

- Stop the Monitoring server
- Restart the Monitoring server
- Stop executing service checks
- Stop executing host checks
- Disable notifications
- Disable event handlers

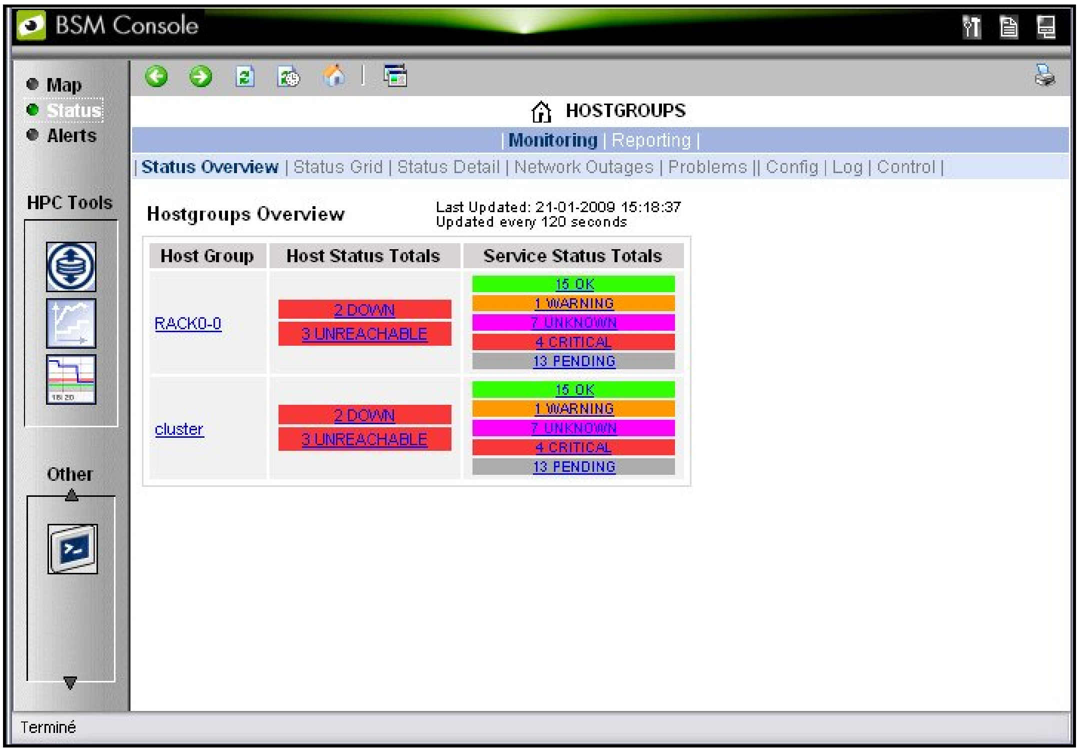
Figure 3-5. Monitoring Control Window

3.5.5 Ping Map View

The **ping** map view is similar to the **all status** map view, except that it only shows the state of the pings sent to the different components in the cabinets. The state of the services associated with the nodes is not taken into account.

By default the **Monitoring Problems** window will appear at the bottom of the screen.

3.6 Status Button


BSM Console
Map
Status
Alerts
HPC Tools
Other
HOSTGROUPS
| Monitoring | Reporting |
| Status Overview | Status Grid | Status Detail | Network Outages | Problems || Config | Log | Control |
Terminé

Hostgroups Overview

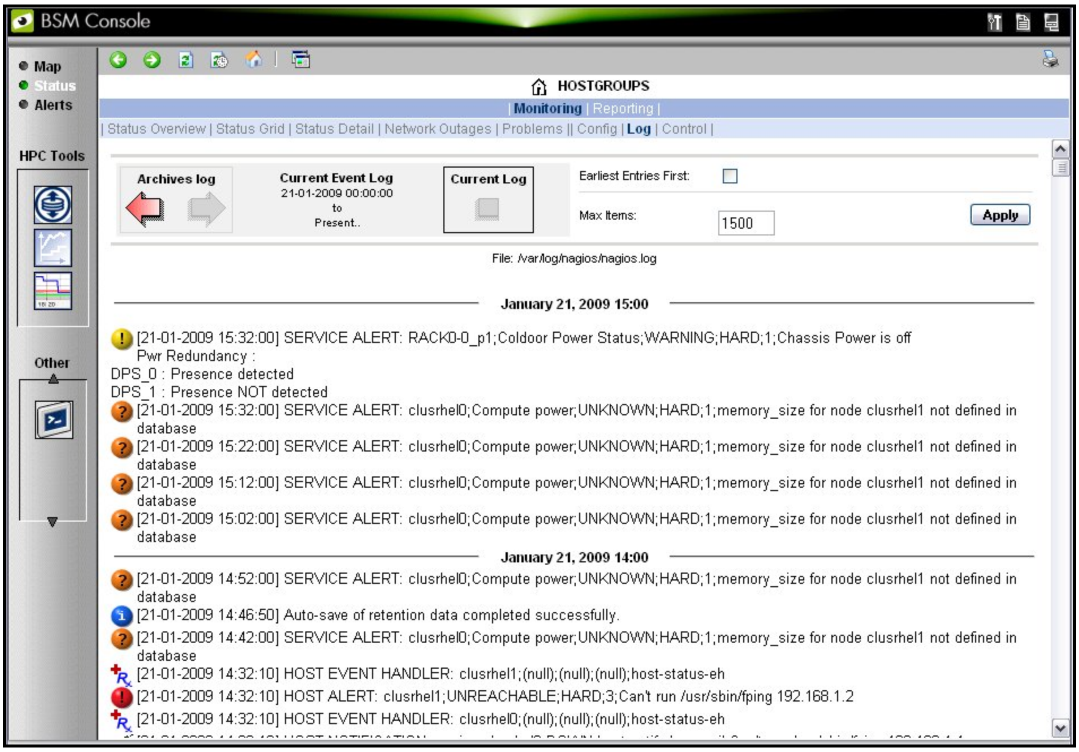
Last Updated: 21-01-2009 15:18:37
Updated every 120 seconds

Host Group	Host Status Totals	Service Status Totals
RACK0-0	2 DOWN 3 UNREACHABLE	15 OK 1 WARNING 7 UNKNOWN 4 CRITICAL 13 PENDING
cluster	2 DOWN 3 UNREACHABLE	15 OK 1 WARNING 7 UNKNOWN 4 CRITICAL 13 PENDING

Figure 3-6. **Status Overview** screen

When the **Status** button is clicked, a screen appears which lists all the hosts, and the status of the services running on them, as shown in *Figure 3-6*. More detailed information may be seen for each **Host Group** by selecting either the individual **Host Group**, or by selecting the links in the **Host Status Totals** or **Service Status Totals** columns.

3.7 Log Window


BSM Console
Map
Status
Alerts
HPC Tools
Other
HOSTGROUPS
| Monitoring | Reporting |
| Status Overview | Status Grid | Status Detail | Network Outages | Problems || Config | Log | Control |
Archives log
Current Log
Apply

Current Event Log
21-01-2009 00:00:00
to
Present..

Earliest Entries First: □

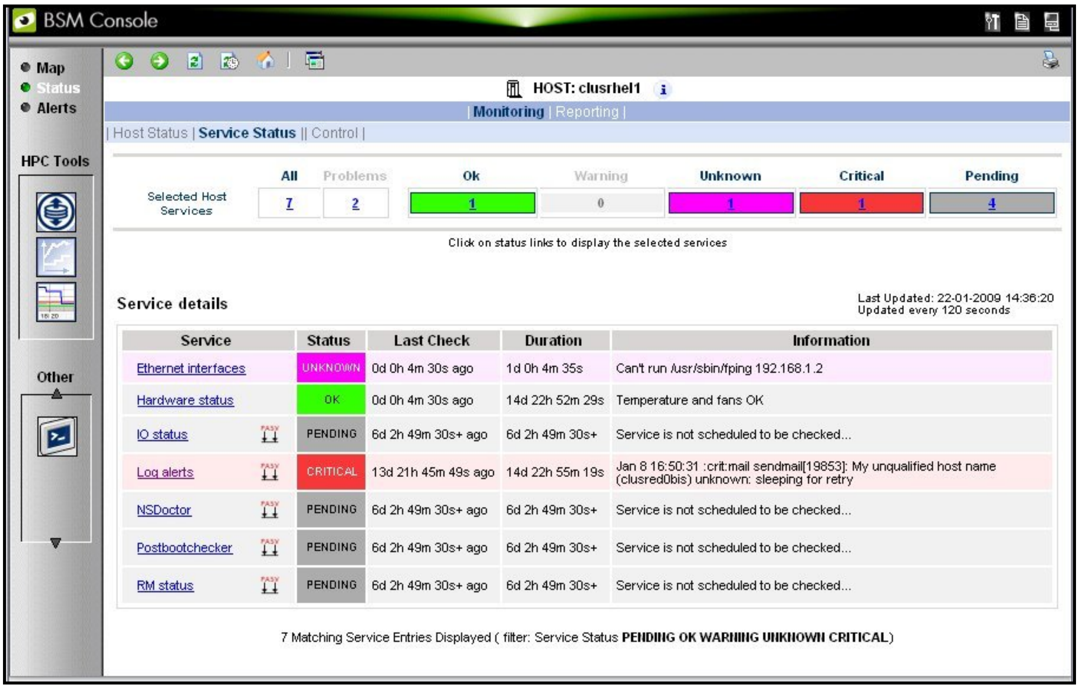
Max Items: 1500

File: /var/log/nagios/nagios.log

```
January 21, 2009 15:00
[21-01-2009 15:32:00] SERVICE ALERT: RACK0-0_p1;Coldoor Power Status;WARNING;HARD;1;Chassis Power is off
Pwr Redundancy :
DPS_0 : Presence detected
DPS_1 : Presence NOT detected
[21-01-2009 15:32:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
[21-01-2009 15:22:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
[21-01-2009 15:12:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
[21-01-2009 15:02:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
January 21, 2009 14:00
[21-01-2009 14:52:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
[21-01-2009 14:46:50] Auto-save of retention data completed successfully.
[21-01-2009 14:42:00] SERVICE ALERT: clusrhel0;Compute power;UNKNOWN;HARD;1;memory_size for node clusrhel1 not defined in database
[21-01-2009 14:32:10] HOST EVENT HANDLER: clusrhel1;(null);(null);(null);host-status-eh
[21-01-2009 14:32:10] HOST ALERT: clusrhel1;UNREACHABLE;HARD;3;Can't run /usr/sbin/fping 192.168.1.2
[21-01-2009 14:32:10] HOST EVENT HANDLER: clusrhel0;(null);(null);(null);host-status-eh
```

Figure 3-7. Monitoring - Log Window

The **Log Window** which is useful for tracing problems appears when the **Monitoring** - **Log** button is clicked. This displays a screen similar to that in *Figure 3-7*.The current Nagios log file is **/var/log/nagios/nagios.log**. The log archives for the preceding weeks is saved **/var/log/nagios/archives**. The Service **Log Alert** window may be displayed by selecting it in the **Service Status** window as shown below.


BSM Console
Map
Status
Alerts
HPC Tools
Other
HOST: clusrhel1
| Monitoring | Reporting |
| Host Status | Service Status || Control |

	All	Problems	Ok	Warning	Unknown	Critical	Pending
Selected Host Services	7	2	1	0	1	1	4

Click on status links to display the selected services

Service details

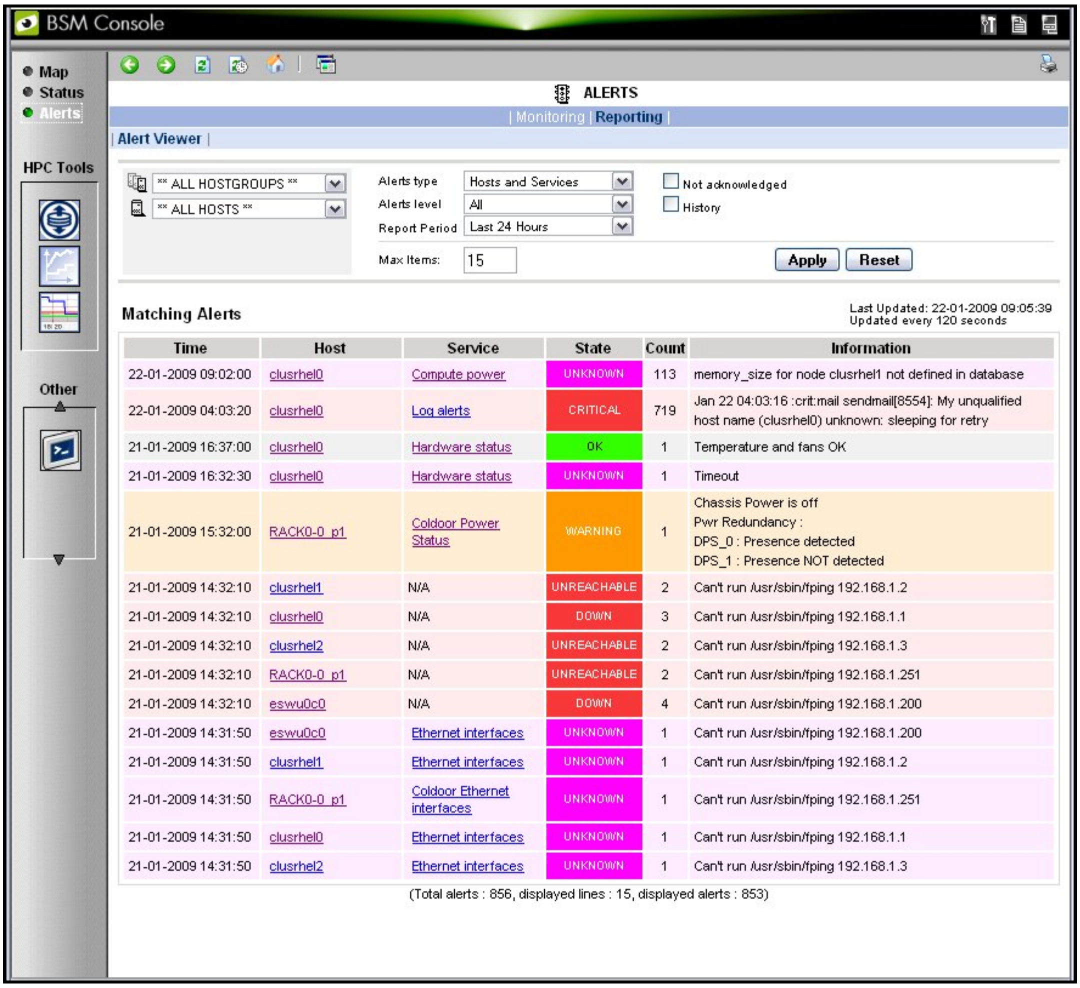
Last Updated: 22-01-2009 14:36:20
Updated every 120 seconds

Service	Status	Last Check	Duration	Information
Ethernet interfaces	UNKNOWN	0d 0h 4m 30s ago	1d 0h 4m 35s	Can't run /usr/sbin/fping 192.168.1.2
Hardware status	OK	0d 0h 4m 30s ago	14d 22h 52m 29s	Temperature and fans OK
IO status	PENDING	6d 2h 49m 30s+ ago	6d 2h 49m 30s+	Service is not scheduled to be checked...
Log alerts	CRITICAL	13d 21h 45m 49s ago	14d 22h 55m 19s	Jan 8 16:50:31 :crit:mail sendmail[19853]: My unqualified host name (clusred0bis) unknown: sleeping for retry
NSDoctor	PENDING	6d 2h 49m 30s+ ago	6d 2h 49m 30s+	Service is not scheduled to be checked...
Postbootchecker	PENDING	6d 2h 49m 30s+ ago	6d 2h 49m 30s+	Service is not scheduled to be checked...
RM status	PENDING	6d 2h 49m 30s+ ago	6d 2h 49m 30s+	Service is not scheduled to be checked...

7 Matching Service Entries Displayed (filter: Service Status **PENDING OK WARNING UNKNOWN CRITICAL**)

Figure 3-8. Monitoring Service Status window for a host with the Log Alerts link highlighted.

3.8 Alerts Button



** ALL HOSTGROUPS **
** ALL HOSTS **
Alerts type: Hosts and Services
Alerts level: All
Report Period: Last 24 Hours
Max Items: 15
□ Not acknowledged
□ History

Matching Alerts

Last Updated: 22-01-2009 09:05:39
Updated every 120 seconds

Time	Host	Service	State	Count	Information
22-01-2009 09:02:00	clusrhel0	Compute power	UNKNOWN	113	memory_size for node clusrhel1 not defined in database
22-01-2009 04:03:20	clusrhel0	Log alerts	CRITICAL	719	Jan 22 04:03:16 :crit:mail sendmail[8554]: My unqualified host name (clusrhel0) unknown: sleeping for retry
21-01-2009 16:37:00	clusrhel0	Hardware status	OK	1	Temperature and fans OK
21-01-2009 16:32:30	clusrhel0	Hardware status	UNKNOWN	1	Timeout
21-01-2009 15:32:00	RACK0-0_p1	Coldoor Power Status	WARNING	1	Chassis Power is off Pwr Redundancy : DPS_0 : Presence detected DPS_1 : Presence NOT detected
21-01-2009 14:32:10	clusrhel1	N/A	UNREACHABLE	2	Can't run /usr/sbin/fping 192.168.1.2
21-01-2009 14:32:10	clusrhel0	N/A	DOWN	3	Can't run /usr/sbin/fping 192.168.1.1
21-01-2009 14:32:10	clusrhel2	N/A	UNREACHABLE	2	Can't run /usr/sbin/fping 192.168.1.3
21-01-2009 14:32:10	RACK0-0_p1	N/A	UNREACHABLE	2	Can't run /usr/sbin/fping 192.168.1.251
21-01-2009 14:32:10	eswu0c0	N/A	DOWN	4	Can't run /usr/sbin/fping 192.168.1.200
21-01-2009 14:31:50	eswu0c0	Ethernet interfaces	UNKNOWN	1	Can't run /usr/sbin/fping 192.168.1.200
21-01-2009 14:31:50	clusrhel1	Ethernet interfaces	UNKNOWN	1	Can't run /usr/sbin/fping 192.168.1.2
21-01-2009 14:31:50	RACK0-0_p1	Coldoor Ethernet interfaces	UNKNOWN	1	Can't run /usr/sbin/fping 192.168.1.251
21-01-2009 14:31:50	clusrhel0	Ethernet interfaces	UNKNOWN	1	Can't run /usr/sbin/fping 192.168.1.1
21-01-2009 14:31:50	clusrhel2	Ethernet interfaces	UNKNOWN	1	Can't run /usr/sbin/fping 192.168.1.3

(Total alerts : 856, displayed lines : 15, displayed alerts : 853)

Figure 3-9. **Alert Window** showing the different alert states

The **Bull System Manager Alert Viewer** application displays monitoring alerts (also called events) for a set of **hostgroups**, **hosts** and **services**.

Alert Types

The alerts can be filtered according to the following alert types:

- Hosts and Services
- Hosts
- Services

Note By default, **Hosts and Services** is selected.

Alerts are visible following the selection of the **Alert** Button, followed by the **Reporting** button, and then by the **Alert Viewer** – see *Figure 3-9*.

Whenever a service or host status change takes place, the monitoring server generates an alert, even when status passes from **CRITICAL** to **RECOVERY** and then to **OK**. Alerts are stored in the current monitoring log and are archived.

Bull System Manager - HPC Edition Alert Viewer utility scans the current monitoring log and archives according to **Report Period** filter settings.

Alert Level

The following **Alert Level** filters are available:

- **All** – Displays all alerts.
- **Major and Minor problems** - Displays Host alerts with **DOWN** or **UNREACHABLE** status levels or displays Service alerts with **WARNING**, **UNKNOWN** or **CRITICAL** status levels.
- **Major problems** -Displays Host alerts with **DOWN** or **UNREACHABLE** status levels or displays Service alerts with **UNKNOWN** or **CRITICAL** status levels.
- **Current problems** -Display alerts with a current non-OK status level. When this alert level is selected, the Time Period is automatically set to '*This Year*' and cannot be modified.

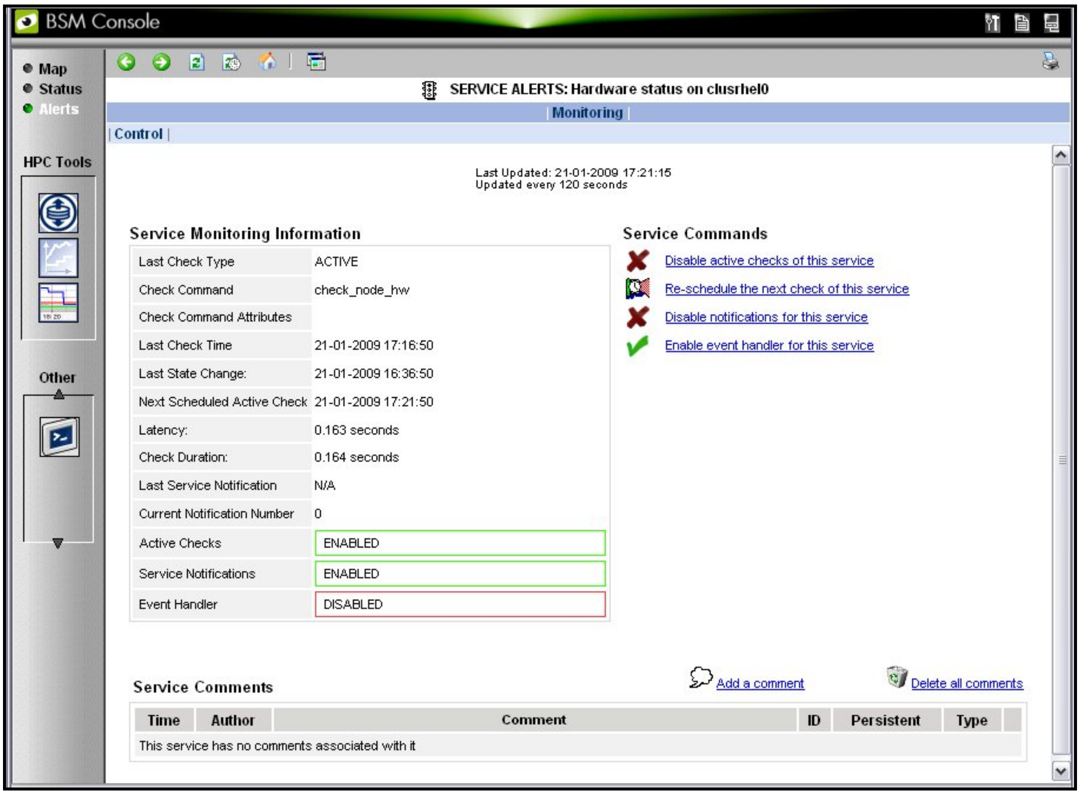
Note By default, **All** is selected.

Report Period

This setting can be changed using the drop down menu.

3.8.1 Active Checks

Active monitoring consists in running a plug-in at regular intervals for a service, this carries out checks and sends the results back to **Nagios**. **Active checks** are set by selecting the **Service** in the **Alert Viewer** window and using the Service Command listed, shown below, to either enable or disable the **Active Check** type.



SERVICE ALERTS: Hardware status on clusrhel0

Last Updated: 21-01-2009 17:21:15
Updated every 120 seconds

Service Monitoring Information

Last Check Type	ACTIVE
Check Command	check_node_hw
Check Command Attributes	
Last Check Time	21-01-2009 17:16:50
Last State Change:	21-01-2009 16:36:50
Next Scheduled Active Check	21-01-2009 17:21:50
Latency:	0.163 seconds
Check Duration:	0.164 seconds
Last Service Notification	N/A
Current Notification Number	0
Active Checks	ENABLED
Service Notifications	ENABLED
Event Handler	DISABLED

Service Commands

- ✗ Disable active checks of this service
- Re-schedule the next check of this service
- ✗ Disable notifications for this service
- ✓ Enable event handler for this service

Service Comments

Add a comment Delete all comments

Time	Author	Comment	ID	Persistent	Type
This service has no comments associated with it					

Figure 3-10. **Monitoring Control** Window used to set **Active Checks** for a Service

The **Nagios plug-in** returns a code corresponding to the **Alert** alarm state. The state is then displayed in a colour coded format, in the **Alert Viewer** window - see *Figure 3-9* - as follows:

0 for **OK/UP** (Green background)
1 for **WARNING** (Orange background)
2 for **CRITICAL/DOWN/UNREACHABLE** (Red background)
3 for **UNKNOWN** (Violet background)

The plug-in also displays an explanatory text for the alarm level in the adjacent **Information** column.

3.8.2 Passive Checks

With this form of monitoring a separate third-party program or plug-in will keep Nagios informed via its external command file (**/var/spool/nagios/nagios.cmd**). It submits the result in the form of a character string which includes a timestamp, the name of the **Host** and/or **Service** concerned, as well as the return code and the explanatory text.

Passive checks appear with a GREY background in the list of alerts.

3.8.3 Alert Definition

The different parameters which may be used for an alert are as follows:

$HOSTNAME$: The name of the host from which the alert is returned.

$HOSTALIAS$: The content of the comma separated field ':'

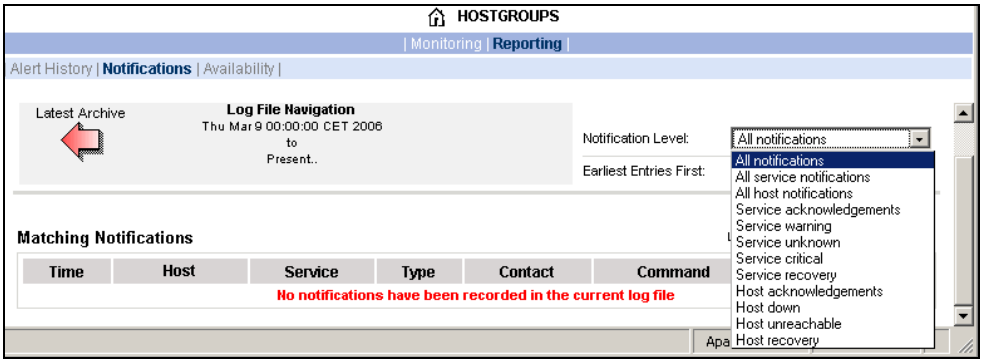
For a node this is: **node:<type>:<model>**
with **<type>** = for example A—, -C–, AC-M-
with **<model>** = for example NS423.

For an Ethernet switch: **eth_switch:<model>**
with **<model>** = for example. CISCO 3750G24TS.

For an interconnect switch : **ic_switch:<model>**
with **<model>** = for example the type of material (**node**, **eth_switch**, **ic_switch**).

3.8.4 Notifications

Notifications are sent out if a change or a problem occurs. The Notification may be one of 3 types - e-mail, **SNMP** trap, or via a User Script. Set the **<notification_interval>** value to 0 to prevent notifications from being sent out more than once for any given problem or change.
The **Monitoring Control** window - see *Figure 3-10* provides the facility to Enable or Disable notifications.
The Notification level is set in the Maps →Hostgroups → Reporting →Notifications window. The different notification levels are indicated below.



Log File Navigation
Thu Mar 9 00:00:00 CET 2006
to
Present..

Notification Level: All notifications

Earliest Entries First:

Matching Notifications

Time	Host	Service	Type	Contact	Command

No notifications have been recorded in the current log file

Figure 3-11. Hostgroups Reporting Notifications Window showing the Notification Levels

3.8.5 Acknowledgments

As the **Administrator**, you may choose whether or not alerts are acknowledged, and decide whether they should be displayed or not.

3.8.6 Running a Script

Bull System Manager - HPC Edition can be configured to run a script when a state changes or an alert occurs. User scripts which define events or physical changes to trigger **Nagios** alerts may also be used. More information on scripts or third party plug-ins is available in the documentation from http://www.nagios.org/docs/

Below is an example of script.

```
#!/usr/bin/perl -w

# Arguments : $SERVICESTATE$ $STATETYPE$ $HOSTNAME$ $HOSTSTATE$ $OUTPUT$

$service_state = shift;
$state_type = shift;
$host_name = shift;
$host_state = shift;
$output = join(" ", @ARGV);

# Sanity checks
if ($state_type !~ "HARD") { exit 0; }
if ($service_state !~ "WARNING" && $service_state !~ "CRITICAL") {
  exit 0;
}

# Launch NSDoctor if needed
if ($host_state =~ "UP" &&
    $output =~ /automatically configured out|no response/) {
  system("/usr/sbin/nsdoctor.pl $host_name");
}
exit 0;
```

In order that e-mail alerts are sent whenever there is a problem, a SMTP server, for example **PostFix** or **Sendmail**, has to be running on the Management node.

By default, the e-mail alerts are sent to nagios@localhost on the Management Node. Normally, by default, only the cluster administrators will receive the alerts for each change for all the Hosts and Services. To send e-mails alerts to other addresses, create the new contacts, and add them to the contact groups. The files to modify are **/etc/nagios/contacts.cfg** and **/etc/nagios/contactgroups.cfg**.

3.8.7 Generating SNMP Alerts

When **Bull System Manager - HPC Edition** receives an alert (Service in a **WARNING** or **CRITICAL** state, Host in **DOWN** or **UNREACHABLE** state), the event handler associated with the service or host sends an SNMP trap, using the **snmptrap** command.
The Management Information Base (**MIB**) is available in the file **/usr/share/snmp/mibs/NSMASTERTRAPMIB.txt**. This describes the different types of traps and the information that they contain.

In order that an SNMP trap is sent the following actions should be performed:

1. Add the IP address of the host(s) that will receive the traps in the **/etc/nagios/snmptargets.cfg** file (one address per line).
2. Add the contact that will receive the traps to a contact group. To do this, edit the **/etc/nagios/contactgroups.cfg** file and change the line:
   ```
   members  nagios
   ```
   in:
   ```
   members   nagios,snmptl
   ```
3. Restart nagios:

```
service nagios reload
```

3.8.8 Resetting an Alert Back to OK

To reset an alert back to zero click the Service or the Host concerned, then on the menu **Submit passive check result for this service**. Set the **Check Result** to OK, if this is not already the case, fill in the **Check Output** field with a short explanation, and then click the **Commit** button. The return to the **OK** state will be visible once Nagios has run the appropriate command.

3.8.9 nsmhpc.conf Configuration file

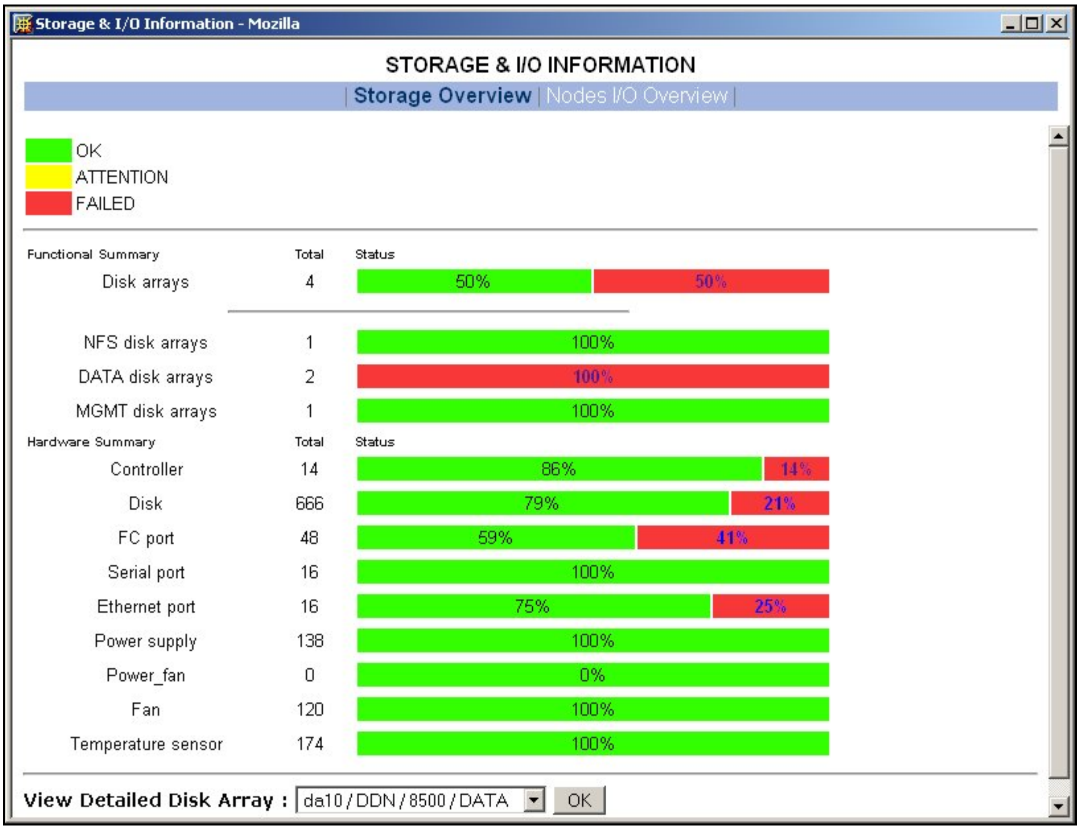
The **/etc/nsmhpc/nsmhpc.conf** file contains several configuration parameters. Most of them have default values, but for some services the administrator may have to define specific parameter values. A message will inform the administrator if a value is missing.

3.8.10 Comments

Users of a particular host or service can post comments from the **Monitoring Control** window - see *Figure 3-10*

3.9 Storage Overview

Select the **Storage Overview** button in the vertical toolbar on the left hand side to display information similar to that shown below.



Storage & I/O Information - Mozilla

STORAGE & I/O INFORMATION

| Storage Overview | Nodes I/O Overview |

🟩 OK
🟨 ATTENTION
🟥 FAILED

Functional Summary	Total	Status
Disk arrays	4	🟩 50% 🟥 50%
NFS disk arrays	1	🟩 100%
DATA disk arrays	2	🟥 100%
MGMT disk arrays	1	🟩 100%
Hardware Summary	**Total**	**Status**
Controller	14	🟩 86% 🟥 14%
Disk	666	🟩 79% 🟥 21%
FC port	48	🟩 59% 🟥 41%
Serial port	16	🟩 100%
Ethernet port	16	🟩 75% 🟥 25%
Power supply	138	🟩 100%
Power_fan	0	🟩 0%
Fan	120	🟩 100%
Temperature sensor	174	🟩 100%

View Detailed Disk Array : da10 / DDN / 8500 / DATA [OK]

Figure 3-12. Storage overview window

More detailed information is provided by clicking on the ATTENTION and FAILED sections of the component summary status bars.

3.10 Shell

The **Shell** button can be used to open a command shell on the Management Node.

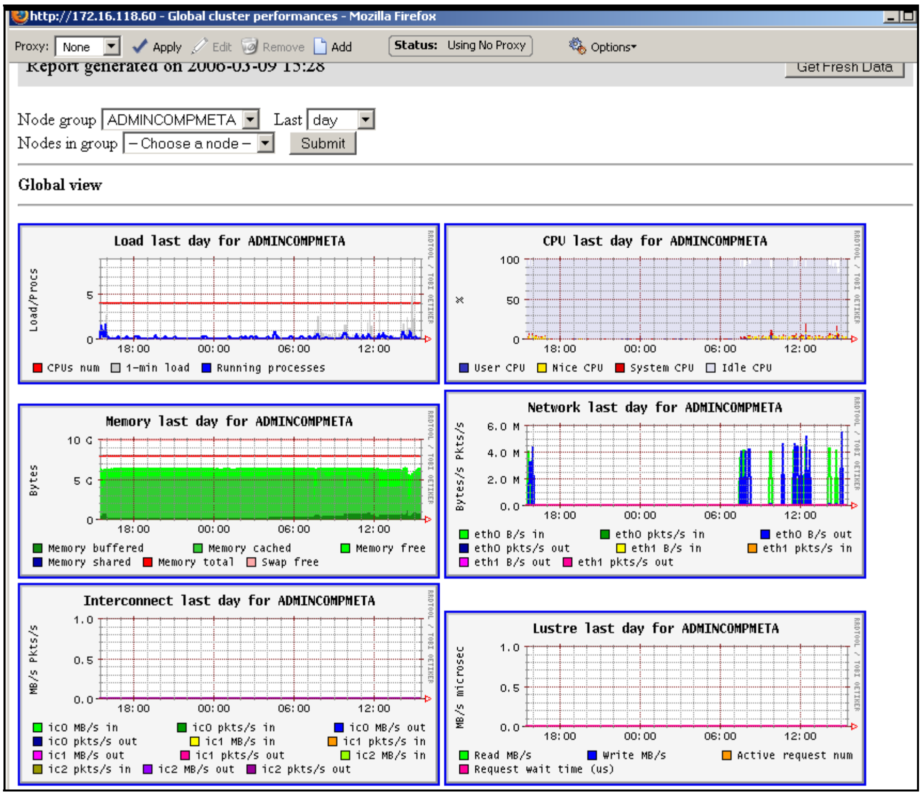
3.11 Monitoring the Performance - Ganglia Statistics

Bull System Manager - HPC Edition provides the means to visualize the performance for the cluster by clicking the icons in the vertical left hand tool bar – see *Figure 3-1*. This can be done either for a **Global Performance View**, which displays data either for a complete cluster or on a node by node basis, or in a **Group Performance View**. These views enable the statistical examination of a predefined group of nodes in the database.

The parameters which enable the calculation of the performance of the cluster are collected on all the nodes by **Ganglia** and are displayed graphically. One can also define the observation period and display the measurement details for a particular node using the Ganglia interface.

3.12 Group Performance View

This view displays the Group Performance for 6 different metric types for the complete cluster, as shown below. Using this view it is possible to see view the nodes in groups, and then to zoom to a particular node.



http://172.16.118.60 - Global cluster performances - Mozilla Firefox

Proxy: None

Status: Using No Proxy

Report generated on 2006-03-09 15:28

Node group ADMINCOMPMETA Last day

Nodes in group – Choose a node –

Global view

Figure 3-13. Group Performance view

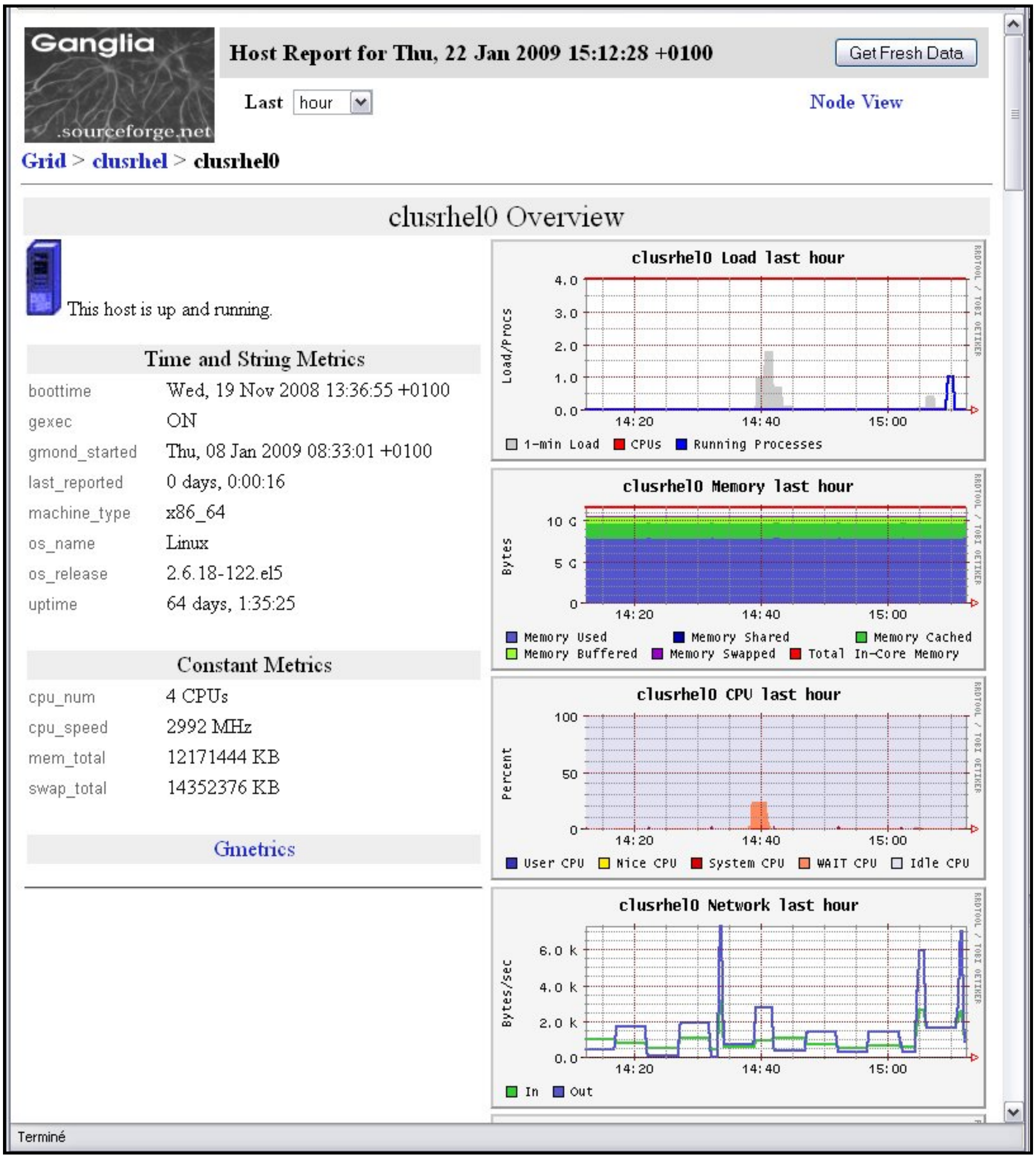
3.13 Global Performance View

The **Global Performance** view gives access to the native interface for **Ganglia**, and provides an overall view of the cluster. It is also possible to view the performance data for individual nodes.

Five categories of data collected. These are:

- Load for CPUS and running processes
- Memory details
- Processor activity
- Network traffic in both bytes and packets
- Storage.

Each graph shows changes for the performance metrics over a user defined period of time.



Host Report for Thu, 22 Jan 2009 15:12:28 +0100

Last hour

Grid > clusrhel > clusrhel0

clusrhel0 Overview

This host is up and running.

Time and String Metrics

boottime	Wed, 19 Nov 2008 13:36:55 +0100
gexec	ON
gmond_started	Thu, 08 Jan 2009 08:33:01 +0100
last_reported	0 days, 0:00:16
machine_type	x86_64
os_name	Linux
os_release	2.6.18-122.el5
uptime	64 days, 1:35:25

Constant Metrics

cpu_num	4 CPUs
cpu_speed	2992 MHz
mem_total	12171444 KB
swap_total	14352376 KB

Gmetrics

Figure 3-14. Global overview for a host (top screen)

More detailed views are shown by scrolling the window down – see *Figure 3-15*.

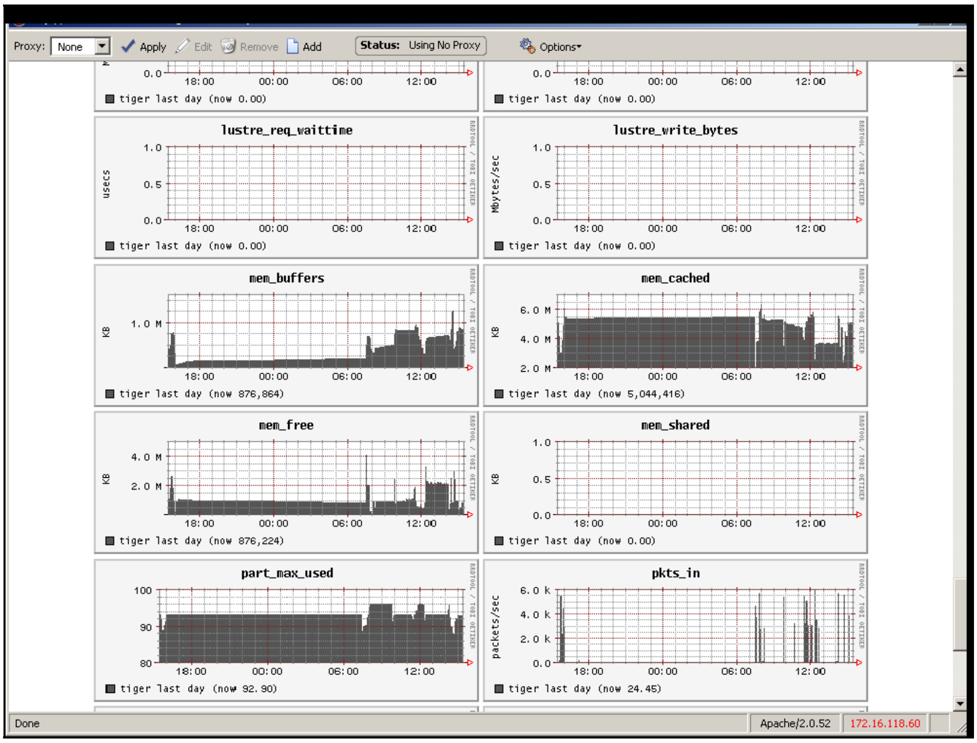

Proxy: None
Apply
Edit
Remove
Add
Status: Using No Proxy
Options
tiger last day (now 0.00)
tiger last day (now 0.00)
lustre_req_waittime
usecs
tiger last day (now 0.00)
lustre_write_bytes
Mbytes/sec
tiger last day (now 0.00)
mem_buffers
KB
tiger last day (now 876,864)
mem_cached
KB
tiger last day (now 5,044,416)
mem_free
KB
tiger last day (now 876,224)
mem_shared
KB
tiger last day (now 0.00)
part_max_used
tiger last day (now 92.90)
pkts_in
packets/sec
tiger last day (now 24.45)
18:00
00:00
06:00
12:00
Done
Apache/2.0.52
172.16.118.60

Figure 3-15. Detailed monitoring view for a host (bottom half of screen displayed in *Figure 3-14*)

3.13.1 Modifying the Performance Graph Views

The format of the graphs displayed in the performance views can be modified by editing the file /usr/share/nagios/conf.inc. The section which follows the line **Metrics** enumeration defines the different graphs; each graph is created by a call to the producer of the Graph class. To create a new graph, it is necessary to add the line:

```
$myGraph = new Graph("<graphname>")
```

<graphname> is the name given to graph.

To specify a metric to the graph, the following command must be edited as many times as there are metrics to be added or changed:

```
$myGraph->addMetric(new Metric("<metricname>", "<legende>",
"<fonction>", "<couleur>", "<trait>"))
```

<metricname> The name given by Ganglia for the metric.

<legende> Text displayed on the graph to describe the metric.

<fonction> Aggregating function used to calculate the metric value for a group of nodes, currently the functions **sum** and **avg** are supported.

<couleur> HTML color code.

<trait> style for feature displayed (**LINE1**, **LINE2**, **AREA**, **STACK**), See the man page for **rrdgraph** for more details.

Use the command below to add the graph to those which are displayed:

```
graphs:$graphSet->addGraph($myGraph)
```

3.13.2 Refresh Period for the Performance View Web Pages

By default the refresh period is 90 seconds. This can be modified by changing the value for the parameter **refresh_rate** in the file **/etc/nagios/cgi.cfg**.

3.14 Configuring and Modifying Nagios Services

3.14.1 Configuring Using the Database

The command used to regenerate the **Nagios** services Database configuration files is:

```
/usr/sbin/dbmConfig configure --service Nagios --restart
```

This command will also restart **Nagios** after the files have been regenerated.

Use the following command to test the configuration:

```
service nagios configtest
```

Important

The services are activated dynamically according to the Cluster type and the functionalities which are detected. For example, the services activated for Quadrics clusters will be different from those which are activated for InfiniBand clusters.

3.14.2 Modifying Nagios Services

The list and configuration of **Nagios** services is generated from the database and from the file **/etc/nagios/services-tpl.cfg**. This file is a template used to generate the complete files. All template modifications require the **Nagios** configuration file to be regenerated using the command:

```
dbmConfig configure --service nagios
```

Note To check that all services have been taken into account, you can use the **dbmServices** command (this command is described in the *Cluster Database Management* chapter in the present guide). If the services have not been taken into account then enter the following commands:

```
/usr/lib/clustmngt/clusterdb/bin/nagiosConfig.pl –init
dbmConfig configure --service nagios
```

Refer to http://nagios.sourceforge.net/docs/3_0/checkscheduling.html for more information on configuring the services.

3.14.2.1 Clients without Customer Relationship Management software

If a **CRM** product is not installed then the **Nagios** configuration files will have to be changed to prevent the system from being overloaded with error messages. This is done as follows:

1. Edit the **/etc/nagios/contactgroups** file and change the line which reads **members nagios,crmwarn,crmcrit** so that it reads **members nagios**
2. In the **/etc/nagios/nagios.cfg** file change the status of the line **process_performance_data=1** so that it is commented.

3.14.3 Changing the Verification Frequency

Usually the application will require that the frequencies of the **Nagios** service checks are changed. By default the checks are carried out once every ten minutes, except on certain services. To change this frequency, the **normal_check_interval** parameter has to be added to the body of the definition of the service and then modified accordingly.

3.14.4 Nagios Services Service

The **Nagios services** service monitors the daemons required for its own usage. If one of them is not up and running, this service will display the CRITICAL state and indicates which daemons are unavailable. The administrator must define a parameter stored in the **/etc/nsmhpc/nsmhpc.conf file**:

nagios.services, which defines the daemons which are monitored by the plugin (the default value is **syslog-ng snmpd snmptrapd**).

3.14.5 Nagios Information

See the **Nagios** documentation for more information, in particular regarding the configuration. Look at the following web site for more information http://nagios.sourceforge.net/docs/3_0/

In addition look at the **Bull System Manager - HPC Edition** documentation suite, this includes an *Installation Guide*, a *User's Guide*, an *Administrator's Guide* and a *Remote Hardware Management CLI Reference Manual.*

3.15 General Nagios Services

Nagios includes a wide range of plug-ins, each of which provides a specific monitoring service that is displayed inside the graphical interface. In addition Bull has developed additional monitoring plug-ins which are included within **Bull System Manager – HPC Edition**. The plug-ins and corresponding monitoring services are listed below.
The services listed in this section apply to all node types. The **Ethernet Interfaces** service applies to all forms of material/devices.

3.15.1 Ethernet Interfaces

The Ethernet interfaces service indicates the state of the Ethernet interfaces for a node. The plug-in associated with this service is **check_fping** which runs the **fping** command for all the Ethernet interfaces of the node. If all the interfaces respond to the ping, the service posts OK. If **N** indicates the total number of Ethernet interfaces, and at least **1** or at most **N-1** interfaces do not answer, then the service will display **WARNING**.

3.15.2 Resource Manager Status

The service reports the state of the node as seen by the Resource Manager (for example **SLURM**) which is in place. The service will be updated every time the state of the node changes.

3.15.3 Hardware Status

The material status (temperature and fan status) of each node is posted to the passive Hardware status service, resulting from information from the **check_node_hw.pl** plug-in which interfaces with the **BMC** associated with the node.

3.15.4 Temperature

The temperature service checks the temperature of the node. Currently, only the temperatures of four QBB boards are monitored by means of the **nsminfo** command. This service returns an alarm equivalent to the level of the worst **QBB** board state and posts for each state the number or boards which are in this state. The different possible states are **NORMAL, WARNING, CRITICAL, FATAL, UNKNOWN**.

3.15.5 Alert Log

The **Log alerts** passive service displays the last alarm raised by system log for the machine – see *Section 3.7* . A mapping is made between the **syslog** severity levels and the **Nagios** alarm levels: **OK** gathers info, debug and notice alarms; **WARNING** gathers warn and err alarms; **CRITICAL** gathers **emerg**, **crit**, **alert**, **panic** alerts.

3.15.6 I/O Status

The I/O status reports the global status of HBA, disks and LUNs on a cluster node.

3.15.7 Postbootchecker

The **postbootchecker** tool carries out various analyses after a node is rebooted. It communicates the results of its analyses to the corresponding passive service.

3.16 Management Node Nagios Services

These services are available on the Management Node only.

3.16.1 MiniSQL Daemon

This active service uses the **check_proc** plug-in to verify that the **msql3d** process is functioning correctly. It remains at the **OK** alert level, whilst the daemon is running but switches to **CRITICAL** if the daemon is stopped.

3.16.2 Resource Manager Daemon

This active service uses the **check_proc** plug-in to verify that the **RMSD** process (**Quadrics** clusters), or the **SLURMCLTD** (**InfiniBand** clusters) process, is functioning correctly. It remains at the **OK** alert level whilst the daemon is running but switches to **CRITICAL** if the daemon is stopped.

3.16.3 Quadrics Switch Manager

This active service uses the **check_proc** plug-in to verify that the **swmgr** process is functioning correctly. It remains at the **OK** alert level whilst the daemon is running but switches to **CRITICAL** if the daemon is stopped.

3.16.4 ClusterDB

This active service uses the **check_clusterdb.pl** plug-in to check that connection to the Cluster Database is being made correctly. It remains at the **OK** alert level whilst the connection is possible, but switches to **CRITICAL** if the connection becomes impossible.

3.16.5 Cron Daemon

This active service uses the **check_proc** plug-in to verify that the **cron** daemon is running on the system. It remains at the **OK** alert level whilst the daemon is running but switches to **CRITICAL** if the daemon is stopped.

3.16.6 Compute Power Available

A Bull plug-in checks the compute power available, and the Alert level associated with it, and then displays the results in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

This plug-in is specific to the **COMP** group of nodes created by the use of the **dbmConfig** command and which consists of all the Compute Nodes in the Cluster database. Note that Login nodes are considered as Compute Nodes in the **Clusterdb**, and if the Login nodes have not been defined in a Compute partition then the **COMP** group of nodes should be deleted by using the **dbmGroup modify** command.

3.16.7 Global File System bandwidth available

A Bull plug-in checks the bandwidth for the Global File System, and the Alert level associated with it, and then displays the results in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

3.16.8 Storage Arrays available

A Bull plug-in checks how much space is available for the storage arrays, and the Alert level associated with it, and then displays the results in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

3.16.9 Global File System Usage

A Bull plug-in checks Global File System Usage, and the Alert level associated with it, and then displays the results in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

3.16.10 I/O pairs Migration Alert

A Bull plug-in checks the I/O pairs status, and the Alert level associated with it, and then displays the results in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

3.16.11 Backbone Ports Available

This service calculates the percentage of ports which are usable for the backbone switches. All the ports which are not usable have to be in the state **administratively down**.
The results are displayed in the **Availability Indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.

3.16.12 Quadrics Ports Available

This service calculates the number of ports (external and internal links) which are available for the Quadrics switches. These are displayed as a percentage of the total number of **Quadrics** ports for the cluster.

Note When internal links are included the total number of ports shown will be greater than the number of physical ports possible for a **Quadrics** switch. For example, a switch with 32 ports will display 96 ports within **Bull System Manager – HPC Edition**.

3.16.13 HA System Status

This service is based on the output of the **clustat** command. It displays the state of the Management Nodes which are running with High Availability. As soon as one or more management nodes rocks to the '*offline*' state the service displays a list of all the nodes in the 'offline' state and returns an alert level of **CRITICAL**. If all the Management Nodes are 'online' then the service returns **OK**.

3.16.14 Kerberos KDC Daemon

This active service uses the plug-in **check_proc** to check if the daemon **krb5kdc** is running on the system. It remains at the **OK** alert level whilst the daemon is running, but switches to **CRITICAL** if the daemon stops.

3.16.15 Kerberos Admin Daemon

This active service uses the plug-in **check_proc** to check if the **kadmind** daemon is running on the system. It remains at the **OK** alert level whilst the daemon is running, but switches to **CRITICAL** if the daemon stops.

3.16.16 LDAP Daemon (Lustre clusters only)

This active service checks if the **check_ldap** plug-in which the Lightweight Directory Access Protocol **(LDAP)** uses with **Lustre** is working correctly. This plug-in makes a connection to **LDAP** using **fs=lustre** as root for the naming hierarchy.

3.16.17 Lustre file system access

This is a passive service which is run every 10 minutes by a cron. The cron connects to a client node taken from a specified group at random, for example a Compute Node, and attempts to create and write (stripe) a file on all the **Lustre** file system directories that are listed in the Cluster DB, and that are mounted on the node. The file is deleted at the end of the test. If the operation is successful an **OK** code is sent to Nagios with the message '*All Lustre file systems writable*'. If not, a **CRITICAL** code is returned with the message *'Lustre problem detected'*.
The service uses the **lustreAccess.group** parameter, defined in the **/etc/nsmhpc/nsmhpc.conf** file, to specify the group containing the nodes that can be used for the test (default: COMP).

3.16.18 NFS file system access

This is a passive service which is run every 10 minutes by a cron. The cron connects to a client node taken from a specified group at random, for example a Compute Node, and looks for all the NFS filesystems mounted on this node. Then it tries to create and write a file in a specified sub-directory, on all NFS filesystems. The file is deleted at the end of the test. If the operation is successful an **OK** code is sent to Nagios. If not, a **CRITICAL** code is returned with detailed information.

The service uses three parameters, defined in the **/etc/nsmhpc/nsmhpc.conf** file:

- **nfsAccess.group**, which specifies the group containing the nodes that can be used for the test (default: COMP).
- **nfsAccess.directory**, which specifies an existing sub-directory in the filesystem where the test file will be created.
- **nfsAccess.user**, which specifies a user authorized to write in the sub-directory defined in the **nfsAccess.directory** parameter.

3.16.19 InfiniBand Links available

This service calculates the percentage of links that are usable for the **InfiniBand** switches. The results are displayed in the **Availability indicators** view pane on the top right hand side of the opening window for the **Map** button as shown in *Figure 3-2*.
The administrator must specify two parameters in the **/etc/nsmhpc/nsmhpc.conf** file:

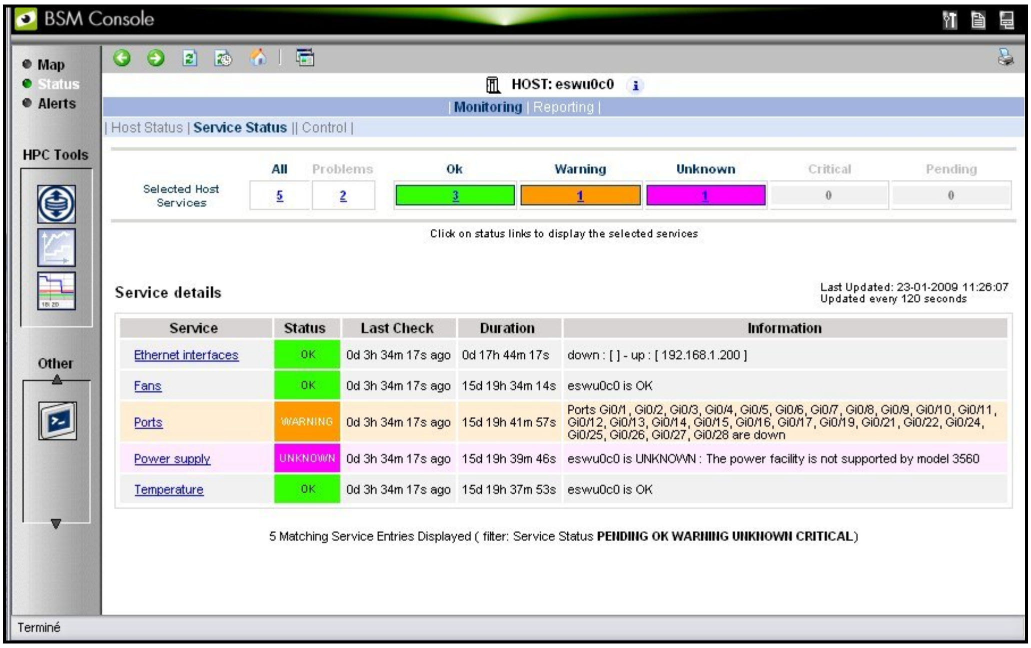
- **indicator.ib.numUpLinks**, which specifies the number of installed up links (top switches <-> bottom switches)
- **indicator.ib.numDownLinks**, which specifies the number of installed down links (bottom-switches <-> nodes)

According to these values and the values returned by the **IBS** tool, the service will be able to define the availability of the **InfiniBand** interconnects.

See The *InfiniBand Guide* for more information regarding the **IBS** tool.

3.17 Ethernet Switch Services

The Ethernet switches which are not used should be set to *disabled* so that Ethernet switch monitoring works correctly. This is usually done when the switches are first configured. The services for the switch are displayed when it is selected in either the cluster **HOSTGROUP** or **HOST** window, followed by the selection of **Service Status** window, as shown below.


BSM Console
Map
Status
Alerts
HPC Tools
Other
HOST: eswu0c0
| Monitoring | Reporting |
| Host Status | Service Status || Control |

	All	Problems	Ok	Warning	Unknown	Critical	Pending
Selected Host Services	5	2	3	1	1	0	0

Click on status links to display the selected services

Service details

Last Updated: 23-01-2009 11:26:07
Updated every 120 seconds

Service	Status	Last Check	Duration	Information
Ethernet interfaces	OK	0d 3h 34m 17s ago	0d 17h 44m 17s	down : [ ] - up : [192.168.1.200]
Fans	OK	0d 3h 34m 17s ago	15d 19h 34m 14s	eswu0c0 is OK
Ports	WARNING	0d 3h 34m 17s ago	15d 19h 41m 57s	Ports Gi0/1 , Gi0/2, Gi0/3, Gi0/4, Gi0/5, Gi0/6, Gi0/7 , Gi0/8, Gi0/9, Gi0/10, Gi0/11 , Gi0/12, Gi0/13, Gi0/14, Gi0/15, Gi0/16, Gi0/17, Gi0/19, Gi0/21 , Gi0/22, Gi0/24, Gi0/25, Gi0/26, Gi0/27, Gi0/28 are down
Power supply	UNKNOWN	0d 3h 34m 17s ago	15d 19h 39m 46s	eswu0c0 is UNKNOWN : The power facility is not supported by model 3560
Temperature	OK	0d 3h 34m 17s ago	15d 19h 37m 53s	eswu0c0 is OK

5 Matching Service Entries Displayed (filter: Service Status **PENDING OK WARNING UNKNOWN CRITICAL**)

Terminé

Figure 3-16. **Ethernet Switch** services

3.17.1 Ethernet Interfaces

The **Ethernet interfaces** service checks that the Ethernet switch is responding by using a ping to its IP address.

3.17.2 Fans

The **Fans** service monitors the fans for the Ethernet switches using the **check_esw_fans.pl** plug-in.

3.17.3 Ports

The **Ports** service monitors the ports for the switches. If one or more ports are detected as being in a *notconnect* state, this service will display the **WARNING** state and indicate which ports are unavailable.

3.17.4 Power supply

The **Power supply** service checks the power supply is functioning properly by using the **check_esw_power.pl** plug-in.

3.17.5 Temperature

The **Temperature** service monitors the temperatures of the Ethernet switches by using the **check_esw_temperature.pl** plug-in.

3.18 Portserver Services

The Portserver has to be configured to send traps. The **Current threshold exceeded** and **Temperature threshold exceeded** traps have to be activated and the destination address for the traps has to be the IP address of the management node (or the IP alias if High Availability is in place).

The configuration settings are:

```
set snmp trap_dest=<@IP noeud d'admin>
set snmp curr_thresh_exc_trap=on
set snmp temp_thresh_exc_trap=on
set snmp login_trap=off
set snmp auth_trap=off
set snmp cold_start_trap=off
set snmp link_up_trap=off
```

3.18.1 Temperature Threshold Service

This passive service indicates when the temperature threshold has been crossed by the portserver. The alert generates a SNMP trap which moves up the temperature monitoring service.

3.18.2 Current Threshold Service

This passive service indicates when the current threshold tolerated for a port has been exceeded by the portserver. The alert generates a SNMP trap which moves up the current monitoring service.

Chapter 4. Parallel Libraries

BAS4 V5.1 Fix12 delivers new versions for **MPI_Bull** (1.6.x) and **MPIBull2** (1.3.x). Both include bug fixes, and the tuning is improved for MPIBull2. But no new functions are delivered.

See *BAS4 User's Guide* for details on Parallel Libraries.

This chapter includes the following sections:

- *Dynamic Process Services*
  This section complements the MPIBull2 Advanced features already described in the *BAS4 User's Guide.*
- *MPIBull2 and NFS Clusters*
  This section gives additional information related to the use of MPI and NFS together.

4.1 Dynamic Process Services

The main goal of dynamic process services is to provide a means to develop software using multi-agent or master/server paradigms. They provide a mechanism to establish communication between newly created processes and an existing MPI application (**MPI_COMM_SPAWN**). They also provide a mechanism to establish communication between two existing **MPI** applications, even when one did not 'start' the other (**MPI_PUBLISH_NAME**).

MPI_PUBLISH_NAME structure

MPI_PUBLISH_NAME (service_name, info, port_name)

IN	service_name	a service name to associate with the port (string)
IN	info	implementation-specific information (handle)
IN	port_name	a port name (string)

Although these paradigms are useful for Extreme Computing clusters there may be a performance impact. **MPIBull2** includes Dynamic Process Services, but with some restrictions:

- Only the **osock** socket **MPI** driver can be used with dynamic processes.
- A **PMI** server implementing spawn answering routines must be used as follows.
  - For all Bull clusters the **MPD** sub-system is used
  - For clusters which use **SLURM**, a **MPD** ring must be deployed once SLURM's allocation has been guaranteed.
  - **PBS Professional** clusters can use **MPD** without any restrictions.
- The quantity of processes which can be spawned depend on the reservation previously allocated with the Batch Manager /Scheduler (if used).

See The chapter on *Process Creation and Management* in the **MPI-2.1** Standard documentation available from http://www.mpi-forum.org/docs/ for more information.

MPI Ports Publishing Example

	Sever	Client
Command	`mpiexec -n 1 ./server`	`mpiexec -n 4 ./toy`
Process	(MPI_Open_port) + (MPI_Publish_name) MPIBull2 1.3.9-s (Astlik) MPI_THREAD_FUNNELED (device osock) Server is waiting for connections (MPI_Comm_accept) Master available, Received from 0 Now time to merge the communication (MPI_Comm_merge) Establish communication with 1st slave Accept communication to port Slave 1 available Slave 2 available Disconnected from slave, Send message to Master Slave 3 available Disconnected from slave, Send message to Master (MPI_Comm_Unpublish_name) (MPI_Close_Port)	MPIBull2 1.3.9-s (Astlik) MPI_THREAD_FUNNELED (device osock) (MPI_Get_attribute) Got the universe size from server (MPI_Lookup_name) Lookup found service attag#0$port#35453$description#10.11.0.11 $ifname#10.11.0.11$ port [x4] (MPI_Comm_connect) + (MPI_Send / MPI_Recv) Sent stuff to the commInter Recv stuff to the commInter Master Process at work, merge comm Master: number of tasks to distribute: 10 Sent a message to the following MPI process Sent stuff to the commInter Recv stuff to the commInter Slave Process at work, merge comm Sent stuff to the commInter Recv stuff to the commInter Slave Process at work, merge comm Sent stuff to the commInter Recv stuff to the commInter Slave Process at work, merge comm Process 1 with 1 Threads runs at work 1: Got task from 900001 to 1000000 Merged and disconnected (MPI_Comm_disconnect) Assigned tasks: —0 0–1 [x10] [compute] I give up 3: Wallclock Time: 45.2732 1: Wallclock Time: 45.2732 Unpublishing my service toyMaster

		2: Wallclock Time: 45.2732 Closing my port of connection (master) master disconnected from 1 master disconnected from 2 master disconnected from 3 Master with 1 Threads joins computation (univ: 1) disconnected from server 0: Wallclock Time: 45.2757

4.2 MPIBull2 and NFS Clusters

To use **MPI** and **NFS** together, the shared NFS directory must be mounted with the no attribute caching (**noac**) option added; otherwise the performance of the Input/Output operations will be impacted. To do this, edit the **/etc/fstab** file for the **NFS** directories on each client machine in a multi-host **MPI** environment.

Note All the commands below must be carried out as root.

Run the command below on the NFS client machines:

```
grep nfs_noac /etc/fstab
```

The **fstab** entry for **/nfs_noac** should appear as below:

```
/nfs_noac /nfs_noac nfs bg,intr,noac 0 0
```

If the **noac** option is not present, add it and then remount the **NFS** directory on each machine using the commands below.

```
umount /nfs_noac
mount /nfs_noac
```

To improve performance, export the **NFS** directory from the **NFS** server with the **async** option. This is done by editing the **/etc/exports** file on the **NFS** server to include the **async** option, as below.

Example

The following is an example of an export entry that includes the **async** option for **/nfs_noac**:

```
grep nfs_noac /etc/exports
```

```
 /nfs_noac       *(rw,async)
```

If the **async** option is not present, add it and export the new value:

```
exportfs -a
```

Chapter 5. Scientific Studio

Bull **Scientific Studio** is included in the **BAS4 V5.1 Fix 12** delivery and includes Open Source libraries that can be used to facilitate the development and execution of a wide range of applications.

Proprietary scientific libraries, that have to be purchased separately, are available from **Intel**®.

Important

Only the new Scientific Libraries for BAS4 V5.1 Fix12 are described in this chapter. See the BAS4 V5.1 Fix11 User's Guide (Reference 86 A2 29ER 09) for details of the Scientific Libraries previously delivered.

See

- The *BAS4v5.1 Fix 12 Software Release Bulletin* for the details of the Scientific Library versions delivered.
- The **Intel**® web site for details of the **Intel**®**MKL** libraries installed on your cluster.

5.1 Bull Scientific Studio

Bull Scientific Studio is based on the Open Source Management Framework (**OSMF**), and provides an integrated set of up-to-date and tested mathematical scientific libraries that can be used in multiple environments. They simplify modeling by fixing priorities, ensuring the cluster is in full production for the maximum amount of time, and are ideally suited for large multi-core systems.

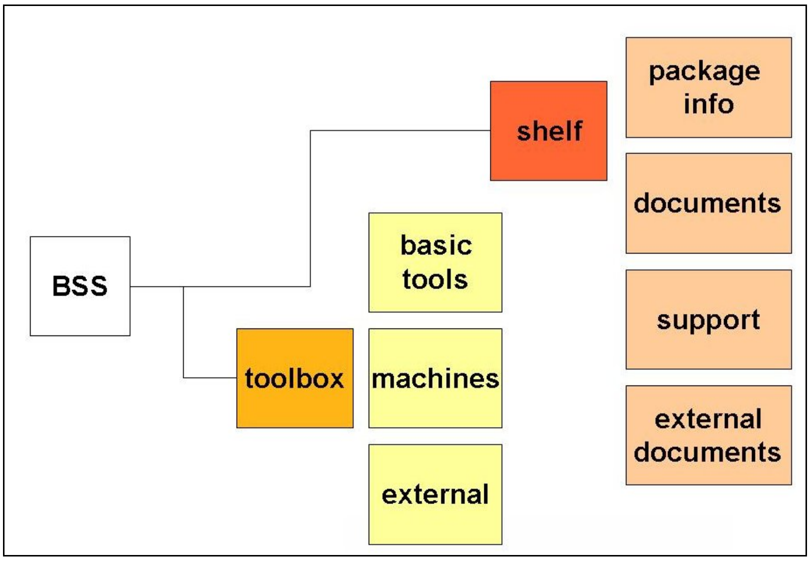


Figure 5-1. Bull Scientific Studio structure

5.1.1 Scientific Libraries and Documentation

All the libraries included in Bull **Scientific Studio** are documented in a **RPM** file called **SciStudio_shelf**.

The install paths are:

/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf -<version>

The **SciStudio_shelf** rpm is generated for each release and contain the documentation for each library included in the release. The documentation for each library is included in the directory for each library based on the type of library. All of the Scientific Studio libraries are found in **/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version**.

For example, the SciStudio libraries are found under **/SCISTUDIO_SHELF/SciStudio_shelf-<version>/<library name>**, for example, the **FFTW** documentation is included in the folder: **/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/FFTW/fftw-<version>**

If there are multiple versions of a library then, there is a separate directory for each version number.
A typical documentation directory structure for shelf RPM files is shown below:

Packaging information

- Configuration information
- README, notice
- Changelogs
- Installation

Documentation

- HowTos, tips
- Manuals
- Examples/tutorials

Support

- Troubleshooting
- Bug reports
- FAQs

External documents

- Documents related to the subject
- Weblinks

The following scientific libraries are included in **Scientific Studio** for **BAS4V5.1 Fix12**.

5.1.2 SCALAPACK

SCALAPACK stands for: SCALable Linear Algebra PACKage.

This library is the scalable version of LAPACK. Both libraries use block partitioning to reduce data exchanges between the different memory levels to a minimum. SCALAPACK is used above all for eigenvalue problems and factorizations (LU, Cholesky and QR). Matrices are distributed using BLACS.

More information is available from documentation included in the SciStudio_shelf rpm. When this is installed the documentation files will be located under:
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/SCALAPACK/ScaLAPACK-<ver>

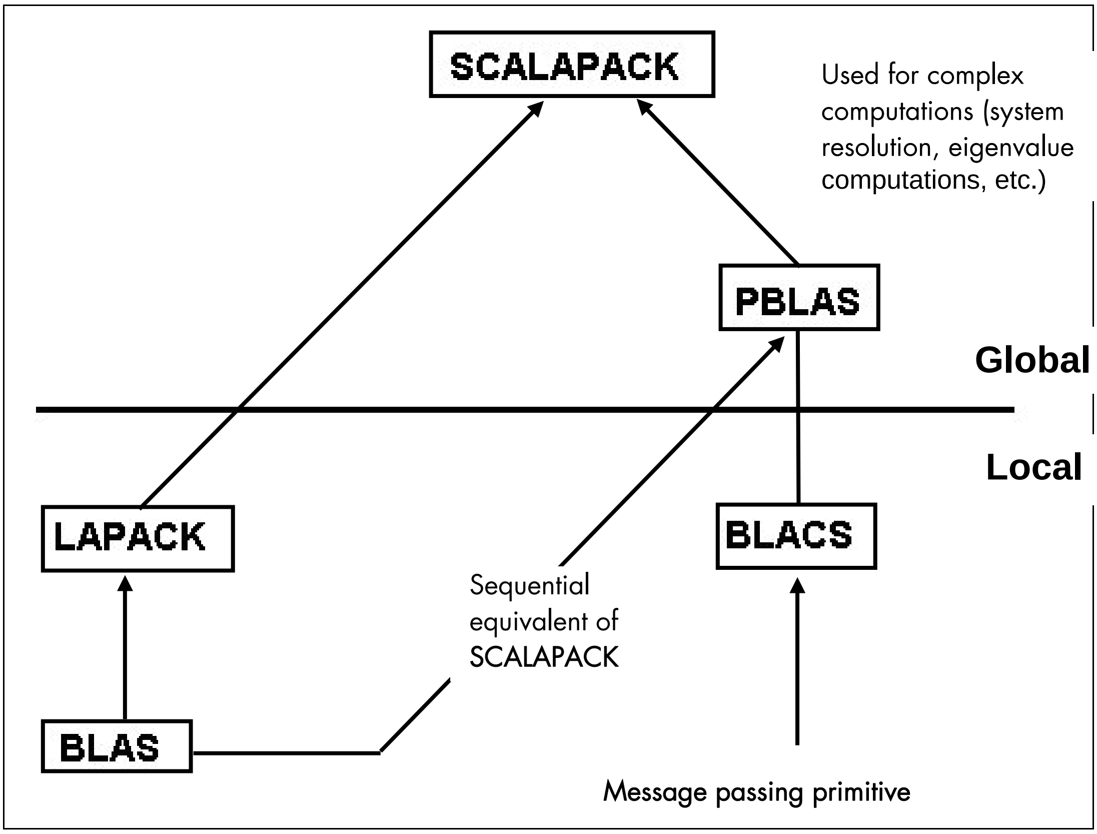

SCALAPACK
Used for complex computations (system resolution, eigenvalue computations, etc.)
PBLAS
Global
Local
LAPACK
BLACS
Sequential equivalent of SCALAPACK
BLAS
Message passing primitive

Figure 5-2. Interdependence of the different mathematical libraries (Scientific Studio and Intel)

5.1.2.1 Using SCALAPACK

Local component routines are called by a single process with arguments residing in local memory. Global component routines are synchronous and parallel. They are called with arguments that are matrices or vectors distributed over all the processes.

SCALAPACK uses MPI and thus it is delivered in three releases, corresponding to the three available MPIs.

The default installation for these three libraries is as follows:

/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/mpich_ethernet_<mpich_ethernet_versions>/lib
/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/mpibull2-<mpibull2_versions>/lib
/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/mpi_bull-<mpi_bull_versions>/lib

The following library is provided:

Libscalapack.a

Several tests are provided in the following directory:

/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/mpich_ethernet_<mpich_ethernet_versions>/tests
/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/mpibull2_<mpibull2_versions>/tests
/opt/scilibs/SCALAPACK/ScaLAPACK-<version>/ mpi_bull _< mpi_bull _versions>/tests

5.1.3 SuperLU

This library is used for the direct solution of large, sparse, nonsymmetrical systems of linear equations on high performance machines. The routines will perform an LU decomposition with partial pivoting and triangular systems solves through forward and back substitution. The factorization routines can handle non-square matrices, but the triangular solves are performed only for square matrices. The matrix commands may be pre-ordered, either through library or user supplied routines. This pre-ordering for sparse equations is completely separate from the factorization.

Working precision iterative refinement subroutines are provided for improved backward stability. Routines are also provided to equilibrate the system, estimate the condition number, calculate the relative backward error and estimate error bounds for the refined solutions. SuperLU_Dist is used for distributed memory.

More information is available from documentation included in the SciStudio_shelf rpm. When this is installed the documentation files will be located under:

/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/SUPERLU_DIST/SuperLU_DISC-<version>
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/SUPERLU_MT/SuperLU_MT-<version>
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/SUPERLU_SEQ/SuperLU_SEQ-<version>

SuperLU Libraires

The following SuperLU Libraries are provided:

/opt/scilibs/SUPERLU_DIST/SuperLU_DIST-<version>/mpich_ethernet-<mpich_ethernet_versions>lib/superlu_lnx_ia64.a

/opt/scilibs/SUPERLU_DIST/SuperLU_DIST-<version>/mpibull2-<mpibull2_versions>lib/superlu_lnx_ia64.a

/opt/scilibs/SUPERLU_DIST/SuperLU_DIST-<version>/mpi_bull-<mpi_bull_versions>lib/superlu_lnx_ia64.a

/opt/scilibs/SUPERLU_SEQ/SuperLU_SEQ-<version>/lib/superlu_ia64.a
/opt/scilibs/ SUPERLU_MT/ SuperLU_MT-<version>/lib/libsuperlu_mt_PTHREAD.a

Tests are provided for each library under the following directory:

/opt/scilibs/SuperLU/<versions>/test directory

5.1.4 FFTW

FFTW stands for the Fastest Fourier Transform in the West. **FFTW** is a C subroutine library for computing a discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and using both real and complex data.

There are three versions of FFTW in this distribution. They are located in the following directories:

/opt/scilibs/FFTW/fftw- 3.2.1 /lib
/opt/scilibs/FFTW/fftw-2.1.5/mpibull2-<mpibull2_version>/lib
/opt/scilibs/FFTW/fftw-2.1.5/mpi_bull-<mpi_bull_version>/lib
/opt/scilibs/FFTW/ fftw-2.1.5/mpich_ethernet-<mpich_ethernet_version>/lib

Tests are also available in the following directory:
/opt/scilibs/FFTW/fftw- 3.2.1 /test

More information is available from documentation included in the **SciStudio_shelf** rpm.

When this is installed the documentation files will be located under:
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/FFTW/fftw-<version>

See www.fftw.org/ for more information.

5.1.5 PETSc

PETSc stands for Portable, Extensible Toolkit for Scientific Computation. **PETSc** is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It employs the **MPI** standard for all message-passing communications (see http://www.mcs.anl.gov/mpi for more details).

The PETSc library is available under the following directories for both MPIs:
/opt/scilibs/PETSC/PETSc-<version>/mpich_ethernet-<mpich_ethernet_version>/lib
/opt/scilibs/PETSC/PETSc-<version>/mpibull2-<mpibull2_version>/lib
/opt/scilibs/PETSC/PETSc-<version>/mpi_bull-<mpi_bull_version>/lib

More information is available from documentation included in the **SciStudio_shelf rpm.**
When this is installed the documentation files will be located under:
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/PETSC/PETSc -<version>

See http://www-unix.mcs.anl.gov/petsc/petsc-2/ for more information.

5.1.6 NETCDF

NetCDF (Network Common Data Form) allows the management of input/output data. NetCDF is an interface for array-oriented data access, and is a library that provides an implementation of the interface. The NetCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data.

The library is located in the following directories:

```
/opt/scilibs/NETCDF/netCDF-<version>/bin
/opt/scilibs/NETCDF /netCDF-<version>/include
/opt/scilibs/NETCDF /netCDF-<version>/lib
/opt/scilibs/NETCDF /netCDF-<version>/man
```

More information is available from documentation included in the SciStudio_shelf rpm. When this is installed the documentation files will be located under:

```
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/NETCDF/netCDF-<version>
```

5.1.7 gmp_sci

GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface.

The main target applications for GMP are cryptography applications and research, Internet security applications, algebra systems, computational algebra research, etc.

GMP is carefully designed to be as fast as possible, both for small operands and for huge operands. The speed is achieved by using full words as the basic arithmetic type, by using fast algorithms, with highly optimized assembly code for the most common inner loops for a lot of CPUs, and by a general emphasis on speed.

GMP is faster than any other big num library. The advantage for GMP increases with the operand sizes for many operations, since GMP uses asymptotically faster algorithms.

The libraries for GMP_SCI can be found in the following directory:

```
/opt/scilibs/GMP_SCI/gmp_sci-<version>/lib/
/opt/scilibs/GMP_SCI/gmp_sci-<version>/include
/opt/scilibs/GMP_SCI/gmp_sci-<version>/info
```

More information is available from documentation included in the SciStudio_shelf rpm. When this is installed the documentation files will be located under:

```
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/GMP/gmp -<version>
```

5.1.8 MPFR

The **MPFR** library is a **C** library for multiple-precision, floating-point computations with correct rounding. **MPFR** has continuously been supported by the **INRIA** (Institut National de Recherche en Informatique et en Automatique) and the current main authors come from the **CACAO** and Arénaire project-teams at **Loria** (Nancy, France) and **LIP** (Lyon, France) respectively. MPFR is based on the GMP multiple-precision library.
The main goal of **MPFR** is to provide a library for multiple-precision floating-point computation which is both efficient and has a well-defined semantics.

The libraries for **MPFR** can be found in the following directory:

```
/opt/scilibs/MPFR/MPFR–<version>/lib/
/opt/scilibs/MPFR/MPFR–<version>/include
/opt/scilibs/MPFR/MPFR–<version>/info
```

More information is available from the documentation included in the **SciStudio_shelf** rpm. When this is installed the documentation files will be located under:

```
/opt/scilibs/SCISTUDIO_SHELF/SciStudio_shelf-<version>/MPFR/MPFR-<version>
```

Chapter 6. Profiling Programs - HPC Toolkit

HPC Toolkit provides a set of profiling tools that help you to improve the performance of the system. These tools perform profiling operations on the executables and display information in a user-friendly way.

The main advantage of **HPC Toolkit** over other profiling tools is that you do not need to include profiling options and to re-compile the executable.

Note In this section, the term "executable" refers to a Linux program file, in **ELF** (Executable and Linking Format) format.

Prerequisites:

- The executable must contain debugging information (if not, there will be no correspondence between counters and code at source line level)
- The executable must be dynamically linked because HPC Toolkit overloads the default initialization functions to call PAPI.
- The executable must not use ANSI libstdc++. (The constructor being static with the current libstdc++ at the present time, using HPC Toolkit with such an executable produces a SIGSEGV).

6.1 HPC Toolkit Tools

HPC Toolkit provides four main capabilities:

- *analysis* of an executable to recover the program structure
- *measurement* of performance metrics as the executable runs
- *correlation* of the performance metrics with the program structure
- *presentation* of the performance metrics with the associated source code

Note HPC Toolkit provides the most complete performance information when working with fully-optimized executables that include line map information within the object code. Since compilers often provide line map information for fully-optimized code, this requirement need not require a special build process.

HPC Toolkit includes the following tools:

hpcstruct *analyzes* an executable to determine its static program structure. The goal is to search for execution loops and to identify the corresponding source code procedures, loop nests, functions, and inlined code.

hpcrun-flat *measures* the execution of an executable by statistical sampling of the hardware performance counters to create flat profiles. A flat profile is an IP histogram, where IP is the instruction pointer.

hpcprof-flat *correlates* the raw profiling data from **hpcrun-flat** with the program structure file produced by **hpcstruct**. **hpcprof-flat** generates high level metrics in the form of a performance database called the Experiment database. The Experiment database is in the Experiment XML format for use with **hpcviewer**.

hpcproftt *correlates* flat profile metrics with either source code structure or object code and generates textual output suitable for a terminal. **hpcproftt** also generates textual dumps of profile files.

hpcviewer *presents* the Experiment database produced by **hpcprof-flat** by allowing the user to quickly and easily view the performance database generated by **hpcprof-flat.**

6.2 Display Counters

The **hpcrun-flat** tool uses the hardware counters as parameters. To know which counters are available for your configuration, use the **papi_avail** command or the **hpcrun-flat** tool itself:

(1) papi_avail:

```
papi_avail
```

```
Available events and hardware information.
------------------------------------------------------------------
Vendor string and code   : GenuineIntel (1)
Model string and code    : 32 (1)
CPU Revision : 0.000000
CPU Megahertz: 1600.000122
CPU's in this Node : 6
Nodes in this System: 1
Total CPU's  : 6
Number Hardware Counters : 12
Max Multiplex Counters   : 32
-----------------------------------------------------------
The following correspond to fields in the PAPI_event_info_t structure.
Name           Code        Avail    Deriv Description (Note)
PAPI_TOT_CYC   0x8000003b  Yes      No    Total cycles
PAPI_L1_DCM0   x80000000   Yes      No    Level1 data cache  misses
PAPI_L1_ICM0   x80000001   Yes      No    Level 1  instruction cache misses
PAPI_L2_DCM0   x80000002   Yes      Yes   Level 2 data cache misses
...
PAPI_FSQ_INS   0x80000064  No       No    Floating point square root
instructions
PAPI_FNV_INS   0x80000065  No       No    Floating point inverse instructions
PAPI_FP_OPS    0x80000066  Yes      No    Floating point operations
-------------------------------------------------------------------------
Of 103 possible events, 60 are available, of which 17 are derived.
```

The following counters are particularly interesting: `PAPI_TOT_CYC` (number of CPU cycles) and `PAPI_FP_OPS` (number of floating point operations).

To display more details use the **papi_avail -d** command.

(2) hpcrun-flat:

```
hpcrun-flat [informational-options]
```

Informational Options:

–l, --events-short	List available events (some may not be profilable)
–L, --events-long	Similar to events-short but with more information
--paths	Print paths for external PAPI and MONITOR
-V, --version	Print version information
-h, --help	Print help

Example:

```
hpcrun-flat -l
```

```
*** Hardware information ***
-----------------------------------------------------------------------------
Vendor string and code  : GenuineIntel (1)
Model string and code   : 32 (0)
CPU Revision            : 5
CPU Megahertz           : 1599
CPU's in this Node      : 16
Nodes in this System    : 1
Total CPU's             : 16
Number Hardware Counters: 12
Max Multiplex Counters  : 32
=============================================================================
*** Wall clock time ***
WALLCLK     wall clock time (1 millisecond period)
=========================================================================
*** Available PAPI preset events ***
-------------------------------------------------------------------------
Name            Description
-------------------------------------------------------------------------
PAPI_L1_DCM     Level 1 data cache misses
PAPI_L1_ICM     Level 1 instruction cache misses
PAPI_L2_DCM     Level 2 data cache misses
PAPI_L2_ICM     Level 2 instruction cache misses
...
PAPI_L3_TCW     Level 3 total cache writes
PAPI_FP_OPS     Floating point operations
Total PAPI events reported: 60
=========================================================================
*** Available native events ***
-------------------------------------------------------------------------
Name                         Description
-------------------------------------------------------------------------
ALAT_CAPACITY_MISS_ALL       ALAT Entry Replaced -- both integer and
                             floating point instructions
ALAT_CAPACITY_MISS_FP        ALAT Entry Replaced -- only floating point
                             instructions
ALAT_CAPACITY_MISS_INT       ALAT Entry Replaced -- only integer
                             instructions...
BRANCH_EVENT                 Execution Trace Buffer Event Captured.
                             Alias to ETB_EVENT
Total native events reported: 637
==========================================================================
```

6.3 Using HPC Toolkit

Important

It is necessary to run one of these sequences in order to produce complete results that allow you to view metrics and to analyze performance:

- hpcstruct, hpcrun-flat, hpcprof-flat, hpcviewer
- hpcstruct, hpcrun-flat, hpcproftt

6.3.1 Step 1: Analyzing the executable code (hpcstruct)

hpcstruct analyzes an executable to determine its static program structure. **hpcstruct** recovers the program structure from the executable's object code and writes a Program XML file (type=PGM) that describes that structure. This XML file is used by **hpcprof-flat** or **hpcproftt**.

hpcstruct works best with highly optimized binaries produced by C, C++, and FORTRAN programs.

Note Default values for options and switches are shown in curly brackets.

Syntax:

```
hpcstruct [options] executable > program_structure_XML_file
```

General Options:

-v, --verbose [<n>]	Verbose mode; generate progress messages to *stderr* (standard error output) at verbosity level <n>
-V, --version	Print version information
-h, --help	Print help information
–debug [<n>]	Use debug level <n> {1}
–debug-proc <glob>	Debug the structure recovery for procedures matching the procedure glob <*glob*>

Recovery and Output Options:

-i, --irreducible-interval-as-loop-off	Do not treat irreducible intervals as loops
-f, --forward-substitution-off	Assume that forward substitution does not occur (helpful for handling erroneous PGI debugging info)
-p <list>, --canonical-paths <list>	Ensure that the program structure tree only contains the files found in the colon-separated <list>. May be passed multiple times.
-n, --normalize-off	Turn off scope tree normalization

-u, - -unsafe-normalize-off Turn off potentially unsafe normalization

-c, - -compact Generate compact output

-s, - -symbolic-only Include only those program structure tree nodes that have DWARF line number information

-d, - -skip-disconnected-nodes Skip those program structure tree nodes that are disconnected from the enclosing procedure

-t [<n>], –thread [<n>] Set number of parallel threads. Default = 1. This is useful for reducing analysis time for large executables. If the specified **n** is greater than the maximum value, t will be set to the maximum value.

Example:

```
hpcstruct -v 2 -s smath.exe >smath.psxml
```

```
> hpcstruct -v 2 -s smath.exe >smath.psxml
msg: Building scope tree for [MAIN__] ...
msg: Building scope tree for [ft250_] ...
msg: Building scope tree for [xyz1_] ...
```

hpcstruct writes the Program XML structure tree for the **smath.exe** program to the file **smath.psxml**. All nodes without line number information are ignored because the **-s** option was used.

6.3.2 Step 2: Measuring the execution (hpcrun-flat)

hpcrun-flat is a flat statistical sampling-based profiler. It supports multiple sample sources during one execution and creates an Instruction Pointer (IP) histogram, or flat profile, for each sampled source. It can profile complex applications and can be used in conjunction with parallel process launchers.

The executable executes under control of **hpcrun-flat**. For an event e and a period p, after every p instances of e, a counter associated with the current IP is incremented.

When the executable terminates, **hpcrun-flat** writes the histogram into a file with the name `*executable*.hpcrun-flat.*hostname*.*pid*.*tid*`. This file is known as a profile file and contains a histogram of counts for each load module.

The user can abort the process by sending the Interrupt signal (INT or Ctrl-C). **hpcrun-flat** will write the partial profile. This technique is useful for programs that run a long time or are not well-behaved.

Syntax 1:

```
hpcrun-flat [profiling-options] [--] executable [executable-arguments]
```

General Options:

`--` The special option '`--`' stops the **hpcrun-flat** option processing; this is useful when the program specified by `executable` takes arguments of its own.

`--`**debug [<n>]** Run with debug level <n>. {1}

Profiling Options:

–e <event>[:<period>] `--`**event <event>[:<period>]**

An event to profile and its corresponding sample period. <event> can be a PAPI or native processor event. This option can be passed multiple times. It is recommended that a period always be specified. {PAPI_TOT_CYC:999999}

-r [<yes|no>], `--`**recursive [<yes|no>],**

Profile process spawned by executable_name. {no}

-t <each|all>, `--`**threads <each|all>**

Select thread profiling mode. With each, separate profiles are generated for each thread. With all, profiles of all threads are combined. Only POSIX threads are supported. {each}

-o <outpath>, `--`**output<outpath>**

Directory for output data {.}

`--`**papi-flag <flag>** Profile style flag {PAPI_POSIX_PROFIL}

Notes

- Because **hpcrun-flat** uses LD_PRELOAD to initiate profiling, it cannot be used to profile *setuid* commands. For the same reason, it cannot profile statically linked applications.
- Some events are not compatible. To resolve this problem, specify a period of time for each event using the **:period** parameter. When this option is specified **hpcrun-flat** retrieves each event in sequence, thus avoiding conflicts.
- The WALLCLK event can be used to profile the "wall" clock. It may be used only once, cannot be used with another event, and cannot have a period specified. The WALLCLK event cannot be used in a multithreaded process.

Examples:

```
hpcrun-flat -e PAPI_TOT_INS -e PAPI_TOT_CYC -o hpcrun.data -- smath.exe
```

```
>hpcrun-flat -e PAPI_TOT_INS -e PAPI_TOT_CYC -o hpcrun.data -- smath.exe
hpcrun-flat [pid 24024, tid 0x0]:
Using output file hpcrun.data/smath.exe.hpcrun-flat.sysj.24024.0x0
 The computed answer is:     500500
```

To retrieve the counters for 3000 events, enter:

```
hpcrun -e PAPI_TOT_INS:3000 -e PAPI_TOT_CYC:3000 ...
```

6.3.3 Step 3: Correlating flat metrics with program structure (hpcprof-flat)

hpcprof-flat generates high level metrics from raw profiling data produced by **hpcrun-flat** and correlates it with logical source code abstractions produced by **hpcstruct**.

Syntax 1:

```
hpcprof-flat [options] [output-options] [correlation-options]<profile-file>
```

The inputs to this usage of **hpcprof-flat** are (1) the Program XML file created by the **hpcstruct** tool and (2) the profile files created by the **hpcrun-flat** tool. If the Program XML file is not provided, **hpcprof-flat** will default to correlation using the line map information.

By default, **hpcprof-flat** generates an Experiment database file (Experiment XML format) to be used with **hpcviewer** as well as a configuration file that can be used as input to a subsequent invocation of **hpcprof-flat**.

General Options:

-v, --verbose [<n>] Verbose mode; generate progress messages to stderr (standard error output) at verbosity level <n>

-V, --version Print version information

-h, --help Print help information

–debug [<n>] Use debug level <n> {1}

Source Structure Correlation Options:

-I <path>, --include <path> Use <path> when searching for source files. A '*' after the last slash indicates recursion. This option may be used multiple times.

-S <file>, --structure <file> Use the program structure file <file> generated by the hpcstruct tool. This option may be used multiple times (e.g., for shared libraries).

Output Options:

-o <db-path>, --db <db-path>, –output <db-path>
Specify experiment database name <db-path> {./experiment-db}

–src [yes|no], --source[yes|no]
Indicates if source code files should be copied into experiment database. {yes}

Output Format Options:

Select different output formats and optionally specify the output filename *file* (located within the Experiment database). The output is sparse in the sense that it ignores program areas without profiling information. (Set *file* to '-' to write to *stdout*.)

-x [*file*], --**experiment** [*file*]

Default. Experiment XML format. {`experiment.xml`}. NOTE: To disable, set *file* to no.

–csv [*file*]

Comma-separated-value format. It includes flat scope tree and loops and is useful for downstream external tools. {`experiment.csv`}

Example:

```
hpcprof-flat -S smath.psxml hpcrun.data/*
```

```
> hpcprof-flat -S smath.psxml hpcrun.data/*
msg: Copying source files reached by PATH/REPLACE options to experiment-db
msg: Writing final scope tree (in XML) to experiment.xml
```

Syntax 2:

```
hpcprof-flat [options] [output-options] --config <config-file>
```

The general options and the output options are as listed above for **hpcprof-flat**, Syntax 1. However, the correlation options are contained in the configuration file and cannot be specified on the command line.

<config-file> is a configuration file generated by a previous **hpcprof-flat** activity and optionally edited by the user. The configuration file syntax is briefly described in Section *Configuration File Syntax*, on page6-14.

Example:

For example, the config.xml file produced by the above **hpcprof-flat** command can be modified to insert a computed metric that computes the cycles per instruction:

```
<METRIC name="CPI" displayName="CPI" percent="false">
  <COMPUTE>
    <math>
      <apply> <divide/>
        <ci>PAPI_TOT_CYC</ci>
        <ci>PAPI_TOT_INS</ci>
      </apply>
    </math>
  </COMPUTE>
</METRIC>
```

```
hpcprof-flat -S smath.psxml --config experiment-db/config.new
```

```
> hpcprof-flat -S smath.psxml --config experiment-db/config.new
msg: Computed METRIC CPI: CPI = (PAPI_TOT_CYC / PAPI_TOT_INS)
msg: Copying source files reached by PATH/REPLACE options to experiment-db
msg: Writing final scope tree (in XML) to experiment.xml
```

When the **experiment.xml** file is viewed with **hpcviewer**, it will show three columns of metrics, the native metrics for the PAPI_TOT_CYC and PAPI_TOT_INS events as well as a computed metric for CPI.

6.3.4 Step 3a: Correlating flat metrics with program structure (hpcproftt)

hpcproftt provides an alternative to **hpcprof-flat** and **hpcviewer**. **hpcproftt** correlates profile metrics with either *source code structure* (the first and default mode) or *object code* (second mode) and generates textual output suitable for a terminal. **hpcproftt** also supports a third mode in which it generates textual dumps of profile files. In all modes, **hpcproftt** expects a list of profile files as input.

hpcproftt defaults to *source structure* correlation mode. When --source is not specified, the default switches are `{pgm, lm}`; with --source, the default switch is `{sum}`.

Syntax 1: Source Structure Correlation

```
hpcproftt [--source] [options] <profile-file>...
```

In source mode, **hpcproftt** first creates raw metrics for every native event in the profile files and creates any derived metrics specified by the `--metric` option. It then correlates the metrics to the program structure based on the **hpcstruct** output file specified by the `--structure` option. If this file is not specified, a simple structure is computed from the load module's line map. **hpcproftt** finally generates the metric summaries and annotated source files to *stdout*. Each summary compares a source structure element, such as a procedure, with all other elements of that type throughout the program. Structure elements include Program, Load Module, File, Procedure, Loop, and Statement. The desired elements are chosen by switches specified with the `--source` option.

General Options:

-v, --verbose [<n>] Verbose mode; generate progress messages to *stderr* (standard error output) at verbosity level <n>

-V, --version Print version information

-h, --help Print help information

--debug [<n>] Use debug level <n> {1}

Source Structure Correlation Switches:

--source[=all,sum,pgm,lm,f,p,l,s,src] or
--src[=all,sum,pgm,lm,f,p,l,s,src]

Correlate metrics to source code structure. Without –source, the default is {pgm,lm}; with, it is {sum}

all	all summaries plus annotated source files
sum	all summaries
pgm	program summary
lm	load module summary
f	file summary
p	procedure summary
l	loop summary
s	statement summary
src	annotate source files; equiv to –srcannot '*'

Source Structure Correlation Options:

--srcannot <glob> Annotate source files with path names that match file glob <glob>. Protect globs from the shell with 'single quotes'. May pass multiple times.

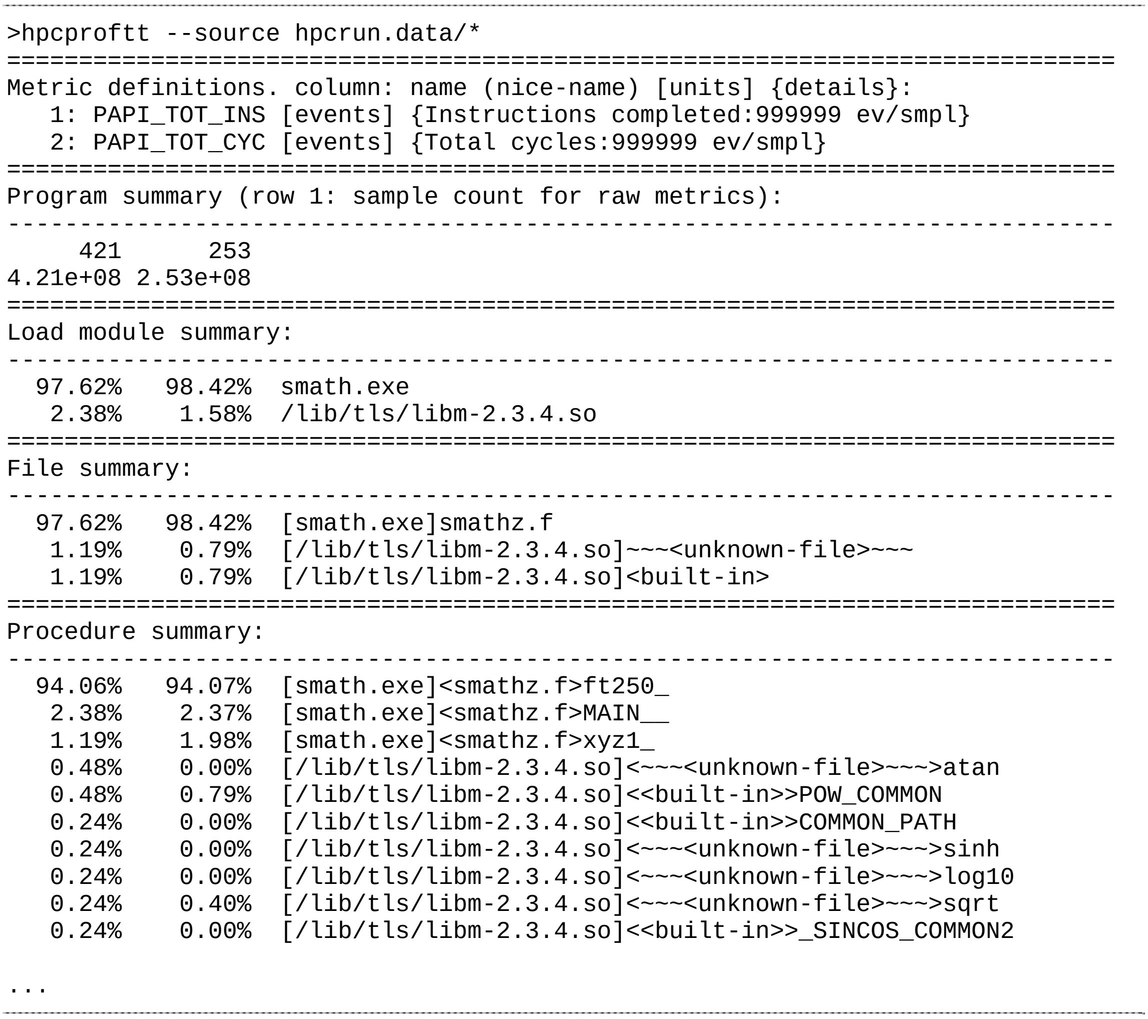
-M <metric>, --metric <metric>
Show a supplemental or different metric set. <metric> is one of the following:
sum Additionally show Mean, RStdDev, Min, Max
sum-only Show only Mean, RStdDev, Min, Max

-I <path>, --include <path>
Use <path> when searching for source files. A '*' after the last slash indicates recursion. This option may be used multiple times.

-S <file>, --structure <file>
Use the program structure file <file> generated by the **hpcstruct** tool. This option may be used multiple times (e.g., for shared libraries).

Example of Source Structure Correlation:

```
hpcproftt --source hpcrun.data/*
```



```
>hpcproftt --source hpcrun.data/*
=============================================================================
Metric definitions. column: name (nice-name) [units] {details}:
   1: PAPI_TOT_INS [events] {Instructions completed:999999 ev/smpl}
   2: PAPI_TOT_CYC [events] {Total cycles:999999 ev/smpl}
=============================================================================
Program summary (row 1: sample count for raw metrics):
-----------------------------------------------------------------------------
     421      253
4.21e+08 2.53e+08
=============================================================================
Load module summary:
-----------------------------------------------------------------------------
  97.62%   98.42%  smath.exe
   2.38%    1.58%  /lib/tls/libm-2.3.4.so
=============================================================================
File summary:
-----------------------------------------------------------------------------
  97.62%   98.42%  [smath.exe]smathz.f
   1.19%    0.79%  [/lib/tls/libm-2.3.4.so]~~~<unknown-file>~~~
   1.19%    0.79%  [/lib/tls/libm-2.3.4.so]<built-in>
=============================================================================
Procedure summary:
-----------------------------------------------------------------------------
  94.06%   94.07%  [smath.exe]<smathz.f>ft250_
   2.38%    2.37%  [smath.exe]<smathz.f>MAIN__
   1.19%    1.98%  [smath.exe]<smathz.f>xyz1_
   0.48%    0.00%  [/lib/tls/libm-2.3.4.so]<~~~<unknown-file>~~~>atan
   0.48%    0.79%  [/lib/tls/libm-2.3.4.so]<<built-in>>POW_COMMON
   0.24%    0.00%  [/lib/tls/libm-2.3.4.so]<<built-in>>COMMON_PATH
   0.24%    0.00%  [/lib/tls/libm-2.3.4.so]<~~~<unknown-file>~~~>sinh
   0.24%    0.00%  [/lib/tls/libm-2.3.4.so]<~~~<unknown-file>~~~>log10
   0.24%    0.40%  [/lib/tls/libm-2.3.4.so]<~~~<unknown-file>~~~>sqrt
   0.24%    0.00%  [/lib/tls/libm-2.3.4.so]<<built-in>>_SINCOS_COMMON2

...
```

Syntax 2:

```
hpcproftt --object[=s] [options] <profile-file>
```

In object mode, **hpcproftt** performs fine-grained correlation and generates annotated object code. It will create raw metrics for every native event in only **one** profile file.

Object Correlation Switches:

--object[=s]
--obj[=s] Correlate metrics with object code by annotating object code procedures and instructions. {}

s intermingle source line info with object code

Object Correlation Options:

–obj-values Show raw metrics as values instead of percents

–obj-threshold <n> Prune procedures with an event count < n {1}

Note On some architectures, delays between event triggers, interrupt generation, and sampling of the IP mean that an event may be associated with a different instruction from the one that caused the event. This gap may be as many as 50 to 70 instructions in length.

Example of Object Code Correlation:

```
hpcproftt --source hpcrun.data/smath.exe.hpcrun-flat.sysj.24024.0x0
```

```
>hpcproftt --object=s hpcrun.data/smath.exe.hpcrun-flat.sysj.24024.0x0
=============================================================================
Load module: smath.exe
-----------------------------------------------------------------------------
Metric definitions. column: name (nice-name) [units] {details}:
   1: PAPI_TOT_INS [samples] {Instructions completed:999999 ev/smpl}
   2: PAPI_TOT_CYC [samples] {Total cycles:999999 ev/smpl}

Metric summary for load module (totals):
       411      249

Procedure: MAIN__ (MAIN__)
------------------------------------------------------------
Metric definitions. column: name (nice-name) [units] {details}:
   1: PAPI_TOT_INS [samples] {Instructions completed:999999 ev/smpl}
   2: PAPI_TOT_CYC [samples] {Total cycles:999999 ev/smpl}
Metric summary for procedure (percents relative to load module):
        10        6
     2.43%    2.41%
Metric details for procedure (percents relative to procedure):
smathz.f:1
0x4000000000001260:                  [MII]
0x4000000000001266:                  [MII]
0x400000000000126c:                  [MII]
0x4000000000001270:                  [MII]
0x4000000000001276:                  [MII]
0x400000000000127c:                  [MII]
0x4000000000001280:                  [MFB]       nop.m 0x0
smathz.f:259
0x4000000000001286:                  [MFB]       nop.m 0x0
```

```
0x400000000000128c:                      [MFB]       nop.m 0x0
0x4000000000001290:                      [MMI]
0x4000000000001296:                      [MMI]
...
```

Syntax 3:

```
hpcproftt --dump <profile-file>
```

This form of the **hpcproftt** command will generate textual representation of raw profile data.

Example:

```
hpcproftt --dump hpcrun.data/*
```

```
> hpcproftt --dump hpcrun.data/*
=============================================================================
hpcrun.data/smath.exe.hpcrun-flat.sysm.29041.0x0
=============================================================================
--- ProfileData Dump ---
{ ProfileData: hpcrun.data/smath.exe.hpcrun-flat.sysm.29041.0x0 }
  { LM: /lib/ld-2.3.4.so, loadAddr: 0x2000000000000000 computed=0 }
    { EventData: PAPI_TOT_INS, period: 999999, outofrange: 0, overflow: 0 }
    { EventData: PAPI_TOT_CYC, period: 999999, outofrange: 0, overflow: 0 }
  { LM: /lib/libdl-2.3.4.so, loadAddr: 0x2000000000470000 computed=0 }
    { EventData: PAPI_TOT_INS, period: 999999, outofrange: 0, overflow: 0 }
    { EventData: PAPI_TOT_CYC, period: 999999, outofrange: 0, overflow: 0 }
  { LM: /lib/libgcc_s-3.4.6-2.so.1, loadAddr: 0x20000000001c0000 computed=0 }
    { EventData: PAPI_TOT_INS, period: 999999, outofrange: 0, overflow: 0 }
    { EventData: PAPI_TOT_CYC, period: 999999, outofrange: 0, overflow: 0 }
  { LM: /lib/tls/libc-2.3.4.so, loadAddr: 0x2000000000200000 computed=0 }
    { EventData: PAPI_TOT_INS, period: 999999, outofrange: 0, overflow: 0 }
    { EventData: PAPI_TOT_CYC, period: 999999, outofrange: 0, overflow: 0 }
  { LM: /lib/tls/libm-2.3.4.so, loadAddr: 0x2000000000100000 computed=0 }
    { EventData: PAPI_TOT_INS, period: 999999, outofrange: 0, overflow: 0 }
      { 0x2000000000115b60: 1 }
      { 0x2000000000116200: 1 }
      { 0x2000000000116450: 1 }
      { 0x2000000000117200: 1 }
      { 0x2000000000117890: 1 }
...
```

6.3.5 Step 4: Presenting the results (hpcviewer)

The **hpcviewer** tool displays the counters values for each code line (Figure 6-1 below).

hpcviewer uses the Experiment XML file generated by **hpcprof-flat**.

Syntax:

```
hpcviewer [experiment-database-file]
```

[experiment-database-file] is the name of the Experiment database file produced by **hpcprof-flat** or **hpcproftt**. When [experiment-database] is not specified, **hpcviewer** will prompt the user to select the Experiment database file from a directory window.

The **hpcviewer** window is divided into three panes.

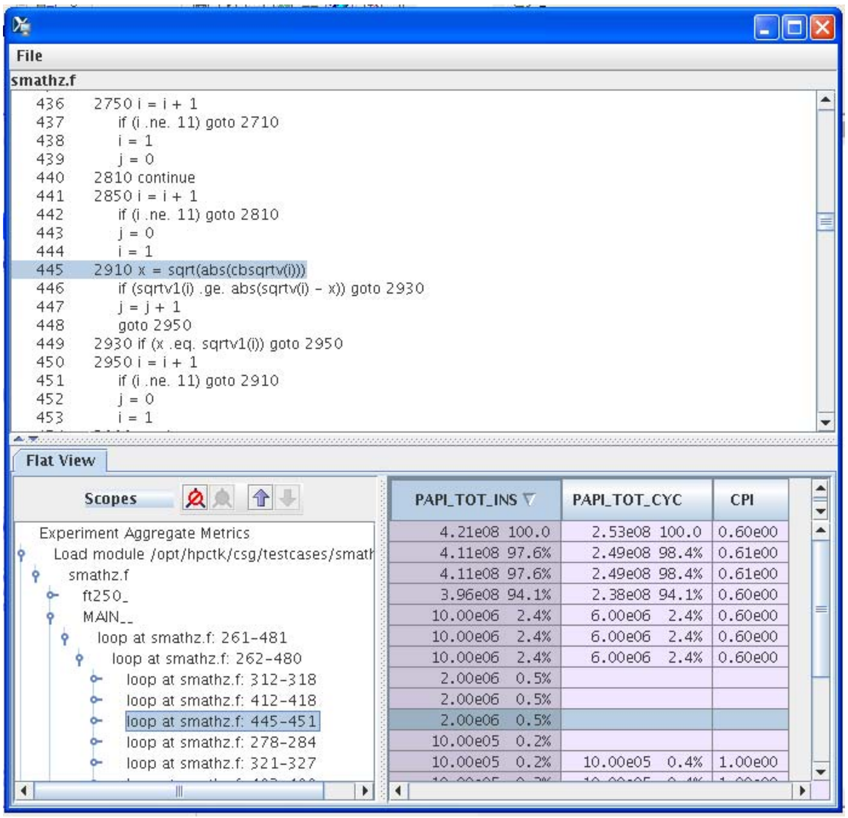
- The *source pane*, at the top of the screen, contains the source code associated with the entity currently selected in the navigation pane.
- The *navigation pane*, at the lower left, presents a hierarchical tree-based structure that identifies the display of the performance data. This pane can include load modules, source files, procedures, loops, and source lines.

  The buttons in the navigation pane control flatten and zoom. From left to right, the four buttons are:

flatten	replaces each top-level scope with its children. Useful to view and rank peers together.
unflatten	inverse of flatten. Makes previously hidden nodes visible again.
up arrow	zooms to show only information for the selected line and its descendants
down arrow	zooms out by reversing a prior zoom operation

- The *metric pane* displays the performance metrics associated with the entities to the left in the navigation pane. Entities in the tree view of the navigation pane are sorted at each level by the metric selected in the metrics pane. Sort order can be reversed by clicking on the arrow at the head of the selected column.

The following figure shows an example of the **hpcviewer** screen. There is a column for each event specified in **hpcrun-flat** as well as a third column for the computed metric that was added by **hpcprof-flat**.



```
2750 i = i + 1
     if (i .ne. 11) goto 2710
     i = 1
     j = 0
2810 continue
2850 i = i + 1
     if (i .ne. 11) goto 2810
     j = 0
     i = 1
2910 x = sqrt(abs(cbsqrtv(i)))
     if (sqrtv1(i) .ge. abs(sqrtv(i) - x)) goto 2930
     j = j + 1
     goto 2950
2930 if (x .eq. sqrtv1(i)) goto 2950
2950 i = i + 1
     if (i .ne. 11) goto 2910
     j = 0
     i = 1
```

Scopes	PAPI_TOT_INS ▽	PAPI_TOT_CYC	CPI
Experiment Aggregate Metrics	4.21e08 100.0	2.53e08 100.0	0.60e00
Load module /opt/hpctk/csg/testcases/smath	4.11e08 97.6%	2.49e08 98.4%	0.61e00
smathz.f	4.11e08 97.6%	2.49e08 98.4%	0.61e00
ft250_	3.96e08 94.1%	2.38e08 94.1%	0.60e00
MAIN__	10.00e06 2.4%	6.00e06 2.4%	0.60e00
loop at smathz.f: 261-481	10.00e06 2.4%	6.00e06 2.4%	0.60e00
loop at smathz.f: 262-480	10.00e06 2.4%	6.00e06 2.4%	0.60e00
loop at smathz.f: 312-318	2.00e06 0.5%		
loop at smathz.f: 412-418	2.00e06 0.5%		
loop at smathz.f: 445-451	2.00e06 0.5%		
loop at smathz.f: 278-284	10.00e05 0.2%		
loop at smathz.f: 321-327	10.00e05 0.2%	10.00e05 0.4%	1.00e00

Figure 6-1. View of the counter values, using hpcviewer

6.4 Configuration File Syntax

A configuration file is an XML document of type `HPCVIEW`. The following top-level elements are used in the configuration file:

```
*   <HPCVIEW>
```

Begin document.

```
    <TITLE name="my-title"/>
```

my-title names the Experiment database.

```
    <PATH name="path"/>
```

A set of PATH directives specifying path names to search for source files. path is a relative or absolute path containing source code to which performance data is correlated. In order to recursively search a directory, append an escaped '*' after the last slash, e.g., /mypath/* (escaping is for the shell).

```
    <REPLACE
    in="old-path-prefix"
    out="new-path-prefix" />
```

A set of REPLACE directives can be used to define one path prefix to operationally match another prefix occuring in profile data files or in a program structure file. This is useful when trying to compare performance metrics between machines with different file structures, e.g., because the executables or the source files are installed in different places.

```
    <STRUCTURE
    name="program.psxml"/>
```

One or more STRUCTURE directives providing program structure files created by hpcstruct

```
*   <METRIC name="name"
    displayName="name-in-display"
    display="true|false"
    percent="true|false">
    ...
    </METRIC>
```

One or more metrics.

```
*   </HPCVIEW>
```

End document.

* element is required

A metric may be of two types, **native** or **derived**. Metrics are introduced using the METRIC element and contain several attributes:

name. A unique name used when creating derived metrics that are expressions of other metrics.

displayName. Name to be displayed. Not necessarily unique.

display. Controls metric visibility. A metric used only as input to a computed metric need not be displayed.

percent. Indicates whether the viewer should display a column of percentages computed as the ratio of the metric for this scope to the metric for the whole program. Percents are useful when metrics are computed by summing contributions from descendants in the scope tree, but are meaningless for computed metrics such as ratio of flops/memory access in a scope.

The elements that appear inside the METRIC element determine its type. A metric may be of two types: **native** (type=FILE) or **derived** (type=COMPUTE).

6.4.1 Native or FILE Metrics

This type of metric appears in profile information generated by **hpcrun-flat** or by **hpcproftt**:

```
<METRIC name="m1" ...>
  <FILE name="file1" select="short-name-in-file1" type="HPCRUN|PROFILE"/>
</METRIC>
```

Because a file may contain multiple metrics, the `FILE` element has an optional 'select' attribute to identify a particular metric within the file. Metrics are identified by their 'shortName' values, typically zero-based indices. The default 'select' value is 0 and corresponds to the first metric.

6.4.2 Derived or COMPUTE Metrics

Derived metrics are specified by a `COMPUTE` element containing a MathML equation in terms of metrics defined earlier in the HPCVIEW document.

hpcprof-flat supports the following operands:

- **constants**: `<cn>2</cn`
- **variables**: `<ci>m1</ci>` (used to refer to other metrics)

and the following MathML operators (used within `<apply>`):

- **negation**: `<minus/>` (1-ary)
- **subtraction**: `<minus/>` (2-ary)
- **addition**: `<plus/>` (n-ary)
- **multiplication**: `<times/>` (n-ary)
- **division**: `<divide/>` (2-ary)
- **exponentiation**: `<power/>` (2-ary)
- **minimum**: `<min/>` (n-ary)
- **maximum**: `<max/>` (n-ary)
- **mean (arithmetic)**: `<mean/>` (n-ary)
- **standard deviation**: `<sdev/>` (n-ary)

Consider the examples from the previous sections with two native metrics for PAPI_TOT_CYC (cycles) and PAPI_TOT_INS (instructions).

The file config.xml from the example produced by hpcprof-flat, contains the following elements, including only native metrics:

```
<HPCVIEW>
<TITLE name=""/>
<STRUCTURE name="smath.psxml"/>
<METRIC name="PAPI_TOT_INS" displayName="PAPI_TOT_INS" sortBy="true">
  <FILE name="hpcrun.data/smath.exe.hpcrun-flat.sysj.29041.0x0"
     select="0" type="HPCRUN"/>
</METRIC>
<METRIC name="PAPI_TOT_CYC" displayName="PAPI_TOT_CYC">
  <FILE name="hpcrun.data/smath.exe.hpcrun-flat.sysj.29041.0x0"
    select="1" type="HPCRUN"/>
</METRIC>
</HPCVIEW>
```

The file config.new from example produced by hpcprof-flat and subsequently edited by the user, contains the following elements, including both native and derived metrics:

```
<HPCVIEW>
<TITLE name=""/>
<STRUCTURE name="smath.psxml"/>
<METRIC name="PAPI_TOT_INS" displayName="PAPI_TOT_INS" sortBy="true">
  <FILE name="hpcrun.data/smath.exe.hpcrun-flat.sysj.29041.0x0"
    select="0" type="HPCRUN"/>
</METRIC>
<METRIC name="PAPI_TOT_CYC" displayName="PAPI_TOT_CYC">
  <FILE name="hpcrun.data/smath.exe.hpcrun-flat.sysj.29041.0x0"
    select="1" type="HPCRUN"/>
</METRIC>
<METRIC name="CPI" displayName="..." percent="false">
  <COMPUTE>
    <math>
      <apply> <divide/>
        <ci>PAPI_TOT_CYC</ci>
        <ci>PAPI_TOT_INS</ci>
      </apply>
    </math>
  </COMPUTE>
</METRIC>
</HPCVIEW>
```

6.5 More Information

For more detailed information about HPC Toolkit go to:

http://hpctoolkit.org/man/hpctoolkit.html

Chapter 7. Profiling Tools - profilecomm

This chapter provides corrections for the following documentation:
HPC BAS4 Application Tuning Guide, (Ref 86 A2 19ER 06),
> Chapter 1 (*Performance Monitoring and Application Tools*)
> Section 1.7 (*mpianalyser and profilecomm*).

Notes
- Only the modified paragraphs are supplied.
- The titles of the sections are identical to those of the *Application Tuning Guide*.

7.1 profilecomm Options

Different options may be specified for **profilecomm** using the **PFC_OPTIONS** environment variable.

For example:

```
export PFC_OPTIONS=”-f foo.pfc”
```

Examples

- To profile the **foo** program and save the results of the data collection in the default file **mpiprofile.pfc**:

```
$ MPIANALYSER_PROFILECOMM=1 srun –p my_partion –N 1 -n 4./foo
```

- To save the results of the data collection in the `foo.pfc` file:

```
$ MPIANALYSER_PROFILECOMM=1 PFC_OPTIONS="-f foo.pfc" srun
–p my_partion –N 1 -n 4./foo
```

- To save the result of the collect in text format in the `foo.txt` file:

```
$ MPIANALYSER_PROFILECOMM=1 PFC_OPTIONS="-t -f foo.txt" srun
–p my_partion –N 1 -n 4./foo
```

7.2 Messages Size Partitions

Note **Profilecomm** supports a maximum of 10 partitions only.

7.3 Profilecomm Data Analysis

Readpfc output

Note The header, histograms, statistics and topology sections are not included in the output when the **-t**, **-text** text format options are used.

7.3.1 Exporting a Matrix or an Histogram

- To export the first part (small messages) of point to point numerical communication matrices in PostScript format in the foo.ps file:

```
$ readpfc -x np.0 -f ps -o foo.ps foo.pfc
$ ls foo.ps
```

```
foo.ps
```

This is a zero (0), not the O letter.

Chapter 8. Intel Tools and Applications

This chapter describes how to install **Intel** compilers and tools.

Intel ® **Math Kernel Library** and the **Intel Debugger (IDB)** are supplied with **Intel Professional Edition for Linux version** 11.1 Compilers.

See Intel compilers require that the **Intel**® **License Manager for FLEXlm** is in place. See the **INSTALL.txt** file provided by **Intel**® for more details regarding the installation of the **Intel**® **License Manager for FLEXlm**. See the Licensing chapter in the *Software Release Bulletin* for more information on **FLEXlm.**

8.1 Installing Intel Compilers with MKL and IDB

Follow the installation routine below to install the **Intel**® **C++** and the **Fortran** compilers, together with the **Intel**® **Math Kernel Library** and the **Intel**® **Debugger**. These tools are installed on the node which contains the Login functionality (this may be a dedicated node or one which is combined with the I/O and/or Management functionalities).

Note Compilers and tools must be installed on each Login Node separately.

1. Install the **Intel** Compilers (**Fortran**, **C/C++**) on the Login Node.

2. Install the **Intel MKL** on the Login Node.

3. Install the **Intel Debugger (IDB)** on the Login Node.

See The **INSTALL.txt** file provided by **Intel** for more details regarding the installation of the **Compilers**, **MKL** and **IDB**.

4. Export the **/opt/intel** directory via **NFS** and mount it on the **Compute** nodes.

8.2 Intel Trace Analyzer and Collector Tool

Intel Trace Analyzer and Collector is supplied directly by **Intel** to the customer. The **Intel Trace Tool** uses the **FlexLM** license scheme. The recommended path for installation is **/opt/intel/itac/<rel number 1>**.

1. Install the **Intel** Trace Tool on the **Login Node**.

2. Export the **/opt/intel** directory via **NFS** and mount it on the **Compute** nodes.

See The **INSTALL.txt** file provided by **Intel**, and the documentation available from the **Intel** site, for more details regarding the installation of Intel Trace Analyzer and Collector.

8.3 Intel VTune Performance Analyzer for Linux

For more details about the installation procedure see the *Intel® VTune Performance Analyzer for Linux Installation Guide* on the internet site:
http://www.intel.com/software/products/cluster

Note If Intel® **VTune Performance Analyzer for Linux** is to be installed on the cluster, the HPC Toolkit (**XTOOLKIT**) product must be installed - see Chapter 6 in this manual.

8.4 Intel Version 11.1 Runtime Libraries

Intel version 11.1 runtime libraries are included with **Intel Compiler Suite** version 11.1 media (a **tgz** file with an **.sh** installation script) and must be installed on all nodes (LOGIN, LOGIN-MANAGEMENT and COMPUTE) which do not have the **version 11.1** compilers installed on them.

BEFORE the **Intel version 11.1** runtime libraries are installed, verify that the **BAS4 V5.1 FIX 11** version 9 Intel Runtime has been removed from all nodes

See *BAS4 V5.1 Fix11 to Fix12 Upgrade Procedure - Sections 3.1.3 and 3.2.4.*

This is done by using the command below:

```
rpm -e intelruntime-cc_fc-b.91044_91039.Bull
```

The applications delivered with the **BAS4 V5.1 Fix12** have been compiled with **Intel version 11.1** compilers.

Note There is no need to recompile programs compiled with earlier **Intel** compiler version, as forward compatibility is guaranteed by Intel.

The **/opt/intelruntime/<version>** path should be added to the **LD_LIBRARY_PATH** environment variable in the shell configuration file.

If a different version of an **Intel** compiler is used, then its runtime libraries have to be installed on the nodes without the compilers, in order to ensure coherency, and the path in the **LD_LIBRARY_PATH** variable modified to include the new version reference.

Index

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

REFERENCE
86 A2 44FD 00